Université de Liège
Aerospace and Mechanical Engineering Department
Multibody and Mechatronic Systems Lab

Numerical simulation of frictional contact problems in flexible multibody dynamics

Thesis submitted in fulfilment of the requirements for
the degree of Doctor in Engineering Sciences

by

Javier GALVEZ

April 2020

Promotor:

Dr. Olivier BRÜLS, Université de Liège

"Young man, in mathematics you don't understand things. You just get used to them."

John von Neumann

Abstract

The simulation of mechanical systems subjected to impacts and friction requires to solve highly non-linear systems of equations stemming from the Signorini's conditions and the Coulomb friction's law, which present several practical difficulties, which are not yet completely solved.

For systems involving nonsmooth phenomena the common methods used to study the dynamics of structures with a finite element spatial discretization, such as the Newmark method, the Hilbert-Hughes-Taylor method (HHT) and the standard generalized-α method, can no longer be applied. These implicit integrators assume that the kinematic variables are smooth. However, in the presence of instantaneous changes of the velocity, which occurs due to the impact effects, these methods may produce numerical solutions with notable precision losses, non physical behaviors and the generation of fictitious energy at the contact instant. Thus, nonsmooth time integration methods able to deal with nonsmooth motion equations are needed.

This work presents the development of a robust and accurate time integrator. The integrator is built upon a previously developed nonsmooth generalized-α scheme time integrator which was able to deal well with nonsmooth dynamical problems avoiding any constraint drift phenomena and capturing vibration effects without introducing too much numerical dissipation. However, when dealing with problems involving nonlinear bilateral constraints and/or flexible elements, it is necessary to adopt small time-step sizes to ensure the convergence of the numerical scheme. In order to tackle these problems more efficiently, a fully decoupled version of the nonsmooth generalized-α method is proposed in this work, avoiding these inconveniences.

To account for friction, a new node-to-face contact element compatible with the proposed nonsmooth generalized-α solver has been developed. The node and face can be attached, each one, to either flexible or rigid bodies. For the sake of robustness and numerical performance, the frictional contact problem is treated using an augmented Lagrangian technique inspired by the work of Alart and Curnier for quasi-static problems.

Finally, this methodology has been implemented in the general purpose finite element software Oofelie. The algorithm and the frictional contact element have been coded using the existing data structure and in a non intrusive manner, in order to preserve the compatibility with the existing utilities and the wide element library. Using this implementation several numerical experiments have been done. The results of these examples have been compared to analytical solutions or previous numerical solutions obtained by other authors showing good agreement and convergence rate.

Acknowledgements

For me, this Ph.D. has not only been another educational step on my career, it has represented an important life experience and a huge challenge, only possible because a lot of incredible people have been involved supporting me in many different ways. Therefore, this part of the thesis is the one I am going to write with more happiness and passion, trying to reflect my more sincere gratitude.

Firstly, I would like to thank my advisor Olivier Brüls for giving me the chance to carry out the Ph.D. under his supervision at the University of Liege, being always open to answer any question I could have showing and incredible patience.

This work received financial support from the M4 project funded by the Walloon Region (Pole MecaTech), which is gratefully acknowledged.

I would like to specially thank Alejandro Cosimo for his infinite patience explaining me again and again all my questions and for helping me with the writing of the papers. The main reason I have been able to write this thesis is because I have had the luck of having him and Olivier helping me to understand the fundamental concepts.

Thanks to Open Engineering team, specially Fernando, Juan Pablo and Stéphane.

Many thanks also go to Prof. Alberto Cardona for the chance to spend three months under his guidance at the Universidad Nacional del Litoral - CONICET in Santa Fe, Argentina. Also to Federico Cavalieri for welcoming me in Argentina, and all the explanations and help he gave me generously since the beginning of my Ph.D. A Matias, Leo, Luz, Lali y Flor por enseñarme lo amigables que son los argentinos y la pasión por el asado.

I would like to thank all the people in the service and specially my amazing office mates which gave me incredible times, Arthur, Juliano, Robin, Bruno and Asmaa.

Many thanks also go to Stéphane Paquay and Professors Eric Bechet, Ludovic Noels, Jean-Claude Golinval, Vincent Acary and Olivier Verlinden who accepted to participate in the jury of this doctoral thesis.

To the hole group of erasmus I shared the "amazing" residence of Sart Tilman, which helped me land in Liege.

A todo el grupo de amigos que he conocido y me han ayudado a conocer mejor la cultura de otros lugares, con los que he compartido momentos inolvidables y la parte más divertida de estos cuatro últimos años. Con especial cariño a Gus, Celia, Carlos, Hector, Ariel, Vale, JP, Fanny, Fer, Yanis, Bea, Catarina, Antonela, y muchos mas!

A la increíble familia de Rue Sclessin 49, sin duda son lo mejor de estos cuatro últimos años! Gracias por aguantarme, cuidarme, cocinarme, ayudarme... ¡Gracias Miguel, Melisa y Alessio!! Aussi à l'incroyable propriétaire de la maison Colette Gerday.

A mi grupo de amigos de Badalona que cuando necesitaba desconectar estaban ahí como si nunca me hubiese ido. Víctor, Ivan, Ramón, Josep, Jaume y Cristina. Especialmente a Ferran y Cristian por su gran apoyo moral.

A Juanjo por trasmitirme la pasión por la física desde pequeño. A Iñaki por sus sabios pero no siempre aceptados consejos y animarme a terminar el doctorado.

Muchas gracias a toda mi familia por su apoyo incondicional y animarme siempre a que continuase. Especialmente a mi madre.

*Gracias a mi partner in crime que ha sufrido esta tesis tanto como yo, O MÁS! Anto :**

Contents

List of Figures

List of Tables

List of Abbreviations

DB:	data base
CAE:	computer-aided engineering
CAD:	computer-aided design
COM:	center of mass
COR:	coefficient of restitution
DAE:	differential algebraic equations
DMI:	differential measure inclusions
DOF:	degrees of freedom
FEM:	finite element method
HHT:	Hilbert-Hughes-Taylor method
LCP:	linear complementary problem
CP:	complementarity problem
MBS:	multibody system
NSGA:	nonsmooth generalized-α scheme
ODE:	ordinary differential equations
Oofelie:	Object Orient Finite Elements Led by Interactive Executor

List of Symbols

Latin symbols

a_g	gravity
$\mathcal{C}_{\lambda_N}$	set that defines a disk in the friction cone for the value λ_N
$\mathrm{d}\boldsymbol{i}$	differential measure associated to the impulse of the reaction force
$\mathrm{d}\boldsymbol{v}$	differential measure associated to the velocity
$\mathrm{d}\boldsymbol{w}$	nonsmooth impulsive contribution of the velocity measure
e	restitution coefficient
$\boldsymbol{f}$	force vector in the equations of motion
$\boldsymbol{F}$	elementary force of the frictional contact element
$\boldsymbol{g}(\boldsymbol{q})$	constraints
$\boldsymbol{g}_{\boldsymbol{q}}$	constraint gradient
$g_N(\boldsymbol{q})$	constraint in the normal direction of the contact
$\boldsymbol{g}_T(\boldsymbol{q})$	constraints in the tangential direction of the contact
$\mathring{g}$	impact law
G	Lie group
h	time step size
$\boldsymbol{J}$	inertial tensor with respect the center of mass in the body attached frame
k	scaling factor
$\boldsymbol{K}_G$	iteration matrix of the frictional contact element
$\mathcal{K}$	kinetic energy
$\mathcal{L}$	augmented Lagrangian
m	mass
$\boldsymbol{M}$	mass matrix
$\boldsymbol{n}$	unitary vector normal to the contact surface in the current configuration
$\boldsymbol{N}$	unitary vector normal to the contact surface in the reference configuration
p_i	reaction impulse of constraint i
$\boldsymbol{p}$	reaction impulses
p	penalty parameter
$\boldsymbol{q}$	configuration variable
$\tilde{\boldsymbol{q}}$	smooth contribution to the position
$\boldsymbol{r}$	residue
$\boldsymbol{R}$	rotation matrix in $SO(3)$
$\mathcal{S}$	arbitrary set
$\boldsymbol{S}_t$	iteration matrix of the system
t	time

$\boldsymbol{U}$	position correction
$\boldsymbol{v}$	velocity
$\tilde{\boldsymbol{v}}$	smooth part of the velocity
$\boldsymbol{v}^-$	pre-impact velocity
$\boldsymbol{v}^+$	pos-impact velocity
$\boldsymbol{t}_1$	unitary vector tangent to the contact surface in the current configuration
$\boldsymbol{T}_1$	unitary vector tangent to the contact surface in the reference configuration
$\boldsymbol{t}_2$	unitary vector tangent to the contact surface in the current configuration
$\boldsymbol{T}_2$	unitary vector tangent to the contact surface in the reference configuration
$\mathcal{V}$	potential energy
$\mathcal{W}$	virtual work of a multibody system
$\boldsymbol{W}$	velocity jump
$\boldsymbol{x}$	position vector in the current configuration
$\boldsymbol{X}$	position in the reference configuration
Greek symbols	
α_m	numerical parameter of to the generalized-α scheme
α_f	numerical parameter of to the generalized-α scheme
β	numerical parameter of to the generalized-α scheme
γ	numerical parameter of to the generalized-α scheme
δ_{t_i}	Dirac delta at time t_i
Δt	impact duration
$\Delta\boldsymbol{\Theta}_A$	rotation increment vector
θ_μ	angle of the friction cone
$\boldsymbol{\lambda}$	reaction force during the collision
$\boldsymbol{\Lambda}$	Lagrange multiplier at velocity level
$\tilde{\boldsymbol{\lambda}}$	Lagrange multiplier of the non-impulsive contribution
μ	friction coefficient
$\boldsymbol{\mu}$	additional Lagrange multiplier of the GGL formulation
$\boldsymbol{\nu}$	Lagrange multiplier at position level
$\boldsymbol{\xi}$	augmented Lagrange multiplier at position level
ρ_{inf}	spectral radius at infinity
$\boldsymbol{\sigma}$	augmented Lagrange multiplier at velocity level
ψ	indicator function
$\boldsymbol{\Omega}$	angular velocities in the body-attached frame
Superscript	
$\mathcal{A}$	set of bilateral and active unilateral constraints at position level
$\overline{\mathcal{A}}$	set of inactive unilateral constraints at position level
$\mathcal{B}$	set of bilateral and active unilateral constraints at velocity level
$\overline{\mathcal{B}}$	set of inactive unilateral constraints at velocity level
$\mathcal{C}$	set of bilateral and unilateral constraints
^{damp}	damping
^{ext}	external

^{gyr}	gyroscopic
^{inert}	inertial
^{int}	internal
^{j}	index of the contacts
$^{\mathcal{U}}$	unilateral constraints
$^{\overline{\mathcal{U}}}$	bilateral constraints
^{p}	position
^{v}	velocity
$^{-}$	left limit
$^{+}$	right limit

Subscript

i	index of the instants where an impacts occurs
N	normal direction of the contact
T	tangential direction of the contact
$_{abs}$	absolutely continuous function
$_{step}$	step function
$_{singular}$	singular function
$_{f}$	friction
s	smooth
$_{p}$	position
$_{v}$	velocity

Operators

$(\dot{\ })$	derivative respect to time
$(\ddot{\ })$	second derivative respect to time
$\| \ \|$	Euclidean norm
d	Lebesgue measure
δ	variation
∂	subdifferential
$\boldsymbol{S}(\ )$	linear mapping from R3 to the 3×3 skew-symmetric matrices group
$\boldsymbol{S}_{\mathbb{R}^3 \times SO(3)}(\ )$	linear mapping from R6 to the lie algebra

Symbols

$\perp$	complementarity condition
$\exists$	there exist
$\equiv$	equivalent
$\forall$	for all
$\mathfrak{g}$	Lie algebra
$\in$	in
$\cup$	the union of
$\mathbb{R}^-$	set of all negative real numbers
$\mathbb{R}^+$	set of all positive real numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}^2$	set of the two-dimensional real coordinate space

$\mathbb{R}^3$	set of the three-dimensional real coordinate space
$SO(3)$	special orthogonal group
$\mathbb{R}^3 \times SO(3)$	6-dimensional Lie group
$SE(3)$	special Euclidean group

Chapter 1

Introduction

1.1 Motivation

Since the first rudimentary mechanisms were built with wood and ropes, they have been evolving constantly until the complex mechanisms that we have nowadays. Lately, the automotive, robotic and aerospace industries have pushed the development of more efficient and complex mechanisms, involving e.g. light materials, irregular shape cams, spring-dampers, gears, belts and cables. In the design phase of such mechanism, it is important to calculate the forces acting on the different parts in order to optimize them. Also in robotics or automation process, developing a proper control system is important to accurately predict the dynamic behavior of the system. In consequence, the field of multibody system (MBS) dynamics developed itself encouraged by the industry demands.

The study of MBS consists in the analysis of the behavior along time of different interconnected mechanical elements under the action of external loads for some given initial conditions. The first analysis of interconnected solids was done in the eighteenth century by Lagrange where he introduced the systems of differential algebraic equations (DAE) and ordinary differential equations (ODE) that describe the movements of the mechanical systems. Although MBS simulations represent a considerable cost reduction with respect to the experimental tests [1] it was not until the last decades with the improvement in the computational hardware, that MBS simulations have become a computer-aided engineering tool widely used in the product design.

In the robotic field, to accurately control the most advanced grasping devices, modeling systems with rigid and/or flexible bodies and frictional contact is essential [2]. Also many mechanical engineering applications are subjected to high frequency vibrations produced by impacts between components that can reduce the service life of the entire system, for example: gearboxes [3] and electrical circuit breakers [4]. Thus, to face these problems and improve the design, it is important to have an accurate prediction of the velocities, accelerations and forces that occur within the mechanism. Therefore, it is necessary to have a MBS methodology able to take in account the frictional impact effects as well as the flexibility of bodies to capture the influence of vibrations.

1.2 Multibody systems

In this section an overview of the key consideration about the modeling of MBS dynamics and the main different approaches are explained, also some key references are given. In MBS the bodies involved can be simulated as rigid bodies or flexible bodies. They can translate and rotate, but the main difference is that the flexible bodies undergo deformations under the actions of forces, and the rigid bodies do not. This rigidity assumption is a relevant approximation for stiff rigid bodies that do not have relevant deformations, and it leads to simple and compact equations of motion. This thesis aims at modeling systems which may include both rigid and flexible structural components. The book of Géradin and Cardona [5] addresses the main aspects to model flexible bodies systems used in this work. Also, the book of Bauchau [6] is a useful reference to introduce the reader in the field of flexible multibody dynamics since it starts with a review of the basic mathematical tools and present the concepts gradually. A global review about the different computational strategies for flexible multibody systems can be found in [7].

The methodologies to study the time evolution of MBS composed by flexible bodies depending on the type of reference frame used can be classified in three main groups: floating frame, corotational frame and inertial frame approaches. A description and literature review of the three methods can be found in [7]. In the first two groups, in addition to the inertial frame, understood as a global reference frame to describe the motion, other intermediate reference frames are considered. These intermediate frames are attached to the flexible components and follow the average local rigid body motion. Then the flexibility contribution is computed with respect to these local frames. An important difference between the two methods is that in the floating frame approach the intermediate frame follows an average rigid body motion of the entire flexible component, whilst in the corotational approach it follows an average rigid body motion of an individual finite element within the flexible component. In the inertial frame approach no intermediate frame is used and the deformations are computed directly with respect the global reference frame. The difference between these three approaches can be better understood with the help of the Fig. 1.1. The inertial frame method was developed to deal with nonlinear finite element discretization of flexible bodies, and it was applied to consider large rotations and deformations in the analysis of continuum mechanics, including also large strains and large deflections [7]. Since the objective is to study MBS within a finite element framework, the inertial frame approach is used in this work.

1.3 Impacts

In this section, the key physical aspects of an impact are reviewed. In the case of impacts between rigid bodies, nonsmooth phenomena arise which require a specific

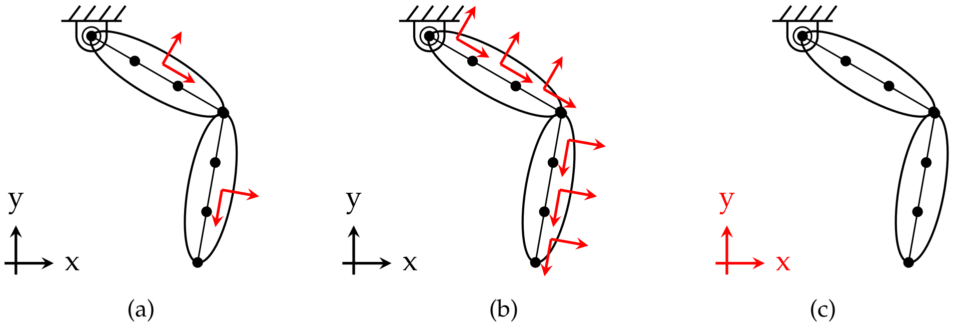


FIGURE 1.1: Reference frames approaches: (a) Floating frame. (b) Corotational frame (c) Inertial frame

modeling framework. The theory background of nonsmooth modeling techniques can be found in the books of Pfeiffer and Glocker [3] and Wriggers [8].

In physics an impact between two bodies is a complex phenomenon which has a very short duration (order of miliseconds [9]), high forces levels, rapid energy dissipation, local elastic or plastic deformations, large changes in the velocities and vibration effects [10].

A fully elastic model of objects impacting each other would capture not only the global behavior during the impact contact but also the local deformation pattern at the contact point and the elastic vibrations after the impact. However, the motion details due to the body elasticity are not relevant in all applications depending on the time scale of interest. Therefore, it is not always appropriate to develop a model that captures the vibrations, it depends on the analyzed system and the studied phenomena.

When the assumption of rigid bodies is done, the system state variables have instantaneous changes in the contact instant, transforming the position into a non differentiable function, the velocity into a discontinuous function and the acceleration into a Dirac measure. To clarify this phenomenon, let us imagine a rigid ball falling and bouncing against a rigid surface. Before and after the impact, the ball has a momentum in opposite direction which means that the velocity has a jump and the contact reaction forces must be impulsive. Notice that when flexible bodies enter in contact, the velocity field of the bodies at the contact point or at the contact surface is also discontinuous in time. Nevertheless, the contact reaction forces or the contact reaction pressure remains finite in this case.

A rough classification of the contact modeling techniques leads to two main groups: contact force based methods, also called compliant methods, and methods based on geometrical constraints.

The contact force based methods consider the contact as a smooth phenomenon with a short, but finite, duration. In this case, the bodies are considered globally rigid, but locally flexible in the contact region. This local elastic/plastic deformations in the contact area can be modeled with some spring-damper elements scattered over its surface. Therefore, the contact force is a smooth function which depends on the bodies penetration, as shown in Fig. 1.3(a)

The methods based on geometrical constraints assume the impact as an instantaneous phenomenon, with a time duration equal to zero, which is represented as a complementarity condition (represented by the symbol $\perp$) between the unilateral constraint and the impulse reaction force. The impulse reaction force is computed such that the penetration constraint is avoided. The idea of the complementarity constraint is to impose that the unilateral constraint or the impulse reaction force are both positive and that one of them is equal to zero, as it can be observed in Fig. 1.3(a). Therefore, their product is always zero. This complementarity condition to describe impact contacts in the MBS with unilateral contacts is also called the Signorini condition [11].

1.4 Friction

In real systems friction appears in all the contacts and it occurs in the interface between bodies. Friction has been deeply studied, nevertheless it is a complex phenomenon which involves several physical processes and overall it is not completely understood [12, 13, 14]. The first mathematical formulation for frictional contact problems was proposed more than 200 years ago by Coulomb and then followed by Hertz [15]. Although the Coulomb law is very simple, it allows one to represent important effects in a wide range of applications. However, this model is not able to account for several phenomena that occur during the friction process, e.g. the pre-sliding friction [16], the Stribeck effect [17], the difference between static and dynamic friction [18] or the viscous friction [19]. A review of these frictional phenomena can be found in Armstrong *et al.* [20], and their influence on the velocity/force characteristics is represented in Fig. 1.2. Therefore, since this seminal work, great efforts were made to better represent the friction process by including tribological properties, such as the lubrication conditions, the plastic deformations and the geometric changes of the contact surfaces, at the cost of an increased difficulty of the analysis. A lot of empirical models have also been proposed in order to describe frictional behavior based on experimental observations. Several authors have presented a broad review of the different frictional laws, and studied their viability to be implemented with numerical methods [21, 22, 23, 24, 25, 12, 26].

Frictional models can be split in two different groups, viscous friction and dry friction [3]. In the first one, the forces are determined by the local tribological effects at the contact. In the second one, the force opposes itself to the relative motion between the two bodies in contact, with two possible states: the static friction for non existing relative displacements and the kinetic friction for sliding between the parts. The Coulomb's friction law is a dry friction model, in which only one macroscopic parameter is considered in the formulation: the friction coefficient μ, which proportionally relates the normal force with the friction force. The friction force always acts in the opposite direction of the relative velocity, $\boldsymbol{v}_{rel}$, between the contact points,

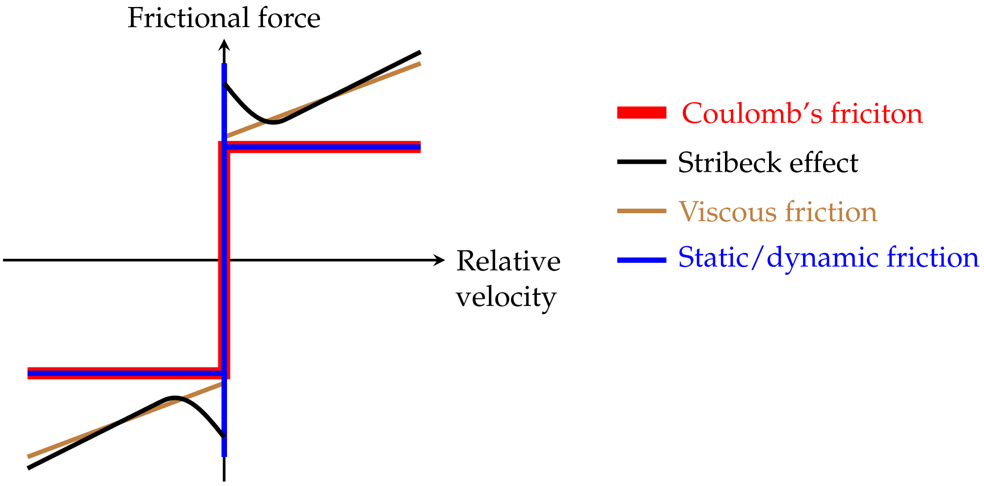


FIGURE 1.2: Friction phenomena

which is expressed as $\boldsymbol{F}_f = -\mu\|\boldsymbol{F}_N\|\boldsymbol{v}_{rel}/\|\boldsymbol{v}_{rel}\|$ for the sliding condition, where $\boldsymbol{F}_f$ is the friction force and $\boldsymbol{F}_N$ is the normal force. On the other hand, if the relative velocity is zero the contact is in sticking condition, and the friction force is not known beforehand but satisfies the condition $\|\boldsymbol{F}_f\| < \mu\|\boldsymbol{F}_N\|$. In the Coulomb's law, the friction force is not a mere function of the relative velocity because of the infinite slope at the origin, as it can be seen in Fig. 1.3(b). This model is not differentiable at the origin, which increases the models complexity and, therefore, the computation of its solution. As for the normal force, some regularization techniques have been developed to obtain a smooth frictional force function, as depicted in Fig. 1.3(b). Nevertheless, these regularization techniques show other limitations and inconveniences, for example the impossibility to represent a block resting in an inclined plane. This limitation occurs because with the regularization techniques zero velocity implies no frictional forces acting. Therefore, the block will start to slide down the plane with an increasing velocity until the frictional force is enough to balance the gravity force. The slope of the friction law function has a great influence on the results, an infinity slope is needed to reproduce a true sticking behavior, which is not feasible based on a regularized contact model. This motivates the use of nonsmooth techniques to model frictional contacts.

Hence, the Coulomb friction law is commonly used for system-level simulation, due to its simplicity and its ability to capture the basic aspects of the dry friction process, as the capability to properly represent the slip and stick behavior. Therefore, it will be the one used in this work. A deep review and explanation of the Coulomb's friction laws can be found in Brogliato [24]. In the field of frictional contact mechanics and nonsmooth techniques, which are the core of this thesis, Pfeiffer and Glocker [3], Acary and Brogliato [27] and Leine and Wouw [28] present numerical methods for nonsmooth dynamical systems involving frictional contacts, including the necessary mathematical tools, such as, differential measures, set-valued functions, differential inclusions, functions of bounded variations and the Coulomb cone,

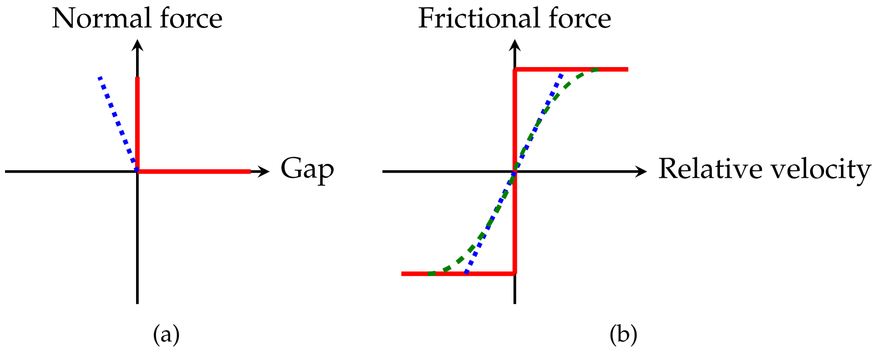


FIGURE 1.3: (a) Contact complementarity condition: red; Regularized contact complementarity condition: blue. (b) Frictional laws: Coulomb's law: red solid line; Continuous smooth law: green and blue dotted lines

which will be explained in Chapter 2.

1.5 Regularization techniques vs complementarity problems solver

Despite the effort done over the last decades for the development of numerical methods for mechanical systems involving frictional contacts, many open questions remain, which stimulates intense research on this subject in the computational mechanics and applied mathematics communities. The enforcement of the unilateral constraint of the contact/impact problem takes the form of a complementarity problem (CP). A CP can be solved using different approaches, such as iterative solvers, optimization methods and variational inequalities, or relaxing the complementarity condition with regularization techniques.

A commonly used iterative method in contact mechanics is the projected Gauss-Seidel method. This method partitions the system in different subdomains to compute them as independent entities, solving in a sequential manner each one of the CP. The main drawbacks of Gauss-Seidel methods are the slow convergence and the loss of symmetry of the solution for symmetric problems. It is inherently a sequential algorithm, which is not easily parallelized. Nevertheless, it is a popular method in the fields of computer graphics [29] and granular materials [30]. Different variations of this method have been proposed in order to improve different aspects of the method, for example [31, 32, 33, 34].

The fact that frictional contact problem can be posed as an optimization problem allows one to apply techniques well posed in the mathematical programming community, resulting in new numerical methods, such as the formulation of the problem as variational inequalities, generalized or semi–smooth equations, second–order cone complementarity problems, or as optimization problems such as quadratic programming problems over second-order cones. In a recent work [21], a review of the main numerical techniques that arise from these approaches was made, leading to the conclusion that there is no universal solver to fix all problems. Nevertheless, this paper gives some hints to choose the appropriate solver depending on the problem under study.

Regularization techniques relax the CP by allowing a certain amount of penetration, as illustrated in Fig. 1.3(a), the reaction force is not instantaneous anymore, but it increases progressively with the penetration amplitude. The slope of this model is the so called penalty coefficient. A smooth evolution of the displacement and velocity field is then observed, which permits the use of standard dynamics solvers. Also, the displacement is the only primary variable in the formulation, leading to a relatively easy numerical implementation. Therefore, this method can be easily introduced in an existing software. The main drawback is that the exact solution is only recovered for an infinite value of the penalty coefficient [35, 36, 8, 37, 38], and high penalty values lead to a set of stiff differential equations that produce ill-conditioned matrices and severe precision losses [39]. It is important to make a remark on the difference between the compliance models explained in a previous section and the regularization solvers presented here. Despite the similitude between the resulting equations of motion, here, the smoothness of the problem comes from a purely numerical strategy to smooth the CP without any relation to the physics of the problem, while in the compliance model the spring-damper parameters represent localized phenomena at the contact point, which sometimes can receive a real physical meaning. In contrast, in regularization techniques, the choice of the penalty factor is made by the user in order to obtain acceptable solutions that limit the unavoidable penetration between contacting bodies, but this choice bears some arbitrariness.

Lagrange multipliers method guarantee the exact fulfillment of the constraints overcoming the ill-conditioning inconvenience of regularization methods at the expense of an increase in the size of the system of equations [40, 4]. A combination of the penalty and the Lagrange multiplier techniques leads to the so-called augmented Lagrangian method. This method was proposed first by Hestenes [41] and Powell [42] to solve optimization problems with equality constraints and after applied to frictional contacts by several authors [43, 44, 45]. The augmented Lagrangian and the regularization methods, both involve an arbitrary penalty parameter. However, this penalty factor in the augmented Lagrangian method only affects the convergence rate, without causing any effect in the solution after reaching the convergence, whilst in the regularization method this factor directly influences the accuracy of the solution. For the Lagrange multiplier and augmented Lagrangian methods, the time evolution of the velocity is generally nonsmooth and the jumps are represented by an impact law, such as the Poisson or Newton impact laws [3], a revision of different impact laws can be found in [9]. In this work the augmented Lagrangian approach and the Newton impact law are used.

1.6 Nonsmooth time integrators

For systems involving nonsmooth phenomena, the common time integration methods used to study the finite element structural dynamics models, such as the Newmark method [46], the Hilbert-Hughes-Taylor method (HHT) [47] and the generalized-α method [48], can no longer be applied. The damped Newmark method is a first order method, while the HHT and the generalized-α methods are second order precision methods. These solvers can deal with stiff problems in structural dynamics with a broad frequency content [5]. These implicit integrators assume that the kinematic variables are smooth [49, 8]. However, in the presence of instantaneous changes of the velocity in case of impacts, these methods may produce numerical solutions with notable precision losses, a non physical behavior and the generation of fictitious energy at the contact instant [50]. Thus, nonsmooth time integration methods able to deal with nonsmooth motion equations are needed. Moreau was a pioneer in the development of a theoretical background for this class of problems [51, 52].

In nonsmooth dynamics, depending on the step size definition strategy, time integrators can be classified into two main groups: event-driven and time-stepping integrators [3, 27, 53, 54]. The former is based on the exact impact detection and the adaptation of the time step in order to capture the impact moment and restart the integration afterwards. Event-driven methods achieve an accurate solution, however, they become inefficient in situations involving a large number of impact events in short periods of time. Hence, a different strategy is adopted in this work, which falls under the category of time-stepping integrators. These techniques share the common feature that the time step is not adapted to impacts events, but that they are able to capture the effects of one or several impacts occurring during the time step. They are even able to capture the accumulation of an infinite number of impacts in a finite time interval. The most widespread time-stepping integrators for nonsmooth dynamical systems are the Schatzman–Paoli scheme [55, 56], which is based on a central difference scheme, and the Moreau–Jean scheme [57, 58, 59], which is based on a θ-method. Despite their robustness in dealing with problems involving a large number of impacts, the Schatzman–Paoli scheme and the Moreau–Jean lead to a first-order approximation of motion, and suffer from high levels of numerical dissipation which generally leads to poor approximations of vibration phenomena in flexible components. Indeed, when using a first-order method, a very small time step is required to get adequate results. Additionally, in the Moreau–Jean scheme, the constraints are only imposed at velocity level, which results in the violation of the constraints at position level and as a consequence, a drift phenomenon is observed.

Another alternative for simulating nonsmooth dynamic mechanical systems was proposed by Chen *et al.* [60], where some contributions, such as external, damping and internal forces are considered smooth and are integrated using a second-order scheme, i.e. the generalized-α method. However, the reaction forces of the bilateral constraints and the unilateral constraints are still integrated using a first order scheme, in a fully implicit approach. In comparison with the Moreau-Jean method the drift phenomenon for the bilateral constraints is reduced and the energy behavior is improved in the low-frequency range, allowing to use larger time steps. Nevertheless, globally the order of convergence of this algorithm is one and the drift phenomena for bilateral constraints are not completely eliminated.

More recently, in order to reduce the numerical dissipation and improve the energy behavior, especially for flexible system with impacts and vibrational effects, Brüls *et al.* in [50] extended the idea of Chen et al. and proposed the nonsmooth generalized-α scheme (NSGA) for frictionless contact problems. The main idea of this method is to artificially split the system into two parts. The smooth part (without consideration of the impact contributions) and the impulsive part (with impact contributions). This allows one to integrate the impulsive terms of the equations of motion with first-order accuracy, meanwhile the smooth contributions are integrated with second-order accuracy, even the reaction forces of the constraints. The generalized-α method was used to integrate the smooth motion, but other methods could be also possible. Thus, the vibration phenomena corresponding to the smooth parts of the motion can be computed with a smaller numerical dissipation than the Moreau scheme, improving the energy behaviour of the dynamic response of the flexible terms, as shown in [61].

Another advantage of this method is that the constraints are imposed at position and velocity levels, following a similar method as proposed by Gear, Gupta and Leimkuhler [62]. In this way no drift of the constraints is allowed, i.e., no unphysical penetration is observed.

Due to the splitting strategy and the imposition of constraints at position and velocity levels, the resulting equations of motion have a structure with three distinguishable subproblems. The first represents the smooth part which is integrated using the generalized-α scheme, the second represents a correction of the position variables which is computed to satisfy the constraints at position level due to the contact conditions and the third represents the velocity jump whose evaluation is based on the constraints at velocity level. These three parts are coupled with each other, therefore, it is not possible to solve them sequentially one after the other, and the system has to be solved at each time step using a monolithic semi-smooth Newton method, which has a high computational cost. To avoid the coupling, Brüls *et al.* [50] propose to neglect some coupling terms in the tangent matrix. The advantage of this procedure is that the algorithm can be described as a sequence of three sub-problems, instead of having to solve the complete set of equations monolithically. This approximation does not affect the final results, only the convergence of the method. It is fully justified as the adopted step size tends to zero. However, for problems with flexible bodies and nonlinear bilateral constraints, this approximation leads to a slow convergence of the global scheme, or even to the divergence of the scheme if a small step size is not small enough, which motivates further improvements of this solver in the context of this thesis.

1.7 Software review

Several software packages are available to model the dynamics of rigid and flexible multibody systems, like Adams, Simpack, Mecano or RecurDyn, that are commonly used in the industry. Some of them also include frictional contact models but are based on the regularization of the nonsmooth phenomena related to contact and friction. These techniques are easy to implement in an existing software but they also have important drawbacks, as discussed in the previous section. Therefore other possibilities are studied in this work.

Other software, such as, Siconos [63], LMGC90 [30], and Chrono Project [64], are scientific software packages which address contact problems using nonsmooth techniques. Siconos provides a general tool to face nonsmooth problems in various scientific fields in applied mathematics, e.g. mechanics, robotics or electrical circuits. LMGC90 is a multi-physics software which aims at modeling the interactions between objects of any shape with complex laws. It also allows to couple other effects, like thermal effects and fluid interaction. The Chrono project focuses on the simulation of the interaction between a large number of rigid bodies, using either nonsmooth or regularization techniques. These software packages mostly address the contacts between rigid bodies, except for the Chrono Project; which has been recently extended to model the contact between flexible beams using isogeometric analysis and nonsmooth techniques [65].

The simulation of MBS involving frictional contacts is not yet a mature and closed field, and nonsmooth methods arise as a relevant approach to solve these problems. Therefore, the idea of this work is to combine the finite element method (FEM) with the nonsmooth techniques to simulate frictional impact contacts between rigid and flexible bodies. The formulation presented in this thesis is implemented in the software Oofelie (Object Oriented Finite Elements Led by Interactive Executor). Oofelie is a multiphysics finite element software able to combine thermal, mechanic, acoustic and electric field problems [66], and it is integrated into the graphical environment Siemens NX.

1.8 Objective of the thesis

As presented in this introduction, reliable numerical methods are available to study smooth multibody dynamic systems. These techniques can model the behavior of flexible and rigid bodies, e.g. using a finite element approach, and they are well established in the literature and in commercial software packages.

When contact between rigid bodies occurs, the contact is an instantaneous phenomenon and the velocity field is no longer smooth, producing jumps in the velocity. In order to properly account for jumps in the velocity, nonsmooth techniques are needed. Existing nonsmooth techniques like the Moreau scheme, have limitations and drawbacks when applied to flexible systems. For example, this method induces high numerical dissipation, especially for the vibrating motion of flexible bodies. Also, the constraints are not imposed at position level yielding a drift of the constraints.

The nonsmooth generalized-α method reduces the numerical damping and imposes hard kinematic constraints removing the constraint drift. In [50] a nonsmooth version of this method was applied to problems involving rigid and flexible bodies with contacts, but only for frictionless problems. This solver suffers from a slow convergence when applied to stiff mechanical systems. Therefore, we can say, that a completely robust and systematic algorithm able to simulate MBS involving rigid and flexible bodies with nonsmooth frictional contacts, does not exist to date.

The general objective of this thesis is to propose a new methodology that combines the flexibility of the finite element approach and the nonsmooth techniques. This methodology should be able to deal systematically with frictional impacts involving rigid and flexible bodies. Specifically, this work focuses on the nonsmooth phenomenona caused by the frictional impacts. For that purpose, the nonlinear finite element approach for flexible MBS [5] is extended to contact problems involving friction and impacts. Some improvements of the nonsmooth generalized-α solver are proposed and are implemented in the general finite element software Oofelie.

The originality of the methodology introduced in this thesis lies in three essential aspects:

- A new formulation of the nonsmooth generelized-α method is proposed, where the splitting strategy has been modified so as to completely isolate the impulsive terms (created by impacts) from the smooth ones. Consequently, the three sub-problems that need to be solved at every time step are fully decoupled; a feature which improves considerably the robustness of the integrator for problems involving nonlinear bilateral constraints and flexible elements.

- Frictional effects are considered. This contribution is considered at the element level of the finite element code, without the necessity to modify the solver scheme. Therefore, a new frictional contact element is developed. For the sake of robustness and numerical performance, the frictional contact problem is treated using an augmented Lagrangian technique inspired by the work of Alart and Curnier [43] for quasi static problems.

- This formulation is developed using a finite element approach, and is implemented in the Oofelie software. This software has the advantage of providing a wide library of readily available elements like rigid bodies, flexible beams, springs, hinges and so on, so that a wide range of applications can then be addressed.

1.9 Overall structure of the thesis

The manuscript is structured as follows:

- Chapter 2: The key aspects of MBS dynamics are reviewed and the equations of motion of MBS involving frictional contacts are introduced. Also the origin of the nonsmoothness coming from frictional contacts and the formulation in terms of measures and inclusions are presented.

- Chapter 3: The time integration method used in this thesis is presented. It results in a new formulation for the nonsmooth generalized-α method with a different splitting strategy. This formulation accounts for the unilateral constraints dealing with friction both at position and velocity levels in order to avoid any drift. Additionally, the conceived numerical scheme should ensure that the convergence to the numerical solution is achieved with at least first-order global accuracy, which is tested in some numerical experiments.

- Chapter 4: The dynamic formulation of a frictional contact element is presented. An augmented Lagrangian technique is used following the methodology presented by Alart and Curnier [43]. Some numerical experiments are carried out and the results are compared with the analytical solution or some results found in the literature.

- Chapter 5: This chapter makes a short overview of the finite element software Oofelie, explaining in detail the implementation of the solver and the frictional contact element presented in Chapters 3 and 4, respectively. The case of the woodpecker toy is used to illustrate the use of the element library applied to MBS.

- Chapter 6: Finally, the conclusions of this work are presented, and possible ideas for further research are discussed.

1.10 Publications and conferences

This manuscript not only reflects the work done by the author, but also the work done in collaboration with Alberto Cardona, Alejandro Cosimo and Federico Cavalieri from the Centro de Investigación de Métodos Computacionales in Argentina, and the supervisor of the thesis Olivier Brüls from the Université de Liège. This collaboration gave as result the following publications.

- Chapter 3 is a reproduction of the journal paper:

  - Cosimo, A., Galvez, J., Cavalieri, F. J., Cardona, A., and Brüls, O., “A robust nonsmooth generalized-α scheme for flexible systems with impacts”. *Multibody System Dynamics*, 48, 127–149, 2020.

- Chapter 4 is an extension of the journal paper:

  - Galvez, J., Cosimo, A., Cavalieri, F. J., Brüls, O. and Cardona, A., “A nonsmooth frictional contact formulation for multibody system dynamics”. *International Journal for Numerical Methods in Engineering*, online since April 7, 2020.

- Chapter 5 is an extension of the conference publication:

  - Galvez, J., Cosimo, A., Cavalieri, F. J., Cardona, A. and Brüls, O., "A general purpose formulation for nonsmooth dynamics including large rotations: application to the woodpecker toy". *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Anaheim, United States, 18-21 August, 2019.

- Finally, a list of the presentation in conferences is given below:

  - Galvez, J., Cavalieri, F. J., Cosimo, A., Brüls, O. and Cardona, A., "Nonsmooth numerical solution of frictional contact problems in multibody systems". *ECCOMAS Multibody Dynamics Conference*, Duisburg, Germany, 15-18 July 2019
  - Galvez, J., Cardona, A., Cavalieri, F. J. and Brüls, O., "Nonsmooth-α time integration for frictional contact problems", *6th European Conference on Computational Mechanics*, Glasgow, United Kingdom, 11-15 June, 2018
  - Galvez J., Cardona A, Cavalieri F. and Brüls O., "An augmented Lagrangian frictional contact formulation for nonsmooth multibody systems", *European NonLinear Dynamics Conference*, Budapest, Hungary, 25-30 June 2017.
  - Cavalieri, F. J., Galvez, J., Brüls, O. and Cardona, A., "A spatial revolute joint model with clearance in mechanisms dynamics", *ECCOMAS Thematic Conference on Multibody Dynamics*, Prague, Czech Republic, 19-22 June, 2017.

Chapter 2

Multibody systems with contacts and impacts

In this chapter the key aspects of MBS dynamics are reviewed in order to introduce the equations of motion for multibody systems involving frictionless and frictional contacts. The developments of this chapter contain the main ideas that will be used for the original developments presented in Chapters 3 and 4.

This chapter is structured as follows. It starts reviewing the different possible coordinates systems and choosing the finite element coordinates for the development of this thesis. Then the equations of motion for constrained systems are presented following the principle of virtual work and contact models are introduced using set-value force laws. Then the non-smooth behavior of impacts between rigid bodies is explained, the decomposition of velocities and reaction forces in smooth and impulsive terms is introduced, and the equations of motion of MBS involving contacts are rewritten as differential measure inclusions. The extension to frictional contacts involves some concepts of convex analysis so that the frictional Coulomb law can be expressed as a differential inclusion. The equations of motion are formulated in a Lie group framework, therefore the structure of the $\mathbb{R}^3 \times S0(3)$ group is introduced and illustrated with the spinning top example. Finally the generalized-α scheme in a Lie group formulation is presented.

2.1 Coordinates system

The number of degrees of freedom (DOF) is the minimum necessary number of independent variables that fully describes the state of a system. The generalized coordinates are a set of variables that allows to completely describe the configuration of a system. In multibody dynamics, the configuration is the location and orientation of each element. The number of coordinates depends on the methodology used to describe the system, and is at least equal to the number of DOF. The main four techniques are: (1) minimal coordinates, (2) relative coordinates, (3) cartesian coordinates and (4) finite element coordinates. The minimal coordinates technique consists in choosing the minimum number of coordinates that are sufficient to describe the system. Using a model with a minimal set of coordinates has the advantage that the resulting equations of motion are compact and only unilateral constraints are involved. The drawbacks of this methodology are the lack of generality and the complexity of the kinematic description. Relative coordinates, also known as Lagrangian coordinates, describe the position and orientation of each element with respect to the previous link, which is particularly suitable for open kinematic chains, such as robotic arms. For open chains, this choice leads to a minimal set of coordinates but some kinematic constraints appear for systems with closed loops. Based on relative coordinates, the equations of motion can be derived according to a recursive Newton-Euler method. The cartesian coordinates technique relies on the position and orientation of the center of mass (COM) for each component. With this technique, the joints are modeled by constraints between the coordinates of the two connected bodies. This results in a large number of equations with a DAEs structure, which can be formulated in a systematic way. These three methods have the inconvenience that the extension to flexible bodies is not trivial. The global motion has to be split into rigid body motion and relative elastic deformation, resulting in a complex expressions of the kinetic energy [67]. Finally, the finite element coordinates is the most suitable method to model flexible bodies. According to this method, the coordinates are defined in a systematic way for each element and the redundant coordinates in the joints are eliminated by boolean identification. It is also quite easy to add nodes in any point of interest. This method uses a large number of coordinates and constraints, however, the equations of motion are sparse so that it is possible to use sparse numerical solvers to solve the problem efficiently.

The most appropriate coordinate system depends on the problem at hand, but when the objective is to develop a general tool able to deal with flexible and rigid bodies, the finite element coordinates appears as good choice. They also have the advantage that it can be embedded in a finite element code allowing to reuse the existing library of elements. Therefore, this method is selected in this work.

2.2 Equations of motion for constrained systems

The d'Alembert principle reformulates the second Newton's law introducing the definition of the so called inertial forces as the negative product of mass times acceleration. These forces are defined negative because they are in the opposite direction of the motion. Then, the dynamic equilibrium is expressed by the sum of the impressed forces and the inertial forces, $\boldsymbol{F} + \boldsymbol{F}^{inert} = \boldsymbol{0}$. The expression of the Newton's second law as an equilibrium of forces is relevant because it permits the application of the principle of virtual work as in the static case [68].

Let us study an unconstrained system. The virtual work is the total work done by all the forces acting on a mechanical system for a set of virtual displacements. The virtual work of a multibody system can be defined as:

$$\delta \mathcal{W} = \delta \mathcal{W}^{inert} + \delta \mathcal{W}^{ext} - \delta \mathcal{W}^{int} - \delta \mathcal{W}^{damp} \tag{2.1}$$

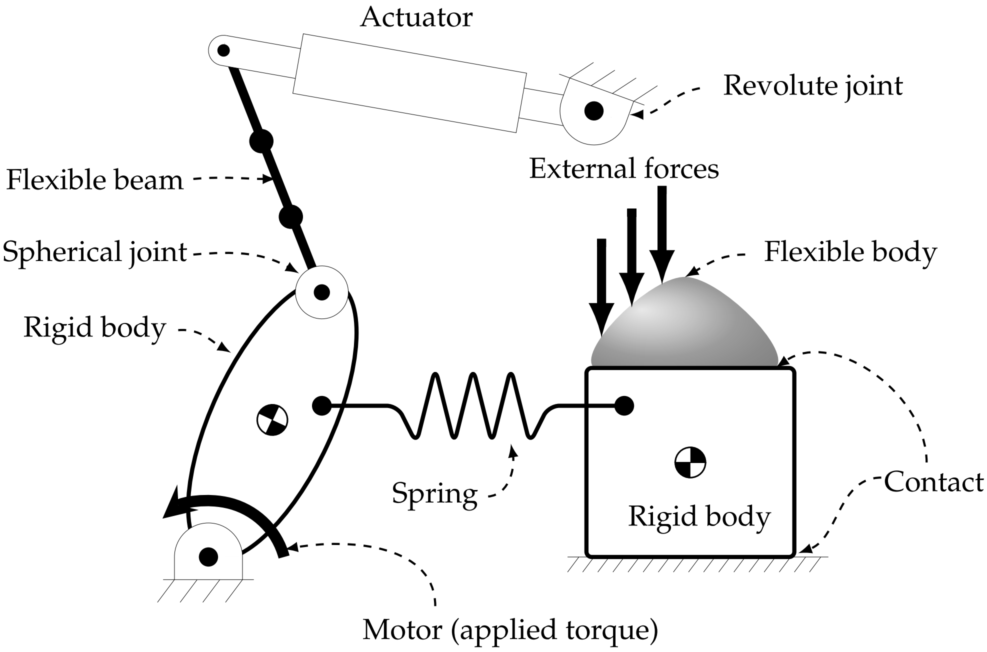


FIGURE 2.1: Example of a multibody system

where $\delta\mathcal{W}^{inert}, \delta\mathcal{W}^{ext}, \delta\mathcal{W}^{int}$ and $\delta\mathcal{W}^{damp}$ are the virtual works of the inertial, external, internal and damping forces, respectively. Assuming that the system is represented by the generalized coordinates $\boldsymbol{q}$, these virtual works are defined as:

$$
\begin{aligned}
\delta\mathcal{W}^{inert} &= -\delta\boldsymbol{q}^T(\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}^{gyr}(\boldsymbol{q},\dot{\boldsymbol{q}})) && (2.2a)\\
\delta\mathcal{W}^{ext} &= \delta\boldsymbol{q}^T\boldsymbol{f}^{ext}(\boldsymbol{q},t) && (2.2b)\\
\delta\mathcal{W}^{int} &= \delta\boldsymbol{q}^T\boldsymbol{f}^{int}(\boldsymbol{q}) && (2.2c)\\
\delta\mathcal{W}^{damp} &= \delta\boldsymbol{q}^T\boldsymbol{f}^{damp}(\boldsymbol{q},\dot{\boldsymbol{q}}) && (2.2d)
\end{aligned}
$$

where $\delta\boldsymbol{q}$ are virtual displacements, i.e., infinitesimal displacements which do not modify the forces in the system, $\boldsymbol{M}$ is the mass matrix, $\dot{\boldsymbol{q}}$ is the derivative of $\boldsymbol{q}$ with respect to time, $\ddot{\boldsymbol{q}}$ are the accelerations of the system, and $\boldsymbol{f}^{gyr}$, $\boldsymbol{f}^{ext}$, $\boldsymbol{f}^{int}$, $\boldsymbol{f}^{damp}$ are the gyroscopic, external, internal and damping forces of the system.

Then, replacing Eq. (2.2) in (2.1) the total virtual work can be expressed as:

$$\delta\mathcal{W} = \delta\boldsymbol{q}^T(-\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},t)) \tag{2.3}$$

where $\boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},t) = \boldsymbol{f}^{ext}(t) - \boldsymbol{f}^{damp}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{f}^{int}(\boldsymbol{q}) - \boldsymbol{f}^{gyr}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the sum of the external, damping, internal and gyroscopic forces.

The d'Alembert principle states that an unconstrained system is in dynamic equilibrium if and only if the virtual work vanishes for any arbitrary virtual displacements $\delta\boldsymbol{q}$:

$$\delta\mathcal{W} = 0 \quad \forall \quad \delta\boldsymbol{q} \tag{2.4}$$

Therefore, the equations of motion for an unconstrained system has the form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{2.5}$$

For constrained systems the constraint force $\boldsymbol{f}^c$ term appears. A bilateral constraint $\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}) = \boldsymbol{0}$ is ideal if, for any virtual displacement compatible with the constraints, the virtual work of the constraint forces vanishes. Therefore, the virtual work of the constraints forces satisfies

$$\delta\mathcal{W}^c = \delta\boldsymbol{q}^T\boldsymbol{f}^c = 0 \quad \forall \quad \delta\boldsymbol{q} \text{ satisfying } \boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q}}\delta\boldsymbol{q} = \boldsymbol{0} \tag{2.6}$$

where $\boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q}}$ is the constraint gradient matrix. Since the constraint forces do no make any virtual work, they do not appear in the d'Alembert principle.

For systems with workless constraints, using the Lagrangian multiplier methods, the equations of motion can be expressed as:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{v} && (2.7a)\\ \boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{v}} - \boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q}}(\boldsymbol{q})^T\boldsymbol{\lambda}^{\overline{\mathcal{U}}} &= \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) && (2.7b)\\ \boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}) &= \boldsymbol{0} && (2.7c) \end{aligned}$$

where $\boldsymbol{v}$ is the velocity vector and $\boldsymbol{\lambda}^{\overline{\mathcal{U}}}$ is the vector of Lagrange multipliers associated with constraints, which represents the reaction forces of the constraints. As the bilateral constraints are algebraic equations Eq. (2.7) is a DAE.

2.3 Multibody systems with contacts

The bilateral constraints are permanently active, and can be used to represent the restrictions imposed by a kinematic joint that connects different bodies. On the other hand, unilateral constraints can be active or inactive depending on the system state. For example, they may represent a non-penetration condition of a contact between bodies.

In a first approach, the unilateral constraints for a contact condition $\boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}) \geq \boldsymbol{0}$ are presented assuming that no impact occurs in the system. It means that detachment phenomena may occur during the motion or the velocities are small enough to neglect any impact contribution. Under this assumption the equations of motion are expressed as:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{v} && (2.8a)\\ \boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{v}} - \boldsymbol{g}_{\boldsymbol{q}}(\boldsymbol{q})^T\boldsymbol{\lambda} &= \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) && (2.8b)\\ \boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}) &= \boldsymbol{0} && (2.8c)\\ 0 \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}) &\perp \boldsymbol{\lambda}^{\mathcal{U}} \geq 0 && (2.8d) \end{aligned}$$

where Eq. (2.8d) represents the complementarity condition between the two terms, $\boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q})$ and $\boldsymbol{\lambda}^{\mathcal{U}}$. The unilateral constraints are also ideal in that case [69].

The total set of constraints includes two sets, the set of unilateral constraints $\mathcal{U}$, and its complementary set $\overline{\mathcal{U}}$ which is the set of bilateral constraints. In that way we have:

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}^{\mathcal{U}} \\ \boldsymbol{g}^{\overline{\mathcal{U}}} \end{bmatrix}, \qquad \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}^{\mathcal{U}} \\ \boldsymbol{\lambda}^{\overline{\mathcal{U}}} \end{bmatrix} \tag{2.9}$$

A complementarity condition for two positive variables a and b is represented by the symbol $0 \leq a \perp b \geq 0$ and it imposes $a \geq 0$, $b \geq 0$ and $ab = 0$. In the field of contact mechanics the complementarity condition in Eq. (2.8d) is known as Hertz - Signorini - Moreau condition [8], and is illustrated in Fig. 2.2. It models the physical behavior of a frictionless contact, ensuring that interacting bodies do not penetrate and allowing only repulsive forces between them. The first concept is represented by $g^j(\boldsymbol{q}) \geq 0$ meaning that the gap between the bodies is always greater than zero, and the second one by $\lambda^j \geq 0$, where the super index j indicates each one of the contacts of the system and λ^j is the reaction force between the bodies, which should be positive if we assume that the contact force is always repulsive. Finally, the contact law relating the gap g^j and the reaction forces λ^j is not represented by a function allowing to compute the force from the gap in an unidirectional way. In contrast, it is a set-valued law, in which one value of g^j can map to several values of λ^j depending on the state of the system, and both variables should always satisfy $g^j(\boldsymbol{q})\lambda^j = 0$. As can be seen in Fig. 2.2, when the contact is active $g^j = 0$ and the reaction force can take any value between zero and infinite. A deeper explanation about set-valued functions can be found in the book of Leine and van de Wouw [28].

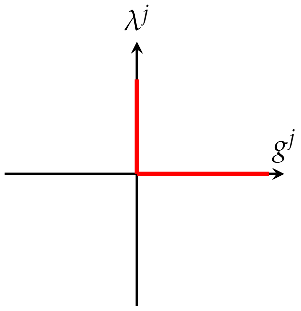


FIGURE 2.2: Contact complementarity condition

If the constraints are satisfied at position level for a finite interval of time, their first and second derivative are also zero. The derivative of the bilateral constraints at position level in Eq. (2.8c) has the expression:

$$\frac{\mathrm{d}\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}(t))}{\mathrm{d}t} = \boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{v} \tag{2.10}$$

The equations of motion in Eq. (2.8) can be written with the constraints formulated at velocity level as:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{v} && \text{(2.11a)} \\ \boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{v}} - \boldsymbol{g}_{\boldsymbol{q}}(\boldsymbol{q})^T\boldsymbol{\lambda} &= \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) && \text{(2.11b)} \\ \boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}(\boldsymbol{q})\boldsymbol{v} &= \boldsymbol{0} && \text{(2.11c)} \\ \text{if } g^j(\boldsymbol{q}) \leq \boldsymbol{0}; \quad \text{then} \quad 0 \leq \boldsymbol{g}_{\boldsymbol{q}}^j(\boldsymbol{q})\boldsymbol{v} &\perp \boldsymbol{\lambda}^j \geq 0, \quad \forall j \in \mathcal{U} && \text{(2.11d)} \end{aligned}$$

Equation (2.11d) reflects that the unilateral constraint at velocity level should only be considered when the contact is closed at position level, i.e., when $\boldsymbol{g}^j(\boldsymbol{q}) \leq \boldsymbol{0}$. For the case of systems without impact it is possible to formulate the equations of motion imposing the constraints at acceleration level, see e.g., Brüls et al. [10]. But in the cases of nonsmooth systems with impacts it is not possible to express the equations of motion with the constraints at acceleration level because the acceleration is not defined at the impact instants.

2.4 Impacts between rigid bodies

In Fig. 2.3 the collision of two rigid bodies is represented, where g is the gap distance between the two bodies and $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ are the velocities of the bodies A and B, respectively. When the two bodies collide the impact occurs instantaneously, which means that it occurs in an infinitesimal amount of time. At the impact instant t_i, the gap g becomes zero and a reaction impulse p_i appears which modifies the velocities in order to prevent any penetration. The impulse of the reaction force due to the impact at the impact time t_i is defined as:

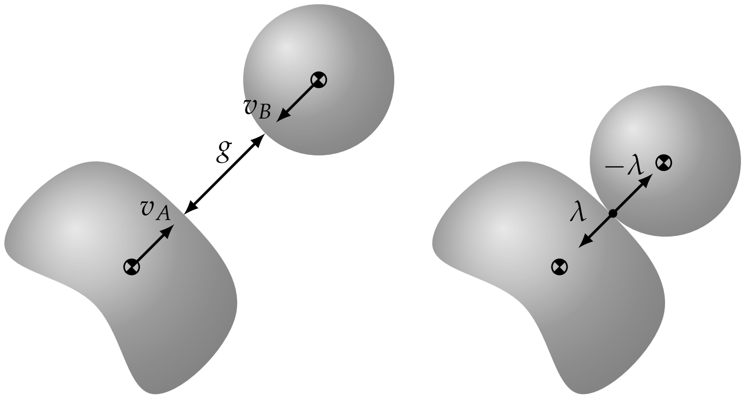
$$p_i = \lim_{\Delta t \to 0} \int_{[t_i, t_i + \Delta t)} \lambda(\tau) d\tau \tag{2.12}$$

where $\lambda(\tau)$ represents the force during the collision, which is zero outside the time interval $[t_i, t_i + \Delta t)$, and Δt is the impact duration. Since the duration of the collision is infinitesimal, it means that $\Delta t \to 0$. The Lebesgue integral of Eq. (2.12) should be zero, unless $\lambda(\tau)$ takes infinite values. It follows that $\lambda(\cdot)$ is not a classical function of time, but that the theory of distributions or measures should be used. For example, the Dirac measure at time t_i, denoted as δ_{t_i}, satisfies $\int_{[t_i, t_i+\Delta t)} \delta_{t_i} = 1$. We can then define $p_i = \int_{[t_i, t_i+\Delta t)} p_i \delta_{t_i}$ [24]. A consequence of this approach is that the impacts imply a discontinuity in the velocity. These discontinuities are modeled with an impact law, which relates the pre- and post- system state with a coefficient of restitution (COR). The most common are the Newton and Poisson impact laws. The first is a kinematic law since its COR relates the velocities before and after the contact, while the second is a kinetic law because its COR relates the impulses [24]. In literature, one also refers to an energetic COR that relates the kinetic energies, before and after the contact. However, Pérès [70] proved for the frictionless case that the kinetic and energetic CORs are equal.

In this thesis, the Newton's impact law [24] is used to describe frictionless impacts between rigid bodies. It is written as:

$$\overset{\circ}{g}^{j} = g^{j}_{\boldsymbol{q}}(\boldsymbol{q}(t_i))\boldsymbol{v}(t_i) + e^{j} g^{j}_{\boldsymbol{q}}(\boldsymbol{q}(t_i))\boldsymbol{v}^{-}(t_i) = 0 \quad \forall j \in \mathcal{U} \tag{2.13}$$

where $e^j \in [0,1]$ is the coefficient of restitution, $\boldsymbol{v}^-(t_i)$ is the pre-impact velocity and $\boldsymbol{v}(t_i)$ is the post-impact velocity. This coefficient is the ratio between the post-impact and the pre-impact velocity for a given contact. It can not be greater than 1 because no energy can be generated during the impact. In the case of a collision between rigid bodies this coefficient represents the dissipated energy during the impact. To formulate a dynamic contact problem for flexible bodies, an impact law is not needed. In this case, the duration of the contact is finite. After space and time discretization, a coefficient of restitution is sometimes introduced to control the numerical response. The physical meaning of the COR in this context is a delicate question, and the common approach is to use $e^j = 0$ to impose the constraints at velocity level $\boldsymbol{g}^j_{\boldsymbol{q}}(\boldsymbol{q}(t_i))\boldsymbol{v}(t_i) = 0$ when the constraint is active [71]. When the bodies enter in contact the kinetic energy of the elements of the meshes at the contact surface is dissipated. This means that the amount of dissipated energy is proportional to the size of the elements and will converge to zero with mesh refinement. The restitution coefficient is an experimental coefficient which depends on several variables, and not only the material. For example two rigid spheres colliding will have different restitution coefficient than a cube colliding a surface even if they have the same material. The most influent variables are the bodies shapes, the size, the masses, the elastic modulus and the material dissipation parameters [24].



(a)

(b)

FIGURE 2.3: Contact between two rigid bodies: (a) Before the contact. (b) Instant of the contact.

This velocity discontinuity can be illustrated in the simple example of the bouncing ball. It consists of a rigid ball of mass m falling vertically due to the gravity a_g and bouncing against a rigid surface. The impact process is modeled using the

Newton's impact law with a COR $e = 0.8$. If the velocity is analyzed in an infinitesimal amount of time, $\Delta t \rightarrow 0$, which captures the collision instant, it is observed that the direction of the velocity has been inverted. The difference between the pre- and post- impact velocities is the aforementioned discontinuity called velocity jump. The position and velocity of the bouncing ball are shown in Fig. 2.4. The position is continuous but not differentiable at the instants where the contact occurs, unlike the velocity, which shows instantaneous jumps in these instants. Then the reaction force at these points, to be compatible with the representation of the velocity jumps, is an impulse which is described as a Dirac measure. In this example, the ball stabilizes itself in the closed contact state after a finite duration, but an infinite number of impacts events occurs. This phenomenon is an impact accumulation phenomenon. The equations of motion for this one degree-of-freedom example are expressed as follows. If there is no impact, they take the form

$$\dot{q} = v \tag{2.14a}$$

$$m\dot{v} + \lambda = -ma_g \tag{2.14b}$$

$$0 \leq q \perp \lambda \geq 0 \tag{2.14c}$$

and at the impact time t_i, they become

$$m(v_i^+ - v_i^-) + p_i = 0 \tag{2.15a}$$

$$\text{if } q \leq 0; \text{ then } 0 \leq v_i^+ + ev_i^- \perp p_i \geq 0 \tag{2.15b}$$

where q is the position of the ball in the vertical axis, v the vertical velocity of the ball, a_g the gravity acceleration and the subindex i identify the impact instants. Equations (2.14a-2.14c) represent the smooth part of motion and Eqs. (2.15a, 2.15b) the impact motion, where Eq. (2.15a) is the balance of impulses and Eq. (2.15b) is the Newton's impact law.

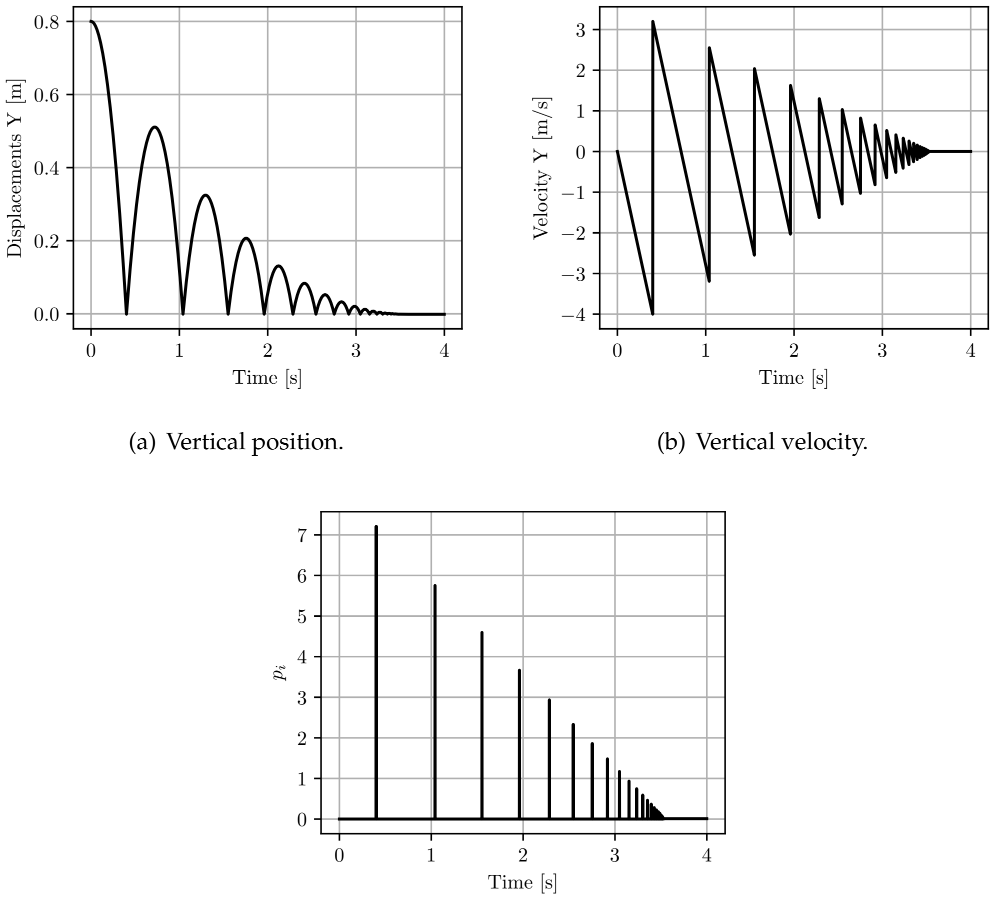
2.5 Multibody systems with impacts

In this section the formulation of the equations of motion of a multibody system is extended to deal with impact phenomena. In the impact instants the velocity field undergoes instantaneous changes, which means that it is no longer continuous and differentiable and it presents some jumps. Therefore, it makes sense to split the equations of motion in two sets of equations. The first one accounts for the motion for almost every time t,

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} - \boldsymbol{g}_{\boldsymbol{q}}^T \boldsymbol{\lambda} \quad = \quad \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) \tag{2.16a}$$

$$\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}) \quad = \quad \boldsymbol{0} \tag{2.16b}$$

$$\boldsymbol{0} \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}) \quad \perp \quad \boldsymbol{\lambda}^{\mathcal{U}} \geq \boldsymbol{0} \tag{2.16c}$$



(a) Vertical position.

(b) Vertical velocity.

(c) Reaction impulses normal to the surface.

FIGURE 2.4: Bouncing ball example. Initial height $h = 0.8$ m, mass $m = 1$ kg, COR $e = 0.8$ and $a_g = 9.8$ m/s^2.

and the second represents the contribution due to impacts at time t_i

$$\boldsymbol{M}(\boldsymbol{q})\,(\boldsymbol{v}(t_i) - \boldsymbol{v}^-(t_i)) - \boldsymbol{g}_{\boldsymbol{q}}^T\,\boldsymbol{p}_i \quad = \quad \boldsymbol{0} \tag{2.16d}$$

$$\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}\,\boldsymbol{v}(t_i) \quad = \quad \boldsymbol{0} \tag{2.16e}$$

$$\text{if } g^j(\boldsymbol{q}(t_i)) \leq 0 \text{ then } 0 \leq g_{\boldsymbol{q}}^j\,\boldsymbol{v}(t_i) + e^j\,g_{\boldsymbol{q}}^j\,\boldsymbol{v}^-(t_i) \quad \perp \quad p_i^j \geq \boldsymbol{0}, \quad \forall j \in \mathcal{U} \tag{2.16f}$$

Due to the non differentiability of the velocity field, the acceleration is not well defined in the usual sense and the dynamics can no longer be represented by ODE or DAE. Instead formulations in terms of measures can be used as suggested by Moreau [58]. The main advantage of formulations in measures is that they can account for jumps in the state evolution. A detailed explanation of the mathematical background behind the theory of measures and its application to contact dynamics can be found in [24, 28, 72], nevertheless a review of the concepts of absolutely continuous function, step function and function of locally bounded variation are reviewed to ease the understanding of the next section.

Definition 1. *(Absolutely continuous function).*

A function $\boldsymbol{f} : \boldsymbol{I} \to \mathbb{R}^n$ is said to be absolutely continuous on a real interval $\boldsymbol{I}$ if, for every $\epsilon > 0$, there exists $\delta > 0$ such that,

$$\sum_{i=1}^{m} \|\boldsymbol{f}(b_i) - \boldsymbol{f}(a_i)\| < \epsilon \tag{2.17}$$

for any m and every finite collection of pairwise disjoint intervals $[a_i, b_i] \in \boldsymbol{I}$ satisfying

$$\sum_{i=1}^{m} (b_i - a_i) < \delta \tag{2.18}$$

These functions have the property that they can be obtained from integration of their derivative, which exists almost everywhere [28]. The absolute continuity property is easier to understand showing a counter example. Figure 2.5(a) shows $\boldsymbol{f} = 1/2x$ on $\boldsymbol{I} = [0, x_{max}]$. For a given ϵ and $m = 1$, it is not possible to find a δ such that Eq. (2.17) is satisfied on any subinterval satisfying Eq. (2.18)

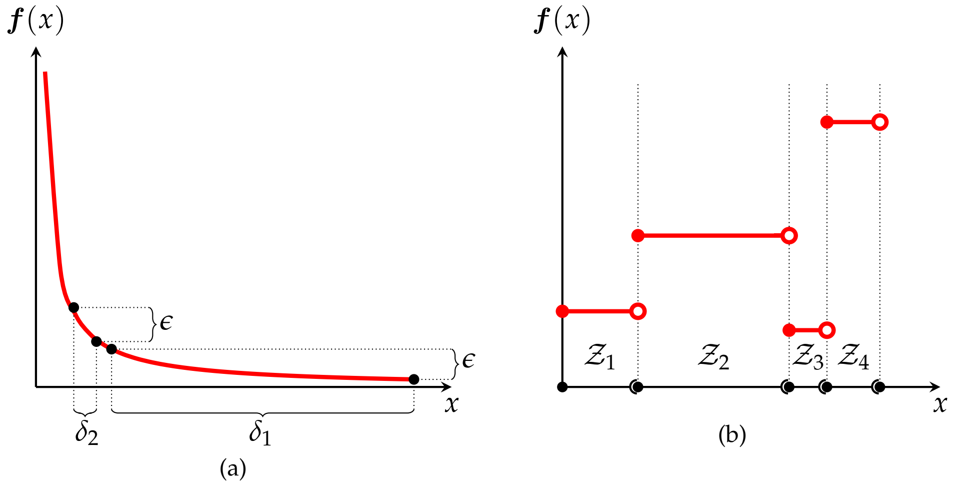


FIGURE 2.5: (a) Example of non absolutely continuous function.
(b) Example of step function.

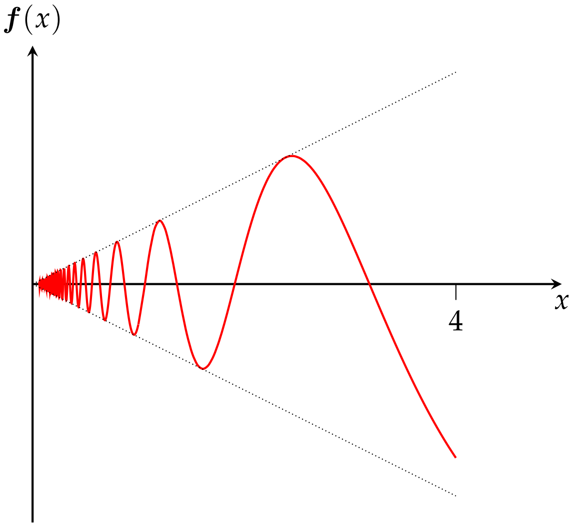
Definition 2. *(Step function).*

A step function is constant in almost its full domain, and is composed of different constant functions as seen in Fig. 2.5(b). More formally, a function $\boldsymbol{f} : \boldsymbol{I} \to \mathbb{R}$, which takes a finite number of values on the domain $\boldsymbol{I}$ is called a step function, such that

$$\boldsymbol{f}(x) = \sum_{i=1}^{m} \alpha_i \Psi_{\mathcal{Z}_i}(x) \tag{2.19}$$

where $m \in \mathbb{N}$, $\alpha_i \in \mathbb{R}$ and the indicator function $\Psi_{\mathcal{Z}_i} : \boldsymbol{I} \to \mathbb{R}$, of a set $\mathcal{Z}_i \in \mathbb{R}$ is defined as:

$$\psi_{\mathbb{Z}_i}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{Z}_i \\ 0 & \text{otherwise} \end{cases} \tag{2.20}$$



$\boldsymbol{f}(x) = x\,cos(15/x)$ for $x \in (0,4]$

FIGURE 2.6: Example of a function which is not of locally bounded variation

Definition 3. *(Function of locally bounded variation).*

A function $\boldsymbol{f} : \boldsymbol{I} \to \boldsymbol{X}$ is of locally bounded variation if and only if

$$\sup \sum_{i=1}^{n} \|\boldsymbol{f}(x_i) - \boldsymbol{f}(x_{i-1})\| < \infty \tag{2.21}$$

for every compact subinterval [a,b], where the supremum is taken over all strictly increasing finite sequences $x_1 < x_2 < ... < x_n$ of points on [a,b]. In the case of a function with one variable, being of locally bounded variation can be visualized as the fact that the distance in the vertical axis direction traveled by a point that follows the graph takes a finite value. In Fig. 2.6 a counter example of locally bounded variation function is presented.

Every function of locally bounded variation f can be decomposed as the sum of three different functions of locally bounded variation as:

$$f = f_{abs} + f_{step} + f_{sing} \tag{2.22}$$

where:

- f_{abs} is absolutely continuous.

- f_{step} is a step function which is constant almost everywhere.
- f_{sing} is singular, which means it is a nonconstant continuous function with a derivative which is zero almost everywhere [28].

As seen in Section 2.4 the impact between rigid bodies creates jumps in the velocity field. Due to these jumps, the accelerations and forces are not defined at the instant of the impacts. This problem can be illustrated by checking the step function s shown in Fig. 2.7(c), which has a unit jump for $t = 0$. Its derivative is zero for almost all time except at the instant $t = 0$, where it is infinite as in Fig. 2.7(d). Fig. 2.7(a) shows the function $s_\epsilon(t)$ which approximates $s(t)$ when $\epsilon \to 0$ and Fig. 2.7(b) shows its derivative $\mathrm{d}s_\epsilon/\mathrm{d}t$. Therefore, the derivative of the step function s is defined as the Dirac function δ_0 defined as:

$$\delta_0 = \begin{cases} +\infty & \text{if} \quad x = 0 \\ 0 & \text{if} \quad x \neq 0 \end{cases}, \quad \int_{-\infty}^{\infty} \delta_0(t)dt \quad = \quad 1 \tag{2.23}$$

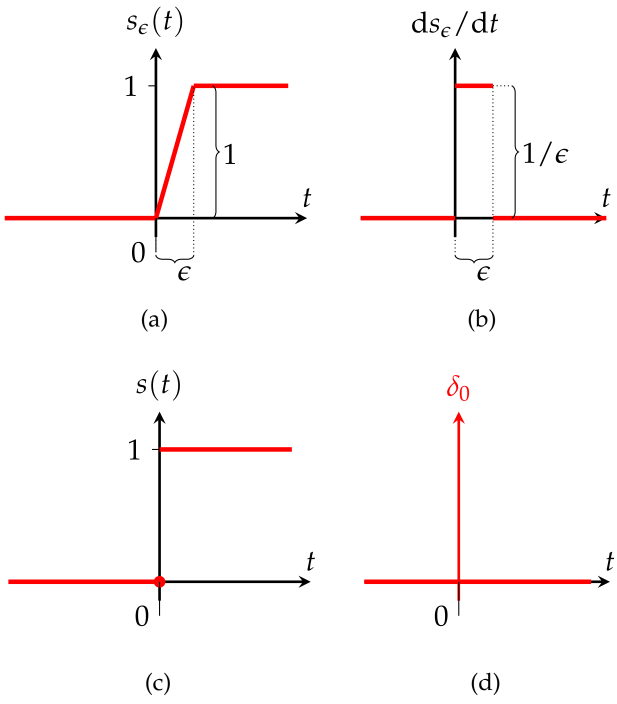


FIGURE 2.7: (a) $s_\epsilon(t)$ function, $\epsilon \geq 0$. (b) Derivative of $s_\epsilon(t)$ function, $\epsilon \geq 0$. (c) Step function $s(t)$. (d) Derivative of $s(t)$, Dirac function.

Assuming that the velocity is a function of bounded variation, analogously to Eq. (2.22) it can be decomposed as:

$$v = v_{cont} + v_{step} + v_{singular} \tag{2.24}$$

where v_{cont} represent the smooth part of the motion and v_{step} is the nonsmooth part which represents the impacts effects. Finally, $v_{singular}$ is considered as zero, as it is usually done when mechanical systems are studied [28, 72]. This decomposition is illustrated in Fig. 2.8.

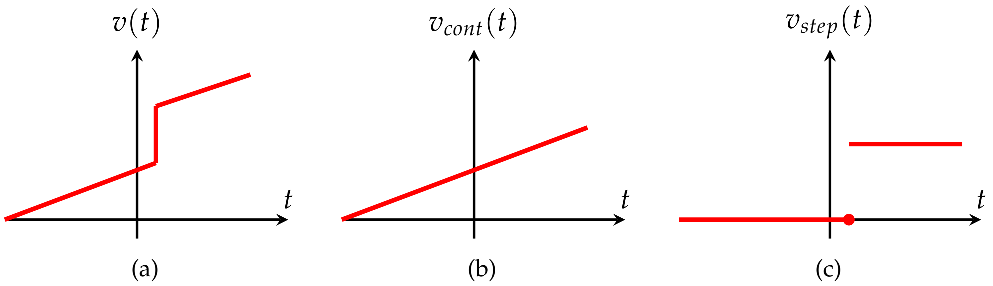


FIGURE 2.8: Velocity decomposition

As the accelerations and forces are not defined in the case of an impact, their description is replaced by the measures $\mathrm{d}\boldsymbol{v}$ and $\mathrm{d}\boldsymbol{i}$, respectively. First, it is necessary to introduce the right and left limits, which always exist for a bounded variation function:

$$\dot{\boldsymbol{q}}^{+}(t) = \lim_{\tau\to t, \tau>t} \dot{\boldsymbol{q}}(\tau), \qquad \dot{\boldsymbol{q}}^{-}(t) = \lim_{\tau\to t, \tau<t} \dot{\boldsymbol{q}}(\tau) \tag{2.25}$$

$$\boldsymbol{v}^{+}(t) = \lim_{\tau\to t, \tau>t} \boldsymbol{v}(\tau), \qquad \boldsymbol{v}^{-}(t) = \lim_{\tau\to t, \tau<t} \boldsymbol{v}(\tau) \tag{2.26}$$

For simplicity $\boldsymbol{v}(t)$ and $\dot{\boldsymbol{q}}(t)$ will be used to denote $\boldsymbol{v}^{+}(t)$ and $\dot{\boldsymbol{q}}^{+}(t)$, respectively.

The measure $\mathrm{d}\boldsymbol{v}$ admits the decomposition

$$\mathrm{d}\boldsymbol{v} = d\boldsymbol{v}_{cont} + d\boldsymbol{v}_{step} \quad = \quad \dot{\boldsymbol{v}}\,\mathrm{d}t + \sum_i (\boldsymbol{v}(t_i) - \boldsymbol{v}^{-}(t_i))\,\delta_{t_i} \tag{2.27}$$

where $\mathrm{d}t$ is the Lebesgue measure, i represents the index of a specific impact, δ_{t_i} is the Dirac delta supported at t_i and since $\boldsymbol{v}_{singular}$ is assumed to be zero also its measure $d\boldsymbol{v}_{singular} = 0$. In the same way the measure $\mathrm{d}\boldsymbol{i}$ which represents the impulse of the reaction forces is decomposed as

$$\mathrm{d}\boldsymbol{i} \quad = \quad \boldsymbol{\lambda}\,\mathrm{d}t + \sum_i \boldsymbol{p}_i\,\delta_{t_i} \tag{2.28}$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier vector which is associated to the smooth motion and $\boldsymbol{p}_i$ is the impulse of the reaction force produced due to the impact at time t_i. The equations of motion can be written in terms of these measures with the constraints imposed at position level:

$$\dot{\boldsymbol{q}} = \boldsymbol{v} \tag{2.29a}$$

$$\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{v} - \boldsymbol{g}_{\boldsymbol{q}}(\boldsymbol{q})^T \mathrm{d}\boldsymbol{i} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t)\mathrm{d}t \tag{2.29b}$$

$$\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}) = \boldsymbol{0} \tag{2.29c}$$

$$0 \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}) \perp \mathrm{d}\boldsymbol{i}^{\mathcal{U}} \geq 0 \tag{2.29d}$$

Finally, to complete the equations of motion, an impact law is needed to specify the post-impact velocity. In this work the aforementioned Newton's impact law is used. As proposed by Moreau [73], this impact law can be embedded in the equations of motion imposing the constraints at velocity level, which results in the final equations of motion used to represent the multibody system involving contacts and impacts as:

$$\begin{aligned}
\dot{\boldsymbol{q}} &= \boldsymbol{v} && \text{(2.30a)}\\
\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{v} - \boldsymbol{g}_{\boldsymbol{q}}(\boldsymbol{q})^T\mathrm{d}\boldsymbol{i} &= \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t)\mathrm{d}t && \text{(2.30b)}\\
-\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q})\boldsymbol{v} &= \boldsymbol{0} && \text{(2.30c)}\\
\text{if } g^j(\boldsymbol{q}) \leq 0; \quad \text{then} \quad 0 \leq g^j_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{v} + e^j g^j_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{v}^- &\perp \mathrm{d}i^j \geq 0, \quad \forall j \in \mathcal{U} && \text{(2.30d)}
\end{aligned}$$

The Moreau's viability lemma ensures that the solution to Eq. (2.30) with the constraints at velocity level also satisfies the constraints at position level, in Eq. (2.29c, 2.29d). A more detailed explanation of this lemma can be found in [24, 27].

The sign of the bilateral constraints is an arbitrary choice, and the minus sign is introduced in Eq. (2.30c) because in that way, for the frictionless cases, the tangent matrix is symmetric as will be seen later.

2.6 Differential inclusions

Convex analysis provides a series of useful tools to deal with set-valued functions and express them as differential inclusions. In what follows, a brief explanation of the key concepts used in this thesis is presented.

The indicator function, normally symbolized by $\psi_{\mathcal{C}}$ denotes the membership of an element to a set $\mathcal{C}$. For a value x_1 belonging to the set, $x_1 \in \mathcal{C}$, the indicator function is zero, $\psi_{\mathcal{C}}(x_1) = 0$. If the value x_1 does not belong to the set, $x_1 \notin \mathcal{C}$, the indicator function is infinity, $\psi_{\mathcal{C}}(x_1) = \infty$, as shown in Fig. 2.9. The contact complementarity condition, Fig. 2.2, can be defined using the indicator function of the real half line $\mathbb{R}^+$, denoted by $\psi^+_{\mathbb{R}}$, and its subdifferential function $\partial\psi_{\mathbb{R}^+}$, represented in Fig. 2.10. The function $\psi_{\mathbb{R}^+}$ takes the following form:

$$\psi_{\mathbb{R}^+}(x) = \begin{cases} 0 & \text{if} \quad x \geq 0 \\ \infty & \text{if} \quad x < 0 \end{cases} \tag{2.31}$$

which means that only real positive values belong to the set $\mathbb{R}^+$. Its subdifferential has the expression of the following set-valued function:

$$\partial\psi_{\mathbb{R}^+}(x) = \begin{cases} 0 & \text{if} \quad x > 0 \\ (-\infty, 0] & \text{if} \quad x = 0 \\ \emptyset & \text{if} \quad x < 0 \end{cases} \tag{2.32}$$

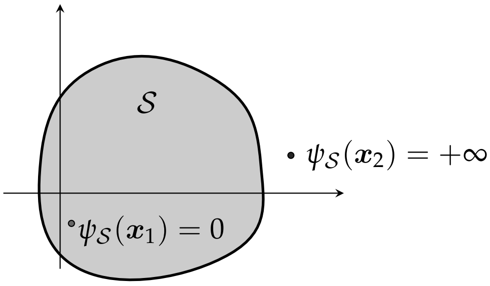
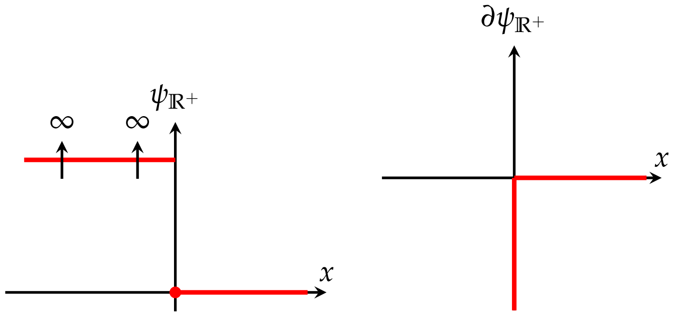

$\mathcal{S}$
$\bullet\ \psi_{\mathcal{S}}(\boldsymbol{x}_2) = +\infty$
$\bullet\psi_{\mathcal{S}}(\boldsymbol{x}_1) = 0$

FIGURE 2.9: Example of indicator function values of two points for a set $\mathcal{S} \in \mathbb{R}^2$.



(a)

(b)

FIGURE 2.10: (a) Indicator function $\psi_{\mathbb{R}}^+$. (b) Subdifferential $\partial\psi_{\mathbb{R}}^+$.

This set-valued function can be used to define the contact behavior in the normal direction. Indeed, the set-valued function in Fig. 2.2 can be written as an inclusion $-\lambda^j \in \partial\Psi_{\mathbb{R}^+}(g^j)$ or $-g^j \in \partial\Psi_{\mathbb{R}^+}(\lambda^j)$. Similarly, the complementarity condition in Eq. (2.30d) can be written as measure differential inclusion [58]$-\mathrm{d}i_N^j \in \partial\Psi_{\mathbb{R}^+}(\overset{\circ}{g}{}_N^j)$ if $g^j \leq 0$. If the gap in the normal direction is zero $g_N^j = 0$ and $\overset{\circ}{g}{}_N^j > 0$, the reaction impulse should be zero $\mathrm{d}i_N^j = 0$. If $g^j = 0$ and $\overset{\circ}{g}{}_N^j = 0$, the reaction impulse should be positive $\mathrm{d}i_N^j \geq 0$. The case $\overset{\circ}{g}{}_N^j < 0$ is not allowed because $\partial\Psi_{\mathbb{R}^+}(\overset{\circ}{g}{}_N^j)$ is the empty set.

2.7 Friction

As mentioned in Section 1.4, the Coulomb's friction law is commonly adopted to model multibody systems involving dry frictional contacts due to its simplicity and the capability to represent the macroscopic behavior of the system, accounting both for slip and stick phenomena. In this section, impacts are not considered and the equations of motion (2.30) are adapted to non-impulsive frictional contact problems adopting the Coulomb's law. In what follows, a brief explanation of the key concepts to express the Coulomb's friction law as an inclusion is presented, a more detailed mathematical presentation can be found in Glocker [54] , Pfeiffer and Glocker [3], Leine and van de Wouw [28] and Studer [72].

As introduced in Section 1.4 the Coulomb friction law defines the frictional force in terms of the normal force and the friction coefficient. In this law the frictional force is a nonsmooth function represented by the set-valued function shown in Fig. 2.11(a). To model the frictional contact phenomenon, the frictional force behavior is split in two different cases, stick and slip. The stick condition occurs when there is no relative displacement between the two bodies in contact and therefore the frictional force is bounded between zero and the maximum frictional force, which is $\|\boldsymbol{\lambda}_{T_{max}}\| = \mu\|\lambda_N\|$. One the other hand the slip phenomenon occurs when sliding exists between the two surfaces and in this case the frictional force is maximum, $\mu\|\lambda_N\|$.

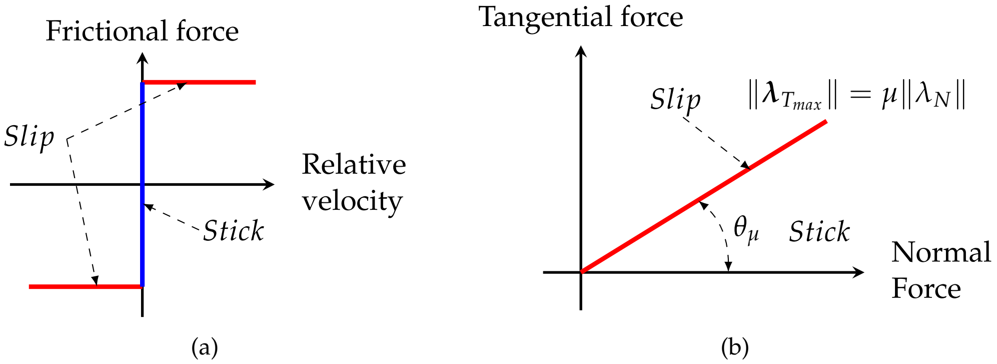
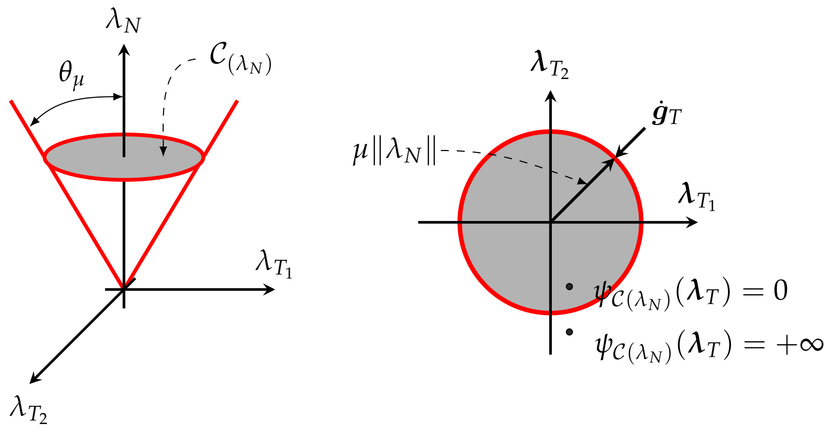


FIGURE 2.11: (a) Coulomb's frictional law (b) Maximum frictional force

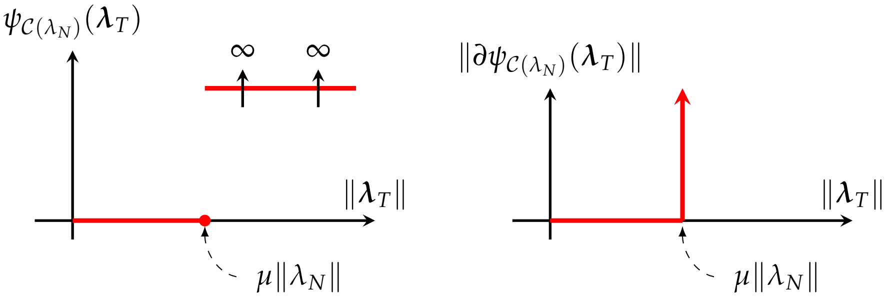
Fig. 2.11(b) shows the representation of the Coulomb law to discriminate between stick and slip contact status for the two dimensional case, where the angle of the maximum frictional force θ_μ is defined as $\mu = \tan\theta_\mu$. In spatial cases, Coulomb law is represented in the 3-dimensional Euclidean space by the so called frictional cone, consisting of an inverted cone at the origin, as depicted in Fig. 2.12(a).As in the two dimensional case the tangent of the cone angle θ_μ is the friction coefficient μ. The frictional cone surface denoted by $\mathcal{C}$ and the sticking condition is written $\|\boldsymbol{\lambda}_T\| < \mu\lambda_N$. Then, if the contact reaction is on the cone surface, the contact is in slip mode and the sliding velocity is in the opposite direction of the frictional force. The possibility that the reaction force lays outside the cone is non-physical.

For an isotropic Coulomb law the set $\mathcal{C}_{(\lambda_N)}$ is defined as a disc centered at origin with a radius equal to $\mu\lambda_N$, as shown in Fig. 2.12(b). In this figure, the tangential sliding velocity $\dot{\boldsymbol{g}}_T$ is represented and is aligned with the frictional force but in the opposite direction. The indicator function $\psi_{\mathcal{C}_{(\lambda_N)}}(\boldsymbol{\lambda}_T)$ indicates if $\boldsymbol{\lambda}_T$ belongs or not to the disc $\mathcal{C}_{(\lambda_N)}$, and since $\mathcal{C}_{(\lambda_N)}$ is a disc belonging to the Coulomb cone $\mathcal{C}$, it also denotes if $\boldsymbol{\lambda}$ belongs or not to the cone $\mathcal{C}$. The graphical representation of the indicator function $\psi_{\mathcal{C}_{(\lambda_N)}}(\boldsymbol{\lambda}_T)$ and its subdifferential are depicted in Fig. 2.13. Since $\boldsymbol{\lambda}_T$ is a vector, Fig. 2.13 shows the plot of the norm of the subdiffential function $\|\partial\psi_{\mathcal{C}_{(\lambda_N)}}(\boldsymbol{\lambda}_T)\|$. This subdifferential is a vector, and its direction is perpendicular to the disc $\mathcal{C}_{(\lambda_N)}$ and points outward the cone. Since the direction of the frictional force is in the opposite direction of the relative velocity, the relative velocity belongs to $-\partial\psi_{\mathcal{C}_{(\lambda_N)}}(\boldsymbol{\lambda}_T)$.


λ_N
θ_μ
$\mathcal{C}_{(\lambda_N)}$
λ_{T_1}
λ_{T_2}
$\boldsymbol{\lambda}_{T_2}$
$\dot{\boldsymbol{g}}_T$
$\mu\|\lambda_N\|$
$\boldsymbol{\lambda}_{T_1}$
• $\psi_{\mathcal{C}(\lambda_N)}(\boldsymbol{\lambda}_T) = 0$
• $\psi_{\mathcal{C}(\lambda_N)}(\boldsymbol{\lambda}_T) = +\infty$

(a) (b)

FIGURE 2.12: (a) Coulomb cone $\mathcal{C}$. (b) Disc $\mathcal{C}(\lambda_N)$ of radius equal to $\mu\lambda_N$.



(a) (b)

FIGURE 2.13: (a) Indicator function $\psi_{\mathcal{C}(\lambda_N)}$. (b) Norm of the subdifferential indicator function $\partial\psi_{\mathcal{C}(\lambda_N)}$.

Following this approach the equations of motion of a multibody system with non-impulsive frictional contact conditions imposing the constraints at velocity level can be written as:

$$\dot{\boldsymbol{q}} = \boldsymbol{v} \tag{2.33a}$$

$$\boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{v}} - \boldsymbol{g}_{\boldsymbol{q}}^T\,\boldsymbol{\lambda} = \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) \tag{2.33b}$$

$$-\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}\,\boldsymbol{v}^+ = \boldsymbol{0} \tag{2.33c}$$

$$-\boldsymbol{g}_{N\boldsymbol{q}}^{j}\boldsymbol{v} \in \partial\psi_{\mathbb{R}^+}(\lambda_N^j), \qquad \text{if } g_N^j(\boldsymbol{q}) \leq 0,\ \forall j \in \mathcal{U} \tag{2.33d}$$

$$-\boldsymbol{g}_{T\boldsymbol{q}}^{j}\boldsymbol{v} \in \partial\psi_{C(\lambda_N^j)}(\boldsymbol{\lambda}_T^j), \quad \text{if } g_N^j(\boldsymbol{q}) \leq 0,\ \forall j \in \mathcal{U} \tag{2.33e}$$

2.8 Finite element formulation

The finite element method is based on the idea to decompose a large system into small but finite elements. Therefore, the contribution of each finite element is computed independently and assembled automatically with the other elements to model the whole system. The geometrical description is done by nodes and the elements describe the relation between them.

In multibody systems a node can be used to represent a rigid body, a beam cross-section or a shell normal direction. Therefore, we assume that each node has six degrees of freedom, three translation and three rotations. The configuration of a node A can be represented by $\boldsymbol{x}_A$ and $\boldsymbol{R}_A$, where the translation $\boldsymbol{x}_A$ belongs to $\mathbb{R}^3$ and the rotation $\boldsymbol{R}_A$ belongs to the group of orthogonal linear transformations $SO(3)$. These rotation variables evolve on a non-linear space, and in order to solve numerically the equations of motion a spatial representation is needed. The classical approach is to introduce a global parametrization of the motion, expressing the rotation transformation with respect to the inertial frame in terms of a chosen set of parameters as $\boldsymbol{R} = \boldsymbol{R}(\alpha_1, \alpha_2, \alpha_3)$. The minimum number of parameters is indeed three and several choices are possible, e.g. the Euler angles, the Cartesian rotation vector or the Conformal Rotation Vector, see e.g. [5, 74]. This methodology presents some drawbacks, as the existence of singularities for any of these parametrization. Alternatively, so-called geometric methods rely on the Lie group structure of the configuration space and can be used to solve the equations of motion without any global rotation parametrization [75, 76]. These methods also naturally preserve the structure of problem.

The configuration of a node $\boldsymbol{q}_A = (\boldsymbol{x}_A, \boldsymbol{R}_A)$ evolves in a Lie group. The formulation of the equations of motion can be done with two different structures. In the first one, the equation of motions are described on the group of Euclidean transformation $SE(3)$, while in the second one they are described on the group of $\mathbb{R}^3 \times SO(3)$.

Brüls et al [50] compared the results obtained with the two formulations for the example of an spinning top. For fast spinning speeds the accuracy is better for the $\mathbb{R}^3 \times SO(3)$ formulation, whereas the formulation of $SE(3)$ leads to more accurate results for low spinning speeds. The $\mathbb{R}^3 \times SO(3)$ formulation is closer to standard approaches modeling of MBS. On the other hand, $SE(3)$ is the proper Lie group of rigid body motions, which implies better numerical properties for specific MBS [77].

In MBS there are several types of elements to represent different mechanical components or material behaviors. Some common elements are rigid bodies, flexible bodies, beams, springs, dampers, hinges and spherical joints. In this work the software Oofelie has been used to exploit all the existing element library. Therefore, for legacy reasons and to ensure the compatibility between the proposed methodology and all the existing elements in Oofelie, the Lie group formulation $\mathbb{R}^3 \times SO(3)$ has been chosen.

2.8.1 Lie group formulation

The use of a Lie group formulation allows us to formulate the equations of motion taking into account the rotation variables without the need of introducing any global parametrization. This is an advantage since the parametrization of rotation variables brings strong non-linearities and singularities. In this work, a Lie group formalism to deal with translation and rotation variables as in [78, 75] has been adopted.

From a geometrical point of view a Lie group is a differentiable manifold G with a Lie group structure. The composition rule and the inverse are smooth.

By definition a Lie group satisfies the four group axioms:

$$\begin{aligned} \boldsymbol{q}_1 \circ \boldsymbol{q}_2 &\in G \quad \forall \quad \boldsymbol{q}_1, \boldsymbol{q}_2 \in G && \text{(2.34a)} \\ (\boldsymbol{q}_1 \circ \boldsymbol{q}_2) \circ \boldsymbol{q}_3 &= \boldsymbol{q}_1 \circ (\boldsymbol{q}_2 \circ \boldsymbol{q}_3) \quad \forall \quad \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3 \in G && \text{(2.34b)} \\ \boldsymbol{q}_1 \circ \mathcal{E} &= \mathcal{E} \circ \boldsymbol{q}_1 = \boldsymbol{q}_1 \quad \forall \quad \boldsymbol{q}_1 \in G && \text{(2.34c)} \\ \forall \boldsymbol{q}_1 \in G, \exists \boldsymbol{q}_1^{-1} \in G : \boldsymbol{q}_1^{-1} \circ \boldsymbol{q}_1 &= \mathcal{E} && \text{(2.34d)} \end{aligned}$$

where $\mathcal{E}$ is the neutral element.

As mentioned before we are interested in the representation of nodal translations and rotations $(\boldsymbol{x}, \boldsymbol{R})$ defined on the Lie group $\mathbb{R}^3 \times SO(3)$ formed by the Cartesian product of $\mathbb{R}^3$ and $SO(3)$. An element of this group represents the configuration of a frame attached to the node by the pair $\boldsymbol{q} = (\boldsymbol{x}, \boldsymbol{R})$, where $\boldsymbol{x} \in \mathbb{R}^3$ is the translation vector and $\boldsymbol{R} \in SO(3)$ is the rotation matrix. The set $\mathbb{R}^3 \times SO(3)$ is a 6-dimensional Lie group with the composition operation defined as $(\boldsymbol{x}_1, \boldsymbol{R}_1) \circ (\boldsymbol{x}_2, \boldsymbol{R}_2) = (\boldsymbol{x}_1 + \boldsymbol{x}_2, \boldsymbol{R}_1\boldsymbol{R}_2)$. The neutral element of $\mathbb{R}^3 \times SO(3)$ is $(\boldsymbol{0}, \boldsymbol{I}_{3\times3})$ and the inverse of $\boldsymbol{q} = (\boldsymbol{x}, \boldsymbol{R})$ is $\boldsymbol{q}^{-1} = (-\boldsymbol{x}, \boldsymbol{R}^T)$.

The Lie algebra $\mathbb{R}^3 \times so(3)$ is defined as the tangent space at the neutral element $(\boldsymbol{0}, \boldsymbol{I}_{3\times3})$. It is a vector space, which is isomorphic to $\mathbb{R}^6$ by an invertible linear mapping

$$\boldsymbol{S}_{\mathbb{R}^3\times SO(3)}(\cdot) : \mathbb{R}^6 \to \mathbb{R}^3 \times so(3), \ \ \boldsymbol{v} \mapsto \boldsymbol{S}_{\mathbb{R}^3\times SO(3)}(\boldsymbol{v}) \tag{2.35}$$

At any configuration the velocity field $\boldsymbol{v} \in \mathbb{R}^6$ is related with the time derivative $\dot{\boldsymbol{q}}$ by the kinematic compatibility equation

$$\dot{\boldsymbol{q}} = \boldsymbol{q} \circ \boldsymbol{S}_{\mathbb{R}^3\times SO(3)}(\boldsymbol{v}) \tag{2.36}$$

Actually, the velocity vector is $\boldsymbol{v}^T = [\boldsymbol{u}^T \boldsymbol{\Omega}^T]$, with $\boldsymbol{u}$ the vector of translational velocities in the inertial frame and $\boldsymbol{\Omega}$ the vector of angular velocities in the body-attached frame. The operator $\boldsymbol{S}_{\mathbb{R}^3 \times SO(3)}$ maps a 6-dimensional velocity vector to velocity representation in the Lie algebra

$$\boldsymbol{S}_{\mathbb{R}^3 \times SO(3)}(\boldsymbol{v}) = (\boldsymbol{u}, \boldsymbol{S}(\boldsymbol{\Omega})) \tag{2.37}$$

where the operator $\boldsymbol{S} : \mathbb{R}^3 \to \mathbb{R}^3 \otimes \mathbb{R}^3$ returns a 3 × 3 skew-symmetric matrix in the Lie algebra of $SO(3)$. Therefore Eq. (2.36) yields

$$\dot{\boldsymbol{x}} = \boldsymbol{u} \tag{2.38a}$$
$$\dot{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{S}(\boldsymbol{\Omega}) \tag{2.38b}$$

Finally, the vector of virtual displacements $\delta\boldsymbol{q}$, which is needed in the formulation of d'Alembert principle, is defined as

$$\delta\boldsymbol{q} = [\delta\boldsymbol{x}^T \ \delta\boldsymbol{\Theta}^T]^T \tag{2.39}$$

where $\delta\boldsymbol{\Theta}$ represents the rotation increment vector such that $\delta\boldsymbol{R} = \boldsymbol{R}\boldsymbol{S}(\delta\boldsymbol{\Theta})$.

In addition, the exponential map of a general Lie group G is a mapping from any element of the Lie algebra $\mathfrak{g}$ to the Lie group G

$$\exp(\cdot) : \mathfrak{g} \to G, \quad \boldsymbol{S}(\Delta\boldsymbol{q}) \mapsto \boldsymbol{q} = \exp(\boldsymbol{S}(\Delta\boldsymbol{q})) \tag{2.40}$$

where $\boldsymbol{q}$ belongs to the Lie group G, $\boldsymbol{S}(\Delta\boldsymbol{q})$ to the Lie algebra $\mathfrak{g}$ and $\Delta\boldsymbol{q}$ to a vector space.

It should be remarked that with the adoption of this paradigm, in addition to unilateral constraints, bilateral constraints dealing with kinematic joints and rigid bodies will have to be managed. Next, let us introduce two examples of bilateral constraints in multibody systems, the rigid link and the spherical joint.

2.8.2 Rigid link between two nodes

A rigid link represents a fixation between two nodes, node A and node B, keeping constant the distance between them and the relative orientation. These constraints can be defined as:

$$\boldsymbol{g}(\boldsymbol{q}) \equiv \begin{Bmatrix} \boldsymbol{x}_B - \boldsymbol{x}_A - \boldsymbol{R}_A \boldsymbol{X}_{AB} \\ \boldsymbol{R}_A^T \boldsymbol{R}_B - \boldsymbol{I} \end{Bmatrix} = 0 \tag{2.41}$$

where $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$ are the rotation matrices of the nodes A and B, respectively, and $\boldsymbol{X}_{AB}$ the vector pointing from node A to B in the body-attached frame and $\boldsymbol{q} = (\boldsymbol{x}_A, \boldsymbol{R}_A, \boldsymbol{x}_B, \boldsymbol{R}_B)$. Following this methodology a broad family of joint elements can be created. For example the formulation of the hinge joint, prismatic joint, cylindrical joint, planar joint, screw joint, and few others can be found in [5, 6].

2.8.3 Spherical joint

The spherical joint is one of the simplest constraints to model. It restricts the relative translations of two nodes, node A and node B, allowing only the relative rotations. This constraints is defined as:

$$\boldsymbol{g}(\boldsymbol{q}) \equiv \boldsymbol{x}_B - \boldsymbol{x}_A = 0 \tag{2.42}$$

where $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ are the positions of the nodes A and B, respectively. This element can be treated as a constraint between two coordinates or by elimination of some coordinates.

2.8.4 Spinning top example

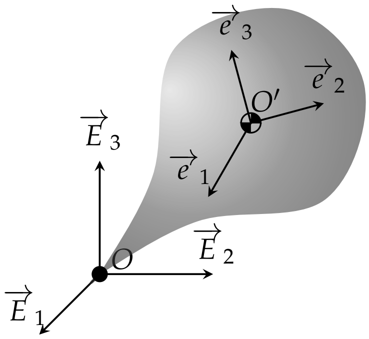


FIGURE 2.14: Spinning top example

Here, to illustrate the previous concepts, the equations of motion of the spinning top are described on a Lie group $\mathbb{R}^3 \times SO(3)$. A more detailed description of this example can be found in [75].

The spinning top consists in a heavy mass with a symmetry axis rotating and fixed to the ground by a spherical joint as depicted in Fig. 2.14. We consider the origin of the material frame at the center of mass O', and the rigid body is linked to the ground at the origin O. According to our finite element framework the configuration of the system is represented by $\boldsymbol{q} = (\boldsymbol{x}, \boldsymbol{R})$, where $\boldsymbol{x}$ is the translation of the center of mass and $\boldsymbol{R}$ is the rotation matrix. Then the kinetic energy and the potential energy of the system take the form

$$\begin{aligned} \mathcal{K}(\dot{\boldsymbol{x}}, \boldsymbol{\Omega}) &= \tfrac{1}{2}(m\dot{\boldsymbol{x}}^T\dot{\boldsymbol{x}} + \boldsymbol{\Omega}^T\boldsymbol{J}\boldsymbol{\Omega}) \\ \mathcal{V}(\boldsymbol{q}) &= -\boldsymbol{x}^T m\boldsymbol{a}_g \end{aligned} \tag{2.43}$$

where m is the mass, $\boldsymbol{a}_g$ is the gravity acceleration, $\boldsymbol{J}$ is the inertial tensor with respect the center of mass in the body attached frame and $\boldsymbol{X}$ is the position of the center of mass in the reference configuration. To completely define the system, the translation restriction at the fixation is introduced through bilateral constraints as

$$\boldsymbol{g}(\boldsymbol{q}) = -\boldsymbol{R}^T\boldsymbol{x} + \boldsymbol{X} \tag{2.44}$$

Applying the d'Alembert principle, the equations of motion for a top spining systems can be expressed as follows:

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{u} && \text{(2.45a)}\\
\dot{\boldsymbol{R}} &= \boldsymbol{R}\boldsymbol{S}(\boldsymbol{\Omega}) && \text{(2.45b)}\\
m\dot{\boldsymbol{u}} - m\boldsymbol{a}_g - \boldsymbol{R}\boldsymbol{\lambda} &= \boldsymbol{0} && \text{(2.45c)}\\
\boldsymbol{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} + \boldsymbol{X} \times \boldsymbol{\lambda} &= \boldsymbol{0} && \text{(2.45d)}\\
-\boldsymbol{R}^T\boldsymbol{x} + \boldsymbol{X} &= 0 && \text{(2.45e)}
\end{aligned}$$

where $\boldsymbol{u}$ is the linear velocity and $\dot{\boldsymbol{u}}$ and $\dot{\boldsymbol{\Omega}}$ represent the linear and angular accelerations, respectively.

Eqs. (2.45a-2.45b) represent the kinematic compatibility conditions for the translation and rotation variables. The balance of linear and angular momentums is represented by Eq. (2.45c) and (2.45d). $m\dot{\boldsymbol{u}}$ represents the inertial forces, $m\boldsymbol{a}_g$ the gravitational forces and $\boldsymbol{\lambda}$ are the Lagrange multipliers associated with the bilateral constraints, which are the necessary internal reaction forces to satisfy the rigid body condition between the origin O and the center of mass of the top. In Eq. (2.45d) the first two terms of the sum represent the inertial forces, including second the gyroscopic forces, and the last one represents the constraint forces. Finally, Eq. (2.45e) imposes the rigid body constraints, ensuring that the distance between the points O and O' does not change during the evolution of the system.

2.9 Generalized-α method with a Lie group formulation

As shown in the previous sections, the configuration of a flexible multibody system can generally be described by a variable $\boldsymbol{q} \in G$, where G is a k-dimensional manifold with a matrix Lie group structure. Then the equations of motions can be solved using a Lie group time integrator. The generalized-α scheme was extended to systems on Lie group in [75]:

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q}_{n+1})\dot{\boldsymbol{v}}_{n+1} + \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}_{n+1})\boldsymbol{\lambda}_{n+1} = -\boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1}) && \text{(2.46a)}\\
\boldsymbol{g}(\boldsymbol{q}_{n+1}) = \boldsymbol{0} && \text{(2.46b)}\\
\boldsymbol{q}_{n+1} = \boldsymbol{q}_n \circ \exp(h\boldsymbol{S}(\Delta\boldsymbol{q}_n)) && \text{(2.46c)}\\
\Delta\boldsymbol{q}_n = \boldsymbol{v}_n + (0.5-\beta)h\boldsymbol{a}_n + \beta h\boldsymbol{a}_{n+1} && \text{(2.46d)}\\
\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + (1-\gamma)h\boldsymbol{a}_n + \gamma h\boldsymbol{a}_{n+1} && \text{(2.46e)}\\
(1-\alpha_m)\boldsymbol{a}_{n+1} + \alpha_m\boldsymbol{a}_n = (1-\alpha_f)\dot{\boldsymbol{v}}_{n+1} + \alpha_f\dot{\boldsymbol{v}}_n && \text{(2.46f)}
\end{aligned}$$

where h is the time step, $\Delta \boldsymbol{q}$ can be interpreted as an average velocity field in the time step, and $\alpha_m, \alpha_f, \beta$ and γ are numerical parameters related to the generalized-α scheme. According to [48], these coefficients can be defined from the so-called spectral radius at infinity, $\rho_\infty \in [0, 1]$, as:

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad \gamma = 0.5 + \alpha_f - \alpha_m, \quad \beta = 0.25(\gamma + 0.5)^2 \tag{2.47}$$

The numerical coefficients are thus chosen to achieve a desired level of high-frequency dissipation, while minimizing unwanted low-frequency dissipation. No dissipation is achieved for a spectral radius $\rho_\infty = 1$, while a high level of dissipation is achieved for $\rho_\infty = 0$, annihilating the high-frequency response after one time step. Chung and Hubert [48] also demonstrate that the method is unconditionally stable for:

$$\alpha_m \leq \alpha_f \leq \frac{1}{2}, \quad \beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m) \tag{2.48}$$

The generalized-α method includes other methods as special cases, like the HHT method ($\alpha_m = 0, \alpha_f \in [0, 1/3]$) or the Newmark algorithm ($\alpha_f = \alpha_m = 0$) [60]. As mentioned in Section 1.6 the second-order convergence of the generalized-α for constrained mechanical systems is proven in [79]. Second-order accuracy is a desirable property to capture the vibration effects of the flexible bodies, which can be particularly determinant for flexible bodies subject to impacts.

As mentioned at the beginning of this section, it is observed that no parametrization of rotations is needed to solve the equations of motion using the Lie group generalized-α method presented in Eq. (2.46).

2.10 Summary

This chapter presents the equations of motion for MBS with rigid/flexible bodies, bilateral constraints and unilateral frictional contact conditions. The system is represented according to the finite element approach and the nodal translation and rotation variables are treated as elements of the Lie group $\mathbb{R}^3 \times SO(3)$. The equations of motion are introduced gradually, starting from the virtual work principle which results in a DAE system to represent constrained MBS. A nonsmooth formalism is then introduced for the description of impacts and discontinuous phenomena. Complementarity conditions and inclusions are exploited to properly represent the unilateral constraints and the frictional Coulomb law. Finally the Lie group generalized-α time integrator for dynamic mechanical systems is introduced. The extension of this solver to deal with nonsmooth dynamic systems will be addressed in the next chapter.

Chapter 3

Nonsmooth generalized-α solver

This chapter presents the development of a robust and accurate time integrator for the dynamic simulation of multibody systems composed by rigid and/or flexible bodies subject to frictionless contacts and impacts. Its content is adapted from the paper [80]. The nonsmooth generalized-α scheme (NSGA) is a time-stepping integrator which is able to deal well with nonsmooth dynamical problems avoiding any constraint drift phenomena and capturing vibration effects without introducing too much numerical dissipation. However, when dealing with problems involving nonlinear bilateral constraints and/or flexible elements, it is necessary to adopt small stepsizes to ensure the convergence of the numerical scheme. In order to tackle these problems more efficiently, a decoupled version of the original nonsmooth generalized-α method is proposed in this work. The chapter is structured as follows. First, the equations of motion are introduced, then the new splitting strategy for the NSGA is presented and the algorithm is explained in detail. Finally several examples are considered to assess its accuracy and robustness.

3.1 Equations of motion

In this chapter, we consider that the configuration of the system is represented by a vector $\boldsymbol{q}$. The complexities of the treatment of rotational variables are thus left aside for the moment.

After spatial semi-discretization, the equations of motion for a frictionless multibody system with unilateral and bilateral constraints expressed at velocity level are written in the following form:

$$\begin{aligned}
\dot{\boldsymbol{q}}^+ &= \boldsymbol{v}^+ && \text{(3.1a)}\\
\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{v} - \boldsymbol{g}_{\boldsymbol{q}}^T\,\mathrm{d}\boldsymbol{i} &= \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t)\,\mathrm{d}t && \text{(3.1b)}\\
-\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}\,\boldsymbol{v}^+ &= \boldsymbol{0} && \text{(3.1c)}\\
\text{if } g^j(\boldsymbol{q}) \leq 0 \text{ then } 0 \leq g_{\boldsymbol{q}}^j\,\boldsymbol{v}^+ + e^j\,g_{\boldsymbol{q}}^j\,\boldsymbol{v}^- &\perp \mathrm{d}i^j \geq 0, \quad \forall\, j \in \mathcal{U} && \text{(3.1d)}
\end{aligned}$$

where

- t is time, and $\mathrm{d}t$ is the corresponding standard Lebesgue measure.

- $\boldsymbol{q}(t)$ is the vector of coordinates, which are absolutely continuous in time.
- $\mathcal{U}$ denotes the set of indices of the unilateral constraints, $\overline{\mathcal{U}}$ is its complementarity set, *i.e.*, the set of bilateral constraints, $\mathcal{C} = \mathcal{U} \cup \overline{\mathcal{U}}$ is the full set of constraints.
- $\boldsymbol{g}$ is the combined set of bilateral and unilateral constraints, and $\boldsymbol{g_q}(\boldsymbol{q})$ is the corresponding matrix of constraint gradients.
- $\dot{\boldsymbol{q}}^+(t) = \lim_{\tau \to t, \tau > t} \dot{\boldsymbol{q}}(\tau)$ and $\boldsymbol{v}^+(t) = \lim_{\tau \to t, \tau > t} \boldsymbol{v}(\tau)$ are the right limits of the velocity, which are functions of bounded variations. Similarly, $\boldsymbol{v}^-(t) = \lim_{\tau \to t, \tau < t} \boldsymbol{v}(\tau)$ is the left limit of the velocity. It is assumed, without loss of generality, that $\boldsymbol{v}^+$ and $\boldsymbol{v}^-$ are related to an impact event by the Newton impact law $g^j_{\boldsymbol{q}} \boldsymbol{v}^+(t) = -e^j g^j_{\boldsymbol{q}} \boldsymbol{v}^-(t)$, where e^j is the coefficient of restitution at the contact point $j \in \mathcal{U}$. In what follows, for simplicity $\boldsymbol{v}(t)$ and $\dot{\boldsymbol{q}}(t)$ will be used to denote $\boldsymbol{v}^+(t)$ and $\dot{\boldsymbol{q}}^+(t)$, respectively.
- $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) = \boldsymbol{f}^{ext}(t) - \boldsymbol{f}^{cin}(\boldsymbol{q}, \boldsymbol{v}) - \boldsymbol{f}^{damp}(\boldsymbol{q}, \boldsymbol{v}) - \boldsymbol{f}^{int}(\boldsymbol{q})$ collects the external, complementary inertia, damping and internal forces.
- $\boldsymbol{M}(\boldsymbol{q})$ is the mass matrix which may, in general, depend on the coordinates.
- $\mathrm{d}\boldsymbol{v}$ is the differential measure associated with the velocity $\boldsymbol{v}$ assumed to be of bounded variations.
- $\mathrm{d}\boldsymbol{i}$ is the impulse measure of the unilateral contact reaction and the bilateral constraint forces.
- The measures $\mathrm{d}\boldsymbol{v}$ and $\mathrm{d}\boldsymbol{i}$ have the following decomposition:

$$\mathrm{d}\boldsymbol{v} = \dot{\boldsymbol{v}}\, \mathrm{d}t + \sum_i (\boldsymbol{v}(t_i) - \boldsymbol{v}^-(t_i))\, \delta_{t_i} \tag{3.2}$$

$$\mathrm{d}\boldsymbol{i} = \boldsymbol{\lambda}\, \mathrm{d}t + \sum_i \boldsymbol{p}_i\, \delta_{t_i} \tag{3.3}$$

  where $\boldsymbol{\lambda}$ is the vector of smooth Lagrange multipliers associated with the Lebesgue measurable constraint forces, $\boldsymbol{\delta}_{t_i}$ the Dirac delta supported at t_i, and $\boldsymbol{p}_i$ is the impulse producing the jump at the instant t_i.

3.2 Decoupled version of the NSGA method

The original NSGA method proposed by Brüls *et al.* [50] is characterized by three coupled sub-problems: one for the smooth part of motion, and two others for the nonsmooth contributions at position and velocity levels. In order to avoid solving the three systems of equations monolithically, the terms coupling the smooth prediction to the corrections at position and at velocity levels are neglected. However, as it will be shown in the numerical examples, this can have serious consequences on the convergence of the method.

The derivation of the time integration scheme to be proposed here follows the same key steps as described in [50]. However, two differences should be highlighted. Firstly, the splitting is modified in order to ensure a full decoupling of the different subsets of equations. Secondly, the equations are formulated using an augmented Lagrangian approach which combines Lagrange multiplier and penalty terms. The advantage of using an augmented Lagrangian method is the presence of a penalty term which adds convexity to the objective function and improves the convergence of the Newton iteration far from the solution. This factor does not influence the accuracy of the computed solution [5].

3.2.1 Splitting strategy

The splitting of the variables aims at isolating the impulsive terms from the smooth contributions to the motion. Let us consider a time interval $(t_n, t_{n+1}]$, and let $\dot{\tilde{\boldsymbol{v}}}(t)$ be a function of bounded variations, which can be defined in several different ways as it will be discussed below. Then, the nonsmooth impulsive contribution to the motion $\mathrm{d}\boldsymbol{w}$ can be defined by decomposing the measure of the velocity $\mathrm{d}\boldsymbol{v}$ as

$$\mathrm{d}\boldsymbol{v} = \mathrm{d}\boldsymbol{w} + \dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t \tag{3.4}$$

where $\dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t$ is a purely diffuse measure. The smooth contribution to the velocity field, denoted by $\tilde{\boldsymbol{v}}(t)$, is computed by integration of $\dot{\tilde{\boldsymbol{v}}}(t)$ with the initial values $\tilde{\boldsymbol{v}}(t_n) = \boldsymbol{v}(t_n)$. On the other hand, the smooth contribution to the position, denoted as $\tilde{\boldsymbol{q}}(t)$, is computed by integration of $\dot{\tilde{\boldsymbol{q}}}(t) = \tilde{\boldsymbol{v}}(t)$ with the initial values $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$. By construction, the variables $\tilde{\boldsymbol{v}}(t)$ and $\tilde{\boldsymbol{q}}(t)$ are absolutely continuous and C^1 in time, respectively. In this sense, they only capture a smooth part of the motion.

The smooth part of the trajectory is obtained from the time integration of the acceleration variable $\dot{\tilde{\boldsymbol{v}}}$ by a second order method, whereas $\mathrm{d}\boldsymbol{w}$ is integrated using a first-order Euler implicit scheme. Hence, it is recommended to capture as much information as possible in $\dot{\tilde{\boldsymbol{v}}}$ to gain accuracy. Several alternatives exist for defining the smooth part of motion in the splitting, as described next.

- The optimal choice would be to take $\dot{\tilde{\boldsymbol{v}}} \triangleq \dot{\boldsymbol{v}}$, so that $\dot{\tilde{\boldsymbol{v}}}$ captures all diffuse contributions in the equation of motion. This alternative was investigated in [71] based on a formulation of the constraints at acceleration level, bringing certain advantages such as the elimination of spurious oscillations of the constraints after impact events.

- Some simplifications can be proposed to avoid the need of using acceleration constraints and to eliminate the Linear Complementarity Problem (LCP) in the definition of $\dot{\tilde{\boldsymbol{v}}}$. For instance, Chen *et al.* [60] defined $\dot{\tilde{\boldsymbol{v}}}$ to satisfy the equations of motion at almost any time t (when there is no impact) without accounting for the bilateral and the unilateral constraints and associated forces.

- A third alternative is proposed in [50], in which a compromise between the option of taking $\dot{\tilde{\boldsymbol{v}}} \triangleq \dot{\boldsymbol{v}}$ and that of Chen *et al.* [60] is made by defining an initial value problem for the smooth contributions in which the bilateral constraints and forces are considered. In this case, the smooth contributions to the position $\tilde{\boldsymbol{q}}(t)$, the velocity $\tilde{\boldsymbol{v}}(t)$, the acceleration $\dot{\tilde{\boldsymbol{v}}}(t)$ and the Lagrange multiplier $\tilde{\boldsymbol{\lambda}}(t)$ satisfy

$$\dot{\tilde{\boldsymbol{q}}} = \tilde{\boldsymbol{v}} \tag{3.5a}$$

$$\boldsymbol{M}(\boldsymbol{q})\,\dot{\tilde{\boldsymbol{v}}} - \boldsymbol{g}_{\boldsymbol{q}}^T\,\tilde{\boldsymbol{\lambda}} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) \tag{3.5b}$$

$$-\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}}(\boldsymbol{q})\,\tilde{\boldsymbol{v}} = \boldsymbol{0} \tag{3.5c}$$

$$\tilde{\boldsymbol{\lambda}}^{\mathcal{U}} = \boldsymbol{0} \tag{3.5d}$$

  with the initial values $\tilde{\boldsymbol{v}}(t_n) = \boldsymbol{v}(t_n)$ and $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$. It is remarked that only bilateral constraints at velocity level are taken into account in this formulation and that the Lagrange multipliers of the unilateral constraints are set to zero.

- A fourth alternative could be obtained by setting the smooth acceleration variable $\dot{\tilde{\boldsymbol{v}}}$ to zero so that all the dynamics would be integrated using the Euler implicit method (with a loss of accuracy with respect to the other options).

- The first three alternatives for the definition of $\dot{\tilde{\boldsymbol{v}}}$ have the disadvantage that the resulting equations depend on the total position and velocity, that is, they depend not only on the smooth components of the position and the velocity, but also on the nonsmooth (impulsive) components. Therefore, these formulations are characterized by a smooth sub-problem which is coupled with the set of equations defining the nonsmooth contributions. In order to avoid this coupling, we propose to define the smooth sub-problem as the solution of the following initial value problem:

$$\dot{\tilde{\boldsymbol{q}}} = \tilde{\boldsymbol{v}} \tag{3.6a}$$

$$\boldsymbol{M}(\tilde{\boldsymbol{q}})\,\dot{\tilde{\boldsymbol{v}}} - \boldsymbol{g}_{\tilde{\boldsymbol{q}}}^T\,\tilde{\boldsymbol{\lambda}} = \boldsymbol{f}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}, t) \tag{3.6b}$$

$$-\boldsymbol{g}_{\tilde{\boldsymbol{q}}}^{\overline{\mathcal{U}}}(\tilde{\boldsymbol{q}})\,\tilde{\boldsymbol{v}} = \boldsymbol{0} \tag{3.6c}$$

$$\tilde{\boldsymbol{\lambda}}^{\mathcal{U}} = \boldsymbol{0} \tag{3.6d}$$

  with the initial values $\tilde{\boldsymbol{v}}(t_n) = \boldsymbol{v}(t_n)$ and $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$, and where the matrix of constraint gradients $\boldsymbol{g}_{\tilde{\boldsymbol{q}}}(\tilde{\boldsymbol{q}})$ is computed in terms of $\tilde{\boldsymbol{q}}$ only. It should be observed that this fifth definition of $\dot{\tilde{\boldsymbol{v}}}$ only depends on the smooth contributions to the motion, $\tilde{\boldsymbol{q}}$ and $\tilde{\boldsymbol{v}}$, a property that naturally leads to a sequence of decoupled sub-problems.

An elimination of $\mathrm{d}\boldsymbol{v}$ and $\dot{\tilde{\boldsymbol{v}}}$ from Eqs. (3.1b, 3.4, 3.6b) yields the equations for the nonsmooth contributions

$$\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{w} - \boldsymbol{g}_{\boldsymbol{q}}^T\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] = \boldsymbol{f}^*(\boldsymbol{q}, \boldsymbol{v}, \tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}, \dot{\tilde{\boldsymbol{v}}}, t)\,\mathrm{d}t \tag{3.7}$$

together with the set of bilateral and unilateral constraints Eqs. (3.1c-3.1d), and where

$$\boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t) = \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) - \boldsymbol{f}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},t) + (\boldsymbol{g}_{\boldsymbol{q}}^T - \boldsymbol{g}_{\tilde{\boldsymbol{q}}}^T)\tilde{\boldsymbol{\lambda}} - (\boldsymbol{M}(\boldsymbol{q}) - \boldsymbol{M}(\tilde{\boldsymbol{q}}))\,\dot{\tilde{\boldsymbol{v}}} \tag{3.8}$$

3.2.2 Constraints at position and velocity levels

The constraints in the derived set of equations are imposed at velocity level only. Moreau's viability Lemma [73] implies that the exact solution of these equations also satisfies the constraints at position level. However, the numerical solution will not satisfy them at position level because of a drift phenomenon. Therefore, the equations of motion are reformulated as in [50] such that the unilateral and bilateral constraints appear both at position and velocity levels. This procedure is inspired by the index-reduction proposed by Gear, Gupta and Leimkuhler for bilaterally constrained mechanical systems [62]. By introducing an additional Lagrange multiplier $\boldsymbol{\mu}$, the velocity equation (3.1a) is relaxed:

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v} + \boldsymbol{g}_{\boldsymbol{q}}^T\boldsymbol{\mu} \tag{3.9}$$

and by adding the constraints on position, the set of equations to be solved becomes

$$\begin{aligned}
\boldsymbol{M}(\tilde{\boldsymbol{q}})\,\dot{\tilde{\boldsymbol{v}}} - \boldsymbol{g}_{\tilde{\boldsymbol{q}}}^{\bar{\mathcal{U}},T}\,\tilde{\boldsymbol{\lambda}}^{\bar{\mathcal{U}}} &= \boldsymbol{f}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},t) && \text{(3.10a)}\\
-\boldsymbol{g}_{\tilde{\boldsymbol{q}}}^{\bar{\mathcal{U}}}\,\tilde{\boldsymbol{v}} &= \boldsymbol{0} && \text{(3.10b)}\\
\tilde{\boldsymbol{\lambda}}^{\mathcal{U}} &= \boldsymbol{0} && \text{(3.10c)}\\
\mathrm{d}\boldsymbol{v} &= \mathrm{d}\boldsymbol{w} + \dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t && \text{(3.10d)}\\
\boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{q}} - \boldsymbol{g}_{\boldsymbol{q}}^T\,\boldsymbol{\mu} &= \boldsymbol{M}(\boldsymbol{q})\,\boldsymbol{v} && \text{(3.10e)}\\
-\boldsymbol{g}^{\bar{\mathcal{U}}}(\boldsymbol{q}) &= \boldsymbol{0} && \text{(3.10f)}\\
\boldsymbol{0} \le \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}) &\perp \boldsymbol{\mu} \ge \boldsymbol{0} && \text{(3.10g)}\\
\boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{w} - \boldsymbol{g}_{\boldsymbol{q}}^T\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] &= \boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t)\,\mathrm{d}t && \text{(3.10h)}\\
-\boldsymbol{g}_{\boldsymbol{q}}^{\bar{\mathcal{U}}}\boldsymbol{v} &= \boldsymbol{0} && \text{(3.10i)}\\
\text{if } g^j(\boldsymbol{q}) \le 0 \text{ then } 0 \le g_{\boldsymbol{q}}^j\,\boldsymbol{v} + e^j\,g_{\boldsymbol{q}}^j\,\boldsymbol{v}^- &\perp \mathrm{d}i^j \ge 0, \quad \forall j \in \mathcal{U} && \text{(3.10j)}
\end{aligned}$$

3.2.3 Time stepping scheme

Velocity jump and position correction variables

Time integration of the velocity measure $\mathrm{d}\boldsymbol{v}$, Eq. (3.4), over the time interval $(t_n, t]$ gives

$$\boldsymbol{v}(t) = \int_{(t_n,t]} \mathrm{d}\boldsymbol{w} + \tilde{\boldsymbol{v}}(t) = \boldsymbol{W}(t_n;t) + \tilde{\boldsymbol{v}}(t) \tag{3.11}$$

with the definition $\boldsymbol{W}(t_n;t) = \int_{(t_n,t]} \mathrm{d}\boldsymbol{w}$ and where the properties $\boldsymbol{v}(t_n) = \tilde{\boldsymbol{v}}(t_n)$ and $\boldsymbol{q}(t_n) = \tilde{\boldsymbol{q}}(t_n)$ were used. By construction, the nonsmooth variable $\boldsymbol{W}(t_n;t)$ captures all velocity jumps taking place in the interval $(t_n, t]$.

In order to obtain a similar decomposition of the position variable, it is first recalled that $\boldsymbol{v}(t)$ and $\dot{\boldsymbol{q}}(t)$ are not formally equivalent, where the variable $\dot{\boldsymbol{q}}(t)$ is related to the position variable $\boldsymbol{q}(t)$ by the relation

$$\int_{(t_n,t]} \dot{\boldsymbol{q}}(\tau)\mathrm{d}\tau = \boldsymbol{q}(t) - \boldsymbol{q}(t_n) \tag{3.12}$$

Time integration of the smooth component $\tilde{\boldsymbol{v}}(t)$ gives

$$\int_{(t_n,t]} \tilde{\boldsymbol{v}}(\tau)\mathrm{d}\tau = \tilde{\boldsymbol{q}}(t) - \boldsymbol{q}(t_n) \tag{3.13}$$

where the property $\tilde{\boldsymbol{q}}(t_n) = \boldsymbol{q}(t_n)$ was used. Subtracting Eq. (3.13) from Eq. (3.12) results in

$$\boldsymbol{q}(t) = \int_{(t_n,t]} [\dot{\boldsymbol{q}}(\tau) - \tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau + \tilde{\boldsymbol{q}}(t) = \boldsymbol{U}(t_n;t) + \tilde{\boldsymbol{q}}(t) \tag{3.14}$$

with the definition of the position correction $\boldsymbol{U}(t_n;t) = \int_{(t_n,t]} [\dot{\boldsymbol{q}}(\tau) - \tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau$. Considering that $\boldsymbol{q}(t)$ is absolutely continuous and that $\tilde{\boldsymbol{q}}(t)$ is C^1, Eq. (3.14) implies that $\boldsymbol{U}(t_n;t)$ is absolutely continuous on the interval $(t_n, t_{n+1}]$.

In summary, we have the following expressions for the splitting of the total velocity and position

$$\boldsymbol{v}(t) = \tilde{\boldsymbol{v}}(t) + \boldsymbol{W}(t_n;t) \tag{3.15a}$$
$$\boldsymbol{q}(t) = \tilde{\boldsymbol{q}}(t) + \boldsymbol{U}(t_n;t) \tag{3.15b}$$

where the nonsmooth contributions $\boldsymbol{W}(t_n;t)$ and $\boldsymbol{U}(t_n;t)$ are defined by

$$\boldsymbol{W}(t_n;t) = \int_{(t_n,t]} \mathrm{d}\boldsymbol{w} \tag{3.16a}$$
$$\boldsymbol{U}(t_n;t) = \int_{(t_n,t]} [\dot{\boldsymbol{q}}(\tau) - \tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau \tag{3.16b}$$

where, by construction, $\boldsymbol{W}(t_n;t_n) = \mathbf{0}$ and $\boldsymbol{U}(t_n;t_n) = \mathbf{0}$. In what follows, the multipliers $\boldsymbol{\Lambda}(t_n;t)$ and $\boldsymbol{\nu}(t_n;t)$ are taken as

$$\boldsymbol{\Lambda}(t_n;t) = \int_{(t_n,t]} [\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}(\tau)\,\mathrm{d}\tau], \tag{3.17a}$$
$$\boldsymbol{\nu}(t_n;t) = \int_{(t_n,t]} [\boldsymbol{\mu}(\tau) + \boldsymbol{\Lambda}(t_n;\tau)]\,\mathrm{d}\tau, \tag{3.17b}$$

with $\boldsymbol{\Lambda}(t_n;t_n) = \boldsymbol{\nu}(t_n;t_n) = \mathbf{0}$.

Discrete approximations of $\boldsymbol{W}$ and $\boldsymbol{U}$

The discrete approximations of the involved variables are now introduced. The time instant at which a discretized variable is evaluated will be indicated with a subscript, *e.g.*, $\boldsymbol{g}_{n+1}$ represents the approximation of $\boldsymbol{g}(t_{n+1})$. Also the discrete approximation of $\boldsymbol{W}(t_n;t_{n+1})$ (resp. $\boldsymbol{U}(t_n;t_{n+1})$, $\boldsymbol{\Lambda}(t_n;t_{n+1})$ or $\boldsymbol{\nu}(t_n;t_{n+1})$) will be denoted as $\boldsymbol{W}_{n+1}$ (resp., $\boldsymbol{U}_{n+1}$, $\boldsymbol{\Lambda}_{n+1}$ or $\boldsymbol{\nu}_{n+1}$).

Firstly, a discrete expression for Eq. (3.10h) is obtained by time integration

$$\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{w} - \int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] = \int_{(t_n,t_{n+1}]} \boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t)\,\mathrm{d}t \tag{3.18}$$

The three terms involved in the above expression can be interpreted as *nonsmooth contributions*, in the sense that each of them represents a difference between the actual motion and the smooth motion. Indeed, $\mathrm{d}\boldsymbol{w}$ is the difference between $\mathrm{d}\boldsymbol{v}$ and $\dot{\tilde{\boldsymbol{v}}}\,\mathrm{d}t$ and the expression of $\boldsymbol{f}^*$ in (3.8) involves the difference between three operators ($\boldsymbol{f}$, $\boldsymbol{g}_{\boldsymbol{q}}$ and $\boldsymbol{M}$) evaluated for the actual motion and for the smooth motion. According to the NSGA method, the integration of these nonsmooth contributions is based on first-order approximations $\boldsymbol{M}(\boldsymbol{q}(\tau)) = \boldsymbol{M}(\boldsymbol{q}(t)) + \mathcal{O}(h)$ and $\boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(\tau)) = \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(t)) + \mathcal{O}(h), \forall\tau \in (t_n,t_{n+1}]$ and an Euler implicit discretization

$$\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q}(t))\,\mathrm{d}\boldsymbol{w} \simeq \boldsymbol{M}(\boldsymbol{q}_{n+1}) \int_{(t_n,t_{n+1}]} \mathrm{d}\boldsymbol{w} \simeq \boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{W}_{n+1} \tag{3.19}$$

$$\int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(t))\,[\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] \simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T \int_{(t_n,t_{n+1}]} [\mathrm{d}\boldsymbol{i} - \tilde{\boldsymbol{\lambda}}\,\mathrm{d}t] \simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T\,\boldsymbol{\Lambda}_{n+1} \tag{3.20}$$

$$\int_{(t_n,t_{n+1}]} \boldsymbol{f}^*(\boldsymbol{q}(t),\boldsymbol{v}(t),\tilde{\boldsymbol{q}}(t),\tilde{\boldsymbol{v}}(t),\dot{\tilde{\boldsymbol{v}}}(t),t)\,\mathrm{d}t \simeq h\boldsymbol{f}_{n+1}^* \tag{3.21}$$

This leads to the discrete equation

$$\boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{W}_{n+1} - \boldsymbol{g}_{\boldsymbol{q},n+1}^T\,\boldsymbol{\Lambda}_{n+1} - h\boldsymbol{f}_{n+1}^* = \boldsymbol{0} \tag{3.22}$$

Secondly, a discrete expression for Eq. (3.10e) is obtained by time integration

$$\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{q}}\,\mathrm{d}t - \int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,\boldsymbol{\mu}\,\mathrm{d}t = \int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,\boldsymbol{v}\,\mathrm{d}t \tag{3.23}$$

Then, Eq. (3.15a) is used together with the approximation

$$\boldsymbol{M}(\boldsymbol{q}(t))\,\boldsymbol{W}(t_n;t) \simeq \boldsymbol{g}_{\boldsymbol{q}}^T(\boldsymbol{q}(t))\,\boldsymbol{\Lambda}(t_n;t) + h\boldsymbol{f}^*(\boldsymbol{q}(t),\boldsymbol{v}(t),\tilde{\boldsymbol{q}}(t),\tilde{\boldsymbol{v}}(t),\dot{\tilde{\boldsymbol{v}}}(t),t) \tag{3.24}$$

which can be derived in a similar way as Eq. (3.22). We obtain

$$\begin{aligned}\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,(\dot{\boldsymbol{q}} - \tilde{\boldsymbol{v}})\,\mathrm{d}t - \int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,(\boldsymbol{\Lambda}(t_n;t) + \boldsymbol{\mu}(t))\,\mathrm{d}t \\ = h\int_{(t_n,t_{n+1}]} \boldsymbol{f}^*(\boldsymbol{q},\boldsymbol{v},\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\dot{\tilde{\boldsymbol{v}}},t)\,\mathrm{d}t\end{aligned} \tag{3.25}$$

Again, the three terms involved in the above equality are interpreted as *nonsmooth contributions* and are thus integrated based on first-order approximations and an

Euler implicit discretization

$$\begin{aligned}\int_{(t_n,t_{n+1}]} \boldsymbol{M}(\boldsymbol{q})\,(\dot{\boldsymbol{q}}-\tilde{\boldsymbol{v}})\,\mathrm{d}t &\simeq \boldsymbol{M}(\boldsymbol{q}_{n+1})\int_{(t_n,t_{n+1}]}(\dot{\boldsymbol{q}}-\tilde{\boldsymbol{v}})\,\mathrm{d}t \\ &\simeq \boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{U}_{n+1}\end{aligned} \tag{3.26}$$

$$\begin{aligned}\int_{(t_n,t_{n+1}]} \boldsymbol{g}_{\boldsymbol{q}}^T\,(\boldsymbol{\Lambda}(t_n;t)+\boldsymbol{\mu}(t))\,\mathrm{d}t &\simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T\int_{(t_n,t_{n+1}]}(\boldsymbol{\Lambda}(t_n;t)+\boldsymbol{\mu}(t))\,\mathrm{d}t \\ &\simeq \boldsymbol{g}_{\boldsymbol{q},n+1}^T\boldsymbol{\nu}_{n+1}\end{aligned} \tag{3.27}$$

and Eq. (3.21). This leads to the discrete equation

$$\boldsymbol{M}(\boldsymbol{q}_{n+1})\,\boldsymbol{U}_{n+1}-\boldsymbol{g}_{\boldsymbol{q},n+1}^T\boldsymbol{\nu}_{n+1}-h^2\boldsymbol{f}_{n+1}^*=\boldsymbol{0} \tag{3.28}$$

We will see later that this equation shall be used to compute the position correction $\boldsymbol{U}_{n+1}$, after the evaluation of the smooth motion $\tilde{\boldsymbol{q}}_{n+1}$, $\tilde{\boldsymbol{v}}_{n+1}$, $\dot{\tilde{\boldsymbol{v}}}_{n+1}$ but before the evaluation of the velocity jump $\boldsymbol{W}_{n+1}$ and of the total velocity $\boldsymbol{v}_{n+1}$. The dependency of the operator $\boldsymbol{f}^*$ on the velocity then leads to a coupling between the equation for the position correction and the equation for the velocity jump. The mass matrix depends continuously on the configuration $\boldsymbol{q}$. Since $\boldsymbol{q}=\tilde{\boldsymbol{q}}+\boldsymbol{U}$ and $\boldsymbol{U}$ is $\mathcal{O}(h)$, then $\boldsymbol{M}(\boldsymbol{q})=\boldsymbol{M}(\tilde{\boldsymbol{q}})+\mathcal{O}(h)$. Therefore, a simplified version of Eq. (3.28) is considered to avoid the coupling between the position and velocity variables and to eliminate the dependency of the mass matrix on $\boldsymbol{U}$

$$\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\,\boldsymbol{U}_{n+1}-\boldsymbol{g}_{\boldsymbol{q},n+1}^T\boldsymbol{\nu}_{n+1}-h^2\boldsymbol{f}_{n+1}^p=\boldsymbol{0} \tag{3.29}$$

with

$$\boldsymbol{f}_{n+1}^p=\boldsymbol{f}(\boldsymbol{q}_{n+1},\tilde{\boldsymbol{v}}_{n+1},t_{n+1})-\boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1},\tilde{\boldsymbol{v}}_{n+1},t_{n+1})+(\boldsymbol{g}_{\boldsymbol{q},n+1}^T-\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^T)\tilde{\boldsymbol{\lambda}}_{n+1} \tag{3.30}$$

In summary, we get the following discrete equations

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q}_{n+1})\;\boldsymbol{W}_{n+1}-\boldsymbol{g}_{\boldsymbol{q},n+1}^T\;\boldsymbol{\Lambda}_{n+1}-h\boldsymbol{f}_{n+1}^* &= \boldsymbol{0} &\text{(3.31a)}\\ \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\,\boldsymbol{U}_{n+1}-\boldsymbol{g}_{\boldsymbol{q},n+1}^T\boldsymbol{\nu}_{n+1}-h^2\boldsymbol{f}_{n+1}^p &= \boldsymbol{0} &\text{(3.31b)}\end{aligned}$$

Let us remark that, if impacts occur and $h\to 0$, $\boldsymbol{W}(t_n;t_{n+1})=\mathcal{O}(1)$ and $\boldsymbol{\Lambda}(t_n;t_{n+1})=\mathcal{O}(1)$ since the velocity may exhibit finite jumps, whereas $\boldsymbol{U}(t_n;t_{n+1})=\mathcal{O}(h)$ and $\boldsymbol{\nu}(t_n;t_{n+1})=\mathcal{O}(h)$ since the position remain continuous. The position corrections eliminate the introduced drift, because the exact instant at which impacts develop is not known, and the velocity jumps are approximated over the time step.

It should be observed that Eqs. (3.31) can be further simplified by neglecting the terms which are multiplied by h and h^2, getting the following results:

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q}_{n+1})\;\boldsymbol{W}_{n+1}-\boldsymbol{g}_{\boldsymbol{q},n+1}^T\;\boldsymbol{\Lambda}_{n+1} &= \boldsymbol{0} &\text{(3.32a)}\\ \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\;\boldsymbol{U}_{n+1}-\boldsymbol{g}_{\boldsymbol{q},n+1}^T\;\boldsymbol{\nu}_{n+1} &= \boldsymbol{0} &\text{(3.32b)}\end{aligned}$$

By adopting this simplification, the order of the resulting integration algorithm would still be $\mathcal{O}(h)$. However, for some problems it is necessary to use very small stepsizes for obtaining the accuracy of an integrator based on Eqs. (3.31) as it will be shown in the numerical examples. It can be argued that the force term $\boldsymbol{f}$ is very sensitive to the position correction $\boldsymbol{U}$ for problems involving nonlinear flexible components, and it should be well represented to avoid needing small stepsizes. Therefore, the complete version of the jump equations in Eqs. (3.31) is retained.

The discrete complementarity conditions at velocity and at position levels are the same as those introduced in [50]:

$$\text{if } g^j(\boldsymbol{q}_{n+1}) \leq 0 \text{ then } 0 \leq g^j_{\boldsymbol{q},n+1}\,\boldsymbol{v}_{n+1} + e^j\, g^j_{\boldsymbol{q},n}\,\boldsymbol{v}_n \perp \Lambda^j_{n+1} \geq 0, \quad \forall j \in \mathcal{U} \tag{3.33a}$$

$$\boldsymbol{0} \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}_{n+1}) \perp \boldsymbol{\nu}^{\mathcal{U}}_{n+1} \geq \boldsymbol{0} \tag{3.33b}$$

The complete time integration scheme is obtained by combining this first-order approximation of the nonsmooth variables and equations, with a one-step and second-order time integration scheme for the smooth variables. The generalized-α method is used for the smooth part, although other DAE time integration schemes could also be considered. This hybrid time integration scheme is formulated to advance the solution at each step as follows.

Computation of the smooth motion

As previously proposed in Eqs. (3.10a–3.10c), the smooth motion is defined by a modified form of the equations of motion at time step $n+1$ where the contributions of the unilateral constraints and associated reaction forces are ignored, i.e.,

$$\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\dot{\tilde{\boldsymbol{v}}}_{n+1} - \boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1}) - \boldsymbol{g}^{\overline{\mathcal{U}},T}_{\tilde{\boldsymbol{q}},n+1}\left(k_s\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1} - p_s\boldsymbol{g}^{\overline{\mathcal{U}}}_{\tilde{\boldsymbol{q}},n+1}\,\tilde{\boldsymbol{v}}_{n+1}\right) = \boldsymbol{0} \tag{3.34a}$$

$$-k_s\boldsymbol{g}^{\overline{\mathcal{U}}}_{\tilde{\boldsymbol{q}},n+1}\,\tilde{\boldsymbol{v}}_{n+1} = \boldsymbol{0} \tag{3.34b}$$

where the Lagrange multiplier $\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}$ has been augmented with a penalty parameter $p_s \geq 0$ in order to add convexity to the objective function [5], and where $k_s > 0$ is a scaling factor for the Lagrange multiplier. The scaling factor k_s contributes to an improvement of the condition number of the iteration matrix yielding a better convergence rate.

These equations are completed with the difference equations:

$$\tilde{\boldsymbol{q}}_{n+1} = \boldsymbol{q}_n + h\boldsymbol{v}_n + h^2(0.5-\beta)\boldsymbol{a}_n + h^2\beta\boldsymbol{a}_{n+1} \tag{3.35a}$$

$$\tilde{\boldsymbol{v}}_{n+1} = \boldsymbol{v}_n + h(1-\gamma)\boldsymbol{a}_n + h\gamma\boldsymbol{a}_{n+1} \tag{3.35b}$$

$$(1-\alpha_m)\boldsymbol{a}_{n+1} + \alpha_m\boldsymbol{a}_n = (1-\alpha_f)\dot{\tilde{\boldsymbol{v}}}_{n+1} + \alpha_f\dot{\tilde{\boldsymbol{v}}}_n \tag{3.35c}$$

where $\boldsymbol{a}_{n+1}$ is a pseudo acceleration term that arises in the generalized-α integrator scheme [79]. The numerical coefficients γ, β, α_m, and α_f can be chosen to achieve a desired level of high-frequency dissipation, represented by spectral radius at infinity

$\rho_\infty \in [0,1]$, while minimizing unwanted low-frequency dissipation [48]:

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad \gamma = 0.5 + \alpha_f - \alpha_m, \quad \beta = 0.25(\gamma + 0.5)^2 \tag{3.36}$$

Equations (3.34-3.35) only involve the smooth position $\tilde{\boldsymbol{q}}_{n+1}$ and velocity $\tilde{\boldsymbol{v}}_{n+1}$ and are thus decoupled from the variables $\boldsymbol{W}_{n+1}$, $\boldsymbol{U}_{n+1}$, $\boldsymbol{q}_{n+1}$, and $\boldsymbol{v}_{n+1}$. Therefore, these five equations can be solved for the five variables $\tilde{\boldsymbol{q}}_{n+1}$, $\tilde{\boldsymbol{v}}_{n+1}$, $\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1}$, $\dot{\tilde{\boldsymbol{v}}}_{n+1}$ and $\boldsymbol{a}_{n+1}$ using a Newton-Raphson algorithm without any information about the other variables.

Computation of the position correction

After the computation of the smooth motion, the position correction $\boldsymbol{U}_{n+1}$ is computed in order to obtain a position $\boldsymbol{q}_{n+1}$ which satisfies the bilateral constraints $\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}_{n+1}) = \boldsymbol{0}$ and the non-penetration constraints $\boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}_{n+1}) \geq \boldsymbol{0}$. Using Eq. (3.31b) and the discrete complementarity condition (3.33b), this problem writes

$$\begin{aligned}
\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\boldsymbol{U}_{n+1} - h^2 \boldsymbol{f}^p_{n+1} - \boldsymbol{g}^T_{\boldsymbol{q},n+1}\,\boldsymbol{\nu}_{n+1} &= \boldsymbol{0} && (3.37a)\\
-\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}_{n+1}) &= \boldsymbol{0} && (3.37b)\\
\boldsymbol{0} \leq \boldsymbol{g}^{\mathcal{U}}(\boldsymbol{q}_{n+1}) \perp \boldsymbol{\nu}^{\mathcal{U}}_{n+1} &\geq \boldsymbol{0} && (3.37c)
\end{aligned}$$

An augmented Lagrangian approach as presented by Alart and Curnier [43] is adopted to solve this LCP. Accordingly, the augmented Lagrangian for the set of bilateral and unilateral constraints $\mathcal{C}$ of the sub-problem at position level is given by

$$\begin{aligned}
\mathcal{L}^{\mathcal{C}}_p(\boldsymbol{U}_{n+1}, \boldsymbol{\nu}_{n+1}) = &\sum_{j\in\mathcal{U}} \left[-k_p \nu^j_{n+1} g^j_{n+1} + \frac{p_p}{2} g^j_{n+1} g^j_{n+1} - \frac{1}{2p_p}\,\mathrm{dist}^2(\xi^j_{n+1}, \mathbb{R}^+) \right] \\
&+ \sum_{i\in\overline{\mathcal{U}}} \left[-k_p \nu^i_{n+1} g^i_{n+1} + \frac{p_p}{2} g^i_{n+1} g^i_{n+1} \right]
\end{aligned} \tag{3.38}$$

where $\boldsymbol{\xi}_{n+1} = k_p \boldsymbol{\nu}_{n+1} - p_p \boldsymbol{g}_{n+1}$ is the augmented Lagrange multiplier at position level with $k_p > 0$ a scaling factor and $p_p > 0$ a penalty coefficient, and where the distance of a point $\boldsymbol{z} \in \mathbb{R}^n$ to the convex set C is defined as $\mathrm{dist}(\boldsymbol{z}, C) = ||\boldsymbol{z} - \mathrm{prox}(\boldsymbol{z}, C)||$ with $\mathrm{prox}(\boldsymbol{z}, C) = \mathrm{argmin}_{\boldsymbol{z}^* \in C} \frac{1}{2}||\boldsymbol{z} - \boldsymbol{z}^*||^2$. The adoption of this augmented Lagrangian results in the following set of equations for the sub-problem at position level:

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q}_{n+1})\boldsymbol{U}_{n+1} - h^2 \boldsymbol{f}^p_{n+1} - \boldsymbol{g}^{\mathcal{A},T}_{\boldsymbol{q},n+1}\,\boldsymbol{\xi}^{\mathcal{A}}_{n+1} &= \boldsymbol{0} && (3.39a)\\
-k_p \boldsymbol{g}^{\mathcal{A}}_{n+1} &= \boldsymbol{0} && (3.39b)\\
-\frac{k_p^2}{p_p} \boldsymbol{\nu}^{\overline{\mathcal{A}}}_{n+1} &= \boldsymbol{0} && (3.39c)
\end{aligned}$$

where the active set $\mathcal{A} \equiv \mathcal{A}_{n+1}$ and its complement $\overline{\mathcal{A}} \equiv \overline{\mathcal{A}}_{n+1}$ are given by

$$\begin{aligned} \mathcal{A}_{n+1} &= \overline{\mathcal{U}} \cup \left\{ j \in \mathcal{U} : \xi^j_{n+1} \geq 0 \right\}, && (3.40a) \\ \overline{\mathcal{A}}_{n+1} &= \mathcal{C} \setminus \mathcal{A}_{n+1} && (3.40b) \end{aligned}$$

The terms associated to the constraints were obtained from the stationary condition $\delta \mathcal{L}_p^{\mathcal{C}} = 0$. The resulting set of equations can be solved in terms of the unknown variables $\boldsymbol{U}_{n+1}$, $\boldsymbol{q}_{n+1}$ and $\boldsymbol{\nu}_{n+1}$ using a Newton semi-smooth method.

Computation of the velocity jump

After the computation of the position field, the velocity jump $\boldsymbol{W}_{n+1}$ is computed such that the velocity $\boldsymbol{v}_{n+1}$ satisfies the bilateral constraints $\boldsymbol{g}_{\boldsymbol{q}}^{\overline{\mathcal{U}}} \boldsymbol{v}_{n+1} = \boldsymbol{0}$ and the impact law $g^j_{\boldsymbol{q},n+1} \boldsymbol{v}_{n+1} + e^j g^j_{\boldsymbol{q},n} \boldsymbol{v}_n \geq 0$ for all unilateral constraints $j \in \mathcal{U}$ that are active at position level, i.e., that satisfy $\xi^j_{n+1} \geq 0$. Using Eq. (3.31a) and the discrete complementarity condition (3.33a), the equations for this problem are given by

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}_{n+1}) \boldsymbol{W}_{n+1} - h \boldsymbol{f}^*_{n+1} - \boldsymbol{g}^T_{\boldsymbol{q},n+1} \boldsymbol{\Lambda}_{n+1} &= \boldsymbol{0} && (3.41a) \\ -\boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q},n+1} \boldsymbol{v}_{n+1} &= \boldsymbol{0} && (3.41b) \\ \text{if } g^j(\boldsymbol{q}_{n+1}) \leq 0 \text{ then } 0 \leq g^j_{\boldsymbol{q},n+1} \boldsymbol{v}_{n+1} + e^j g^j_{\boldsymbol{q},n} \boldsymbol{v}_n \perp \Lambda^j_{n+1} &\geq 0, && (3.41c) \\ & \quad \forall j \in \mathcal{U} \end{aligned}$$

In order to solve this LCP problem, we proceed in a similar manner as we did for the sub-problem at position level. However, the activation of a given unilateral constraint $j \in \mathcal{U}$ at velocity level depends on the activation condition $g^j(\boldsymbol{q}_{n+1}) \leq 0$, see Eq. (3.41c). This condition is equivalent to $\xi^j_{n+1} \geq 0$, which is more robust from the algorithmic point of view, and therefore adopted in this work. The augmented Lagrange multiplier at velocity level is defined by

$$\boldsymbol{\sigma}_{n+1} = k_v \boldsymbol{\Lambda}_{n+1} - p_v \overset{\circ}{\boldsymbol{g}}_{n+1} \qquad (3.42)$$

where $p_v > 0$ is the penalty parameter, $k_v > 0$ is the scaling factor for the Lagrange multiplier $\boldsymbol{\Lambda}$, and $\overset{\circ}{\boldsymbol{g}}_{n+1}$ is a notation for the impact law

$$\overset{\circ}{g}{}^j_{n+1} = g^j_{\boldsymbol{q},n+1} \boldsymbol{v}_{n+1} + e^j g^j_{\boldsymbol{q},n} \boldsymbol{v}_n \qquad (3.43)$$

which applies to every $j \in \mathcal{C}$. The coefficients associated to bilateral constraints are trivially defined as $e^j = 0 \; \forall \; j \in \mathcal{U}$. Then, the augmented Lagrangian for this case is given by

$$\begin{aligned} \mathcal{L}_v^{\mathcal{C}}(\boldsymbol{W}_{n+1}, \boldsymbol{\Lambda}_{n+1}) = & \sum_{j \in \mathcal{U}} \left[-k_v \Lambda^j_{n+1} \overset{\circ}{g}{}^j_{n+1} + \frac{p_v}{2} \overset{\circ}{g}{}^j_{n+1} \overset{\circ}{g}{}^j_{n+1} - \frac{1}{2 p_v} \operatorname{dist}^2(\sigma^j_{n+1}, \mathbb{R}^+) \right] \\ & + \sum_{i \in \overline{\mathcal{U}}} \left[-k_v \Lambda^i_{n+1} \overset{\circ}{g}{}^i_{n+1} + \frac{p_v}{2} \overset{\circ}{g}{}^i_{n+1} \overset{\circ}{g}{}^i_{n+1} \right] \end{aligned} \qquad (3.44)$$

This results in the following set of equations for the problem at velocity level:

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q}_{n+1})\boldsymbol{W}_{n+1} - h\boldsymbol{f}^{*}_{n+1} - \boldsymbol{g}^{\mathcal{B},T}_{\boldsymbol{q},n+1}\,\boldsymbol{\sigma}^{\mathcal{B}}_{n+1} &= \boldsymbol{0} && (3.45a)\\
-k_v \overset{\circ}{\boldsymbol{g}}{}^{\mathcal{B}}_{n+1} &= \boldsymbol{0} && (3.45b)\\
-\frac{k_v^2}{p_v}\boldsymbol{\Lambda}^{\overline{\mathcal{B}}}_{n+1} &= \boldsymbol{0} && (3.45c)
\end{aligned}$$

where the active set $\mathcal{B} \equiv \mathcal{B}_{n+1}$ and its complement $\overline{\mathcal{B}} \equiv \overline{\mathcal{B}}_{n+1}$ are given by

$$\begin{aligned}
\mathcal{B}_{n+1} &= \overline{\mathcal{U}} \cup \{ j \in \mathcal{A}_{n+1} : \sigma^{j}_{n+1} \geq 0 \} && (3.46a)\\
\overline{\mathcal{B}}_{n+1} &= \mathcal{C} \setminus \mathcal{B}_{n+1} && (3.46b)
\end{aligned}$$

The terms associated to the constraints were obtained from the stationary condition $\delta \mathcal{L}^{\mathcal{C}}_{v} = 0$. The resulting set of equations can be solved in terms of the unknown variables $\boldsymbol{W}_{n+1}$, $\boldsymbol{v}_{n+1}$ and $\boldsymbol{\Lambda}_{n+1}$ using a Newton semi-smooth method.

Global numerical procedure

The three sub-problems (3.34, 3.39, 3.45) need to be solved at each time step, one for the smooth motion, another for the position correction, and, lastly, one for the velocity jump. These are computations which can generally be organized in a sequential manner. In [50] and [71], the sub-problem defining the smooth variables involve the total position $\boldsymbol{q}_{n+1}$ and velocity $\boldsymbol{v}_{n+1}$ fields. As a consequence, some global iterations over the three sub-problems have to be implemented, which tend to penalize the numerical cost of the procedure. In contrast, the sub-problem defining the smooth motion is strictly independent of the position correction and velocity jump, and the problem at position level is independent of the velocity jump. In this way, the three sub-problems can be solved in a purely decoupled sequential manner.

Each sub-problem can be solved using a semi-smooth Newton method. The correction terms should satisfy the integration formulae, therefore, the corrections $\Delta\dot{\tilde{\boldsymbol{v}}}_{n+1}$ and $\Delta\boldsymbol{q}_{n}$ can be eliminated in terms of $\Delta\tilde{\boldsymbol{v}}_{n+1}$, $\Delta\boldsymbol{W}_{n+1}$ and $\Delta\boldsymbol{U}_{n+1}$:

$$\begin{aligned}
\Delta\boldsymbol{v}_{n+1} &= \Delta\tilde{\boldsymbol{v}}_{n+1} + \Delta\boldsymbol{W}_{n+1} && (3.47a)\\
\Delta\dot{\tilde{\boldsymbol{v}}}_{n+1} &= (1-\alpha_m)/((1-\alpha_f)\gamma h)\,\Delta\tilde{\boldsymbol{v}}_{n+1} && (3.47b)\\
\Delta\boldsymbol{q}_{n+1} &= h\,\beta/\gamma\,\Delta\tilde{\boldsymbol{v}}_{n+1} + \Delta\boldsymbol{U}_{n+1} && (3.47c)
\end{aligned}$$

Then, the vectors of independent corrections are given by

$$\Delta\boldsymbol{x}^{s} = \left\{ \begin{array}{c} \Delta\tilde{\boldsymbol{v}}_{n+1} \\ \Delta\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1} \end{array} \right\}, \quad \Delta\boldsymbol{x}^{p} = \left\{ \begin{array}{c} \Delta\boldsymbol{U}_{n+1} \\ \Delta\boldsymbol{\nu}^{\mathcal{A}}_{n+1} \\ \Delta\boldsymbol{\nu}^{\overline{\mathcal{A}}}_{n+1} \end{array} \right\}, \quad \Delta\boldsymbol{x}^{v} = \left\{ \begin{array}{c} \Delta\boldsymbol{W}_{n+1} \\ \Delta\boldsymbol{\Lambda}^{\mathcal{B}}_{n+1} \\ \Delta\boldsymbol{\Lambda}^{\overline{\mathcal{B}}}_{n+1} \end{array} \right\} \qquad (3.48)$$

and the correction equations for each sub-problem are obtained as

$$\boldsymbol{S}_t^i\,\Delta\boldsymbol{x}^i = -\boldsymbol{r}^i, \quad \text{for} \quad i = s, p, v \tag{3.49}$$

where $\boldsymbol{r}^s$, $\boldsymbol{r}^p$ and $\boldsymbol{r}^v$ have been used to denote the residuals of Eqs. (3.34), (3.39) and (3.45), respectively, and where the iteration matrices are given by

$$\boldsymbol{S}_t^s = \begin{bmatrix} \boldsymbol{S}_t^{s^*} & -k_s \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T} \\ -k_s\left(\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}} + \frac{h\beta}{\gamma}\boldsymbol{G}^s\right) & \boldsymbol{0} \end{bmatrix} \tag{3.50}$$

$$\boldsymbol{S}_t^p = \begin{bmatrix} \boldsymbol{S}_t^{p^*} & -k_p \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{A},T} & \boldsymbol{0} \\ -k_p \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{A}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\frac{k_p^2}{p_p}\boldsymbol{I}^{\overline{\mathcal{A}}} \end{bmatrix} \tag{3.51}$$

$$\boldsymbol{S}_t^v = \begin{bmatrix} \boldsymbol{S}_t^{v^*} & -k_v \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{B},T} & \boldsymbol{0} \\ -k_v \boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{B}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\frac{k_v^2}{p_v}\boldsymbol{I}^{\overline{\mathcal{B}}} \end{bmatrix} \tag{3.52}$$

where $\boldsymbol{I}^{\overline{\mathcal{A}}}$ and $\boldsymbol{I}^{\overline{\mathcal{B}}}$ are identity matrices and

$$\begin{aligned}
\boldsymbol{S}_t^{s^*} &= \frac{1-\alpha_m}{h(1-\alpha_f)\gamma}\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1}) + \boldsymbol{C}_t + \frac{h\beta}{\gamma}\boldsymbol{K}_t^s, \\
\boldsymbol{K}_t^s &= \frac{\partial\left(\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\dot{\tilde{\boldsymbol{v}}}_{n+1} - \boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T}\left(k_s\tilde{\boldsymbol{\lambda}}_{n+1}^{\overline{\mathcal{U}}} - p_s\,\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}}\,\tilde{\boldsymbol{v}}_{n+1}\right) - \boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})\right)}{\partial\tilde{\boldsymbol{q}}_{n+1}}, \\
\boldsymbol{C}_t &= p_s\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}},T}\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}^{\overline{\mathcal{U}}} - \frac{\partial\boldsymbol{f}(\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})}{\partial\tilde{\boldsymbol{v}}_{n+1}}, \quad \boldsymbol{G}^s = \frac{\partial(\boldsymbol{g}_{\tilde{\boldsymbol{q}},n+1}\tilde{\boldsymbol{v}}_{n+1})}{\partial\tilde{\boldsymbol{q}}_{n+1}}, \\
\boldsymbol{S}_t^{p^*} &= \boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1}) - \frac{\partial\left(\boldsymbol{g}_{\boldsymbol{q},n+1}^{\mathcal{A},T}\boldsymbol{\xi}_{n+1}^{\mathcal{A}}\right)}{\partial\boldsymbol{q}_{n+1}} - h^2\frac{\partial\boldsymbol{f}^p(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})}{\partial\boldsymbol{q}_{n+1}}, \\
\boldsymbol{S}_t^{v^*} &= \boldsymbol{M}(\boldsymbol{q}_{n+1}) + p_v\boldsymbol{g}_{\boldsymbol{q},n+1}^{\overline{\mathcal{B}},T}\boldsymbol{g}_{\boldsymbol{q},n+1}^{\overline{\mathcal{B}}} - h\frac{\partial\boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1})}{\partial\boldsymbol{v}_{n+1}}
\end{aligned} \tag{3.53}$$

As mentioned before, the solution does not depend on the value of parameters k_s, k_p, k_v, p_s, p_p and p_v. Nevertheless, the matrix conditioning and convergence rate do depend on their values, and therefore they are determined as in [67]:

$$k_s = p_s = \frac{\bar{m}}{h}, \quad k_p = p_p = \bar{m}, \quad k_v = p_v = \bar{m} \tag{3.54}$$

where $\bar{m}$ is a characteristic mass of the problem.

The integrator is summarized in Algorithm 1. The criterion used for checking the Newton scheme convergence in each sub-problem is denoted simply as $||\boldsymbol{r}^i|| < \text{tol}$, for $i = s, p, v$. However, the actual expression of the convergence criterion is given by

$$||\boldsymbol{r}^i|| < \mathrm{tol}_r \left(\sum_k ||\boldsymbol{r}^i_k|| + \mathrm{tol}_f \right) \tag{3.55}$$

where tol_r is a given relative tolerance, $\boldsymbol{r}^i_k$ is the k-th term contributing to the residual $\boldsymbol{r}^i$, tol_f is a reference value of tolerance and $|| \cdot ||$ is the L^2 norm of $\cdot$.

Algorithm 1 Decoupled nonsmooth generalized-α time integration scheme

```
Inputs: initial values $\boldsymbol{q}_0$ and $\boldsymbol{v}_0$;
Compute consistent value of $\dot{\tilde{\boldsymbol{v}}}_0$
$\boldsymbol{a}_0 := \dot{\tilde{\boldsymbol{v}}}_0$
for $n = 0$ to $n_{\text{final}} - 1$ do
    $\dot{\tilde{\boldsymbol{v}}}_{n+1} := \boldsymbol{0}$, $\tilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1} := \boldsymbol{0}$, $\boldsymbol{\nu}_{n+1} := \boldsymbol{0}$
    $\boldsymbol{\Lambda}_{n+1} := \boldsymbol{0}$, $\boldsymbol{U}_{n+1} := \boldsymbol{0}$, $\boldsymbol{W}_{n+1} := \boldsymbol{0}$
    $\boldsymbol{a}_{n+1} := 1/(1-\alpha_m)(\alpha_f \dot{\tilde{\boldsymbol{v}}}_n - \alpha_m \boldsymbol{a}_n)$
    $\boldsymbol{v}_{n+1} := \widetilde{\boldsymbol{v}}_{n+1} := \boldsymbol{v}_n + h(1-\gamma)\boldsymbol{a}_n + h\gamma\boldsymbol{a}_{n+1}$
    $\boldsymbol{q}_{n+1} := \boldsymbol{q}_n + h\boldsymbol{v}_n + h^2(1/2\ -\beta)\boldsymbol{a}_n + h^2\beta\boldsymbol{a}_{n+1}$
    Step 1 (smooth motion):
    for $i = 1$ to $i_{max}$ do
        Compute residual $\boldsymbol{r}^s$
        if $||\boldsymbol{r}^s|| <$ tol then break
         end if
        Compute the iteration matrix $\boldsymbol{S}_t^s$
        $\Delta\boldsymbol{x}^s := -(\boldsymbol{S}_t^s)^{-1}\boldsymbol{r}^s$
        $\tilde{\boldsymbol{v}}_{n+1} := \tilde{\boldsymbol{v}}_{n+1} + \Delta\tilde{\boldsymbol{v}}$
        $\dot{\tilde{\boldsymbol{v}}}_{n+1} := \dot{\tilde{\boldsymbol{v}}}_{n+1} + (1-\alpha_m)/((1-\alpha_f)\gamma h)\Delta\tilde{\boldsymbol{v}}$
        $\boldsymbol{q}_{n+1} := \boldsymbol{q}_{n+1} + h\beta/\gamma\Delta\tilde{\boldsymbol{v}}$
        $\widetilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1} := \widetilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}_{n+1} + \Delta\widetilde{\boldsymbol{\lambda}}^{\overline{\mathcal{U}}}$
    end for
    Step 2 (projection on position constraints):
    for $i = 1$ to $i_{max}$ do
        Compute residual $\boldsymbol{r}^p$
        if $||\boldsymbol{r}^p|| <$ tol then break
         end if
        Compute $\boldsymbol{S}_t^p$
        $\Delta\boldsymbol{x}^p := -(\boldsymbol{S}_t^p)^{-1}\boldsymbol{r}^p$
        $\boldsymbol{U}_{n+1} := \boldsymbol{U}_{n+1} + \Delta\boldsymbol{U}$
        $\boldsymbol{q}_{n+1} := \boldsymbol{q}_{n+1} + \Delta\boldsymbol{U}$
        $\boldsymbol{\nu}_{n+1} := \boldsymbol{\nu}_{n+1} + \Delta\boldsymbol{\nu}$
    end for
    Step 3 (projection on velocity constraints):
    for $i = 1$ to $i_{max}$ do
        Compute residual $\boldsymbol{r}^v$
        if $||\boldsymbol{r}^v|| <$ tol then break
         end if
        Compute $\boldsymbol{S}_t^v$
        $\Delta\boldsymbol{x}^v := -(\boldsymbol{S}_t^v)^{-1}\boldsymbol{r}^v$
        $\boldsymbol{W}_{n+1} := \boldsymbol{W}_{n+1} + \Delta\boldsymbol{W}$
        $\boldsymbol{v}_{n+1} := \tilde{\boldsymbol{v}}_{n+1} + \boldsymbol{W}_{n+1}$
        $\boldsymbol{\Lambda}_{n+1} := \boldsymbol{\Lambda}_{n+1} + \Delta\boldsymbol{\Lambda}$
    end for
    $\boldsymbol{a}_{n+1} := \boldsymbol{a}_{n+1} + (1-\alpha_f)/(1-\alpha_m)\dot{\tilde{\boldsymbol{v}}}_{n+1}$
end for
```

3.3 Numerical examples

Four numerical examples are considered to study the accuracy and robustness of the proposed methodology, Fig. 3.1. Special emphasis is made on showing the improvements brought to the original NSGA [50]. The average and the maximum number of iterations per time step as well as the rate of convergence with each integrator, are compared. In the case of the original NSGA, the reported number of iterations is actually computed as the number of times that a linearized system of equations has to be solved per global time step. The results obtained with the original NSGA are denoted by CS (Coupled-Solution), whilst the ones resulting from the new algorithm are denoted by DS (Decoupled-Solution). The convergence rates are computed with an error evaluated with the L^1 norm:

$$\text{Error}(h) = \frac{\sum_{n=0}^{N} |f_n - f(t_n)|}{\sum_{n=0}^{N} |f(t_n)|} \tag{3.56}$$

where N is the number of time steps, f_n is the numerical solution and $f(t_n)$ is the reference solution. The reference solution is taken as the numerical solution for a very small stepsize using the original NSGA. The tolerances tol_r and tol_f in the convergence criterion of the Newton solver, are both adopted equal to 10^{-5}. The spectral radius is taken as $\rho_\infty = 0.8$.

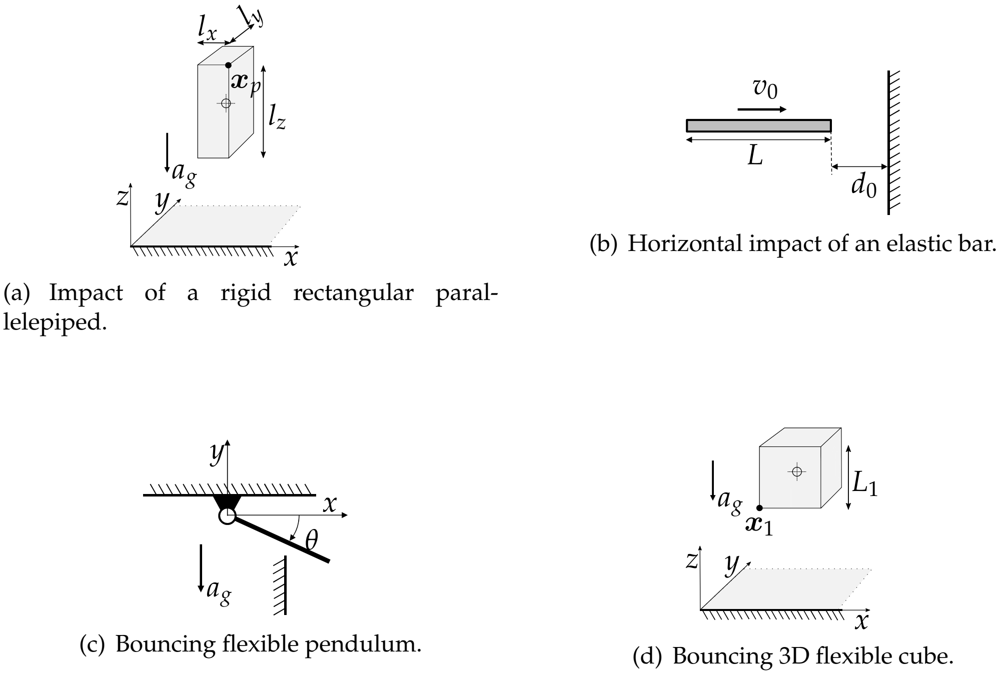
Three of the four examples involve a spatial discretization. When performing the convergence study for these examples, the number of elements is kept constant and only the time stepsize is varied.

Both integrators have been implemented in the finite element code Oofelie [66]. The finite elements for flexible multibody systems are described in [5] and the coordinates are the nodal absolute translations and rotations. The discretization of rotation variables relies on a Lie group time integration method directly adapted from [78].

The examples have been chosen to highlight important aspects of the proposed strategy. In the first example, the impact of a rigid rectangular parallelepiped body with a large initial angular velocity is solved to show how the presence of gyroscopic forces and of the coupling stemming from nonlinear bilateral constraints (which represent the rigid body) influence the solvers performance. Next, the horizontal impact of an elastic bar is considered to study the ability of the proposed algorithm to deal with flexibility. In the third example, the behaviour of the integrators for problems with nonlinear flexible beams is considered by studying the bouncing of a flexible pendulum. Lastly, the bouncing of a 3D flexible cube is examined to investigate the performance of the method for 3D nonlinear finite element models.

3.3.1 Impact of a rigid rectangular parallelepiped body

This example consists in the impact of a rigid rectangular parallelepiped body on a rigid support, Fig. 3.1(a). This problem does not involve flexibility, but is subject



(a) Impact of a rigid rectangular parallelepiped.

(b) Horizontal impact of an elastic bar.

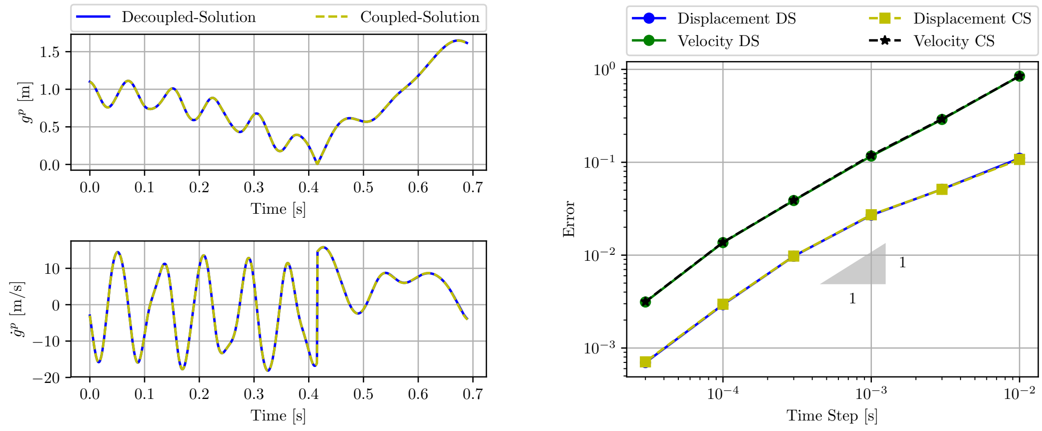
(c) Bouncing flexible pendulum.

(d) Bouncing 3D flexible cube.

FIGURE 3.1: Numerical examples to assess the performance of the proposed solver.

to nonlinear bilateral constraints that model the rigidity of the body. The nonlinear bilateral constraints couple the smooth problem to the sub-problem at position level in the CS algorithm, influencing the required number of iterations for convergence. A large initial angular velocity is imposed, making it necessary to accurately account for gyroscopic effects. The important contribution of gyroscopic forces allows to evaluate the appropriateness of the approximation $\boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1}) \approx \boldsymbol{f}(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})$ at the position level in the DS solver.

The parameters of the problem are $m = 46.8$ kg, $l_x = 0.1$ m, $l_y = 0.2$ m, $l_z = 0.3$ m, $a_g = 9.81$ m/s^2 and restitution coefficient $e = 0.8$. The inertia tensor $\boldsymbol{J}$ is diagonal with entries $J_{xx} = m(l_y^2 + l_z^2)/12$, $J_{yy} = m(l_x^2 + l_z^2)/12$ and $J_{zz} = m(l_y^2 + l_x^2)/12$. The body center of mass has initial conditions: angular velocity $\boldsymbol{\Omega}_0 = [60, 60, 0]^T$ rad/s, translation velocity $\boldsymbol{v}_0 = [3, 0, 0]^T$ m/s and initial position $\boldsymbol{x}_0 = [0, 0, 1]^T$ m. The reference solution is obtained with a stepsize of $h = 10^{-5}$ s using the CS solver. The z-displacement and velocity of node $\boldsymbol{x}_p$ for a stepsize of $h = 10^{-3}$ s is plotted in Fig. 3.2(a), whilst in Fig. 3.2(b) the convergence analysis of both integrators is shown. As it can be seen, both methods converge with the same convergence rate and error. However, the average and the maximum number of iterations, Figs. 3.3(a-b), make clear that the new splitting strategy of DS is more robust than CS for problems with nonlinear bilateral constraints and important gyroscopic contributions. In addition, we observe that the approximation $\boldsymbol{f}(\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}, t_{n+1}) \approx \boldsymbol{f}(\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}, t_{n+1})$ at position level in the DS solver does not affect the accuracy of the obtained results for the considered example.



(a) Solution obtained for both solvers for $h = 10^{-3}$ s. Showing the gap and its time derivative for the node $\boldsymbol{x}_p$, see Fig. 3.1(a).

(b) Convergence analysis performed for the time window $t = (0, 0.69)$ s for the z component of the displacement and the velocity fields of node $\boldsymbol{x}_p$.

FIGURE 3.2: Results obtained for the impact of a rigid rectangular parallelepiped body.

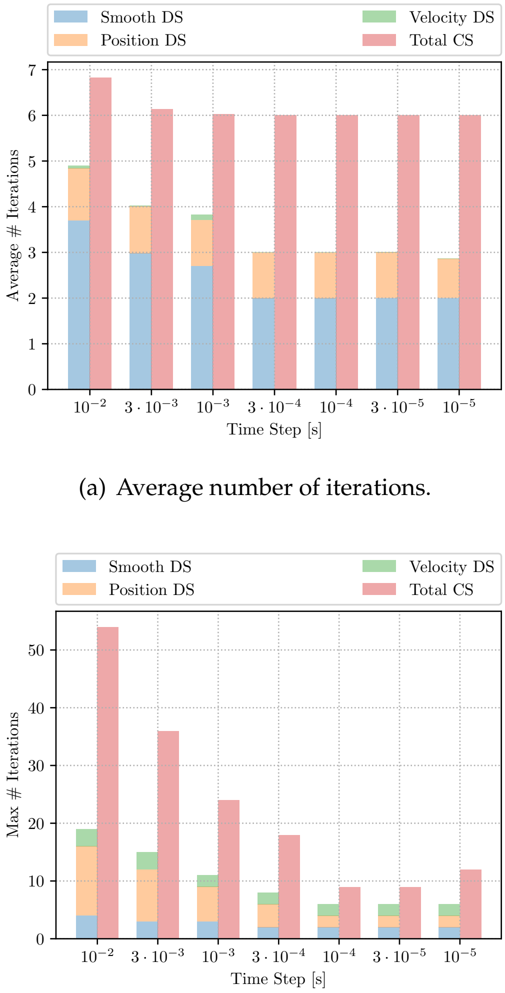
3.3.2 Horizontal impact of an elastic bar

In this section, the horizontal impact of an elastic bar is analyzed, see Fig. 3.1(b). This example deals with a linear flexible model with no bilateral constraints, allowing to understand convergence problems associated to flexibility. It should be noted that in the case of the CS solver, the smooth sub-problem is coupled to the position sub-problem through the internal elastic forces which depend on the total position $\boldsymbol{q}$.

The bar starts moving from a distance d_0 in an undeformed configuration with a uniform initial velocity field v_0, and bounces back after impacting a rigid wall. The impact is horizontal and no gravity is considered [50]. The parameters of this test are: Young modulus $E = 900$ Pa, density $\rho = 1$ kg/m^3, undeformed initial length $L = 10$ m, initial distance from the wall $d_0 = 5.005$ m, initial velocity $v_0 = 10$ m/s and restitution coefficient $e = 0$. The bar is discretized uniformly using 200 beam finite elements. The reference solution is computed with a time step of $h = 10^{-5}$ s using the CS solver. Three different nonsmooth integrators are tested: the original NSGA (CS) solver, the decoupled NSGA based on Eqs. (3.31a, 3.31b), and the decoupled NSGA based on Eqs. (3.32a, 3.32b). The solutions obtained with the two latter options are respectively denoted by DS and DSn (Decoupled-Solution neglecting the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms).

The average and maximum number of iterations for each strategy can be observed in Figs. 3.4(a-b). Since the problem is linear, the maximum number of iterations in the DS and DSn strategies is only one per sub-problem for any value of stepsize h. On the other hand, the number of iterations required by the original solver (CS) is quite high, and it even diverged for stepsizes greater than or equal to $h = 10^{-2}$ s. Therefore, the advantage of the proposed splitting becomes quite clear.

Figures 3.5(a-b) show the convergence analysis for two nodes of the bar (one at



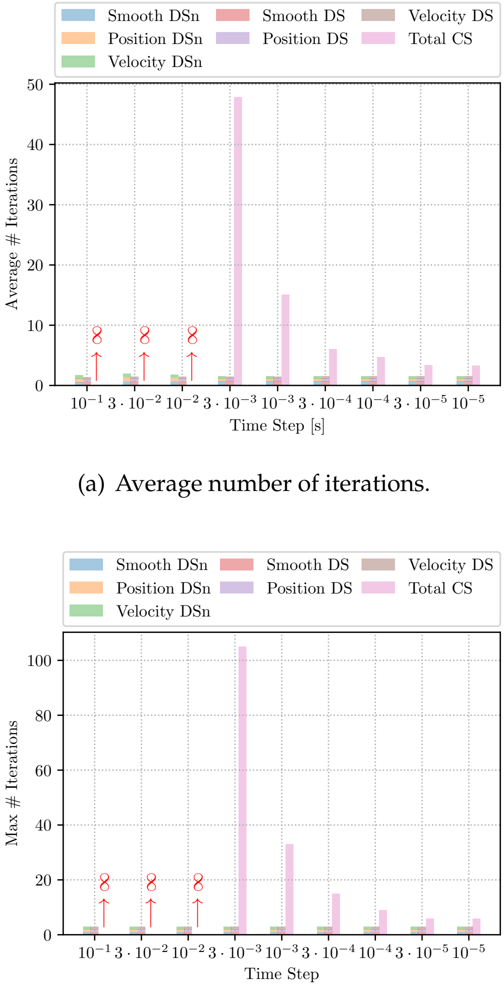
(a) Average number of iterations.

(b) Maximum number of iterations.

FIGURE 3.3: Rigid rectangular parallelepiped problem: number of iterations taken by the DS and CS solvers.

the tip and another located 0.45 m from the tip). It can be seen that the robustness improvement in the DS strategy is obtained without damaging the convergence rate neither the accuracy. However, in the decoupled strategy DSn, where the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms were neglected, the error is larger than for the DS strategy for moderately large stepsizes. A comparison of the results of DSn and DS for different stepsizes is shown in Figs. 3.6(a-b).

It is important to highlight that for moderately large stepsizes, the DSn solution does not reproduce certain key aspects of the physical solution, such as, for instance, the bouncing back ot the bar ($h = 0.03$ s), or a quite long delay in the bouncing back with respect to the reference solution ($h = 0.01$ s), Fig. 3.6(a). In addition, the contact forces deviate considerably from what is expected. Therefore, these results indicate that the term $\boldsymbol{f}(\boldsymbol{q}, \tilde{\boldsymbol{v}}, t_{n+1}) - \boldsymbol{f}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}, t_{n+1}) \approx \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \boldsymbol{U}_{n+1}$ has an important contribution to the position correction for large values of h and cannot be neglected. Another error is evidenced by observing the computed displacement and velocity at the node located



(a) Average number of iterations.

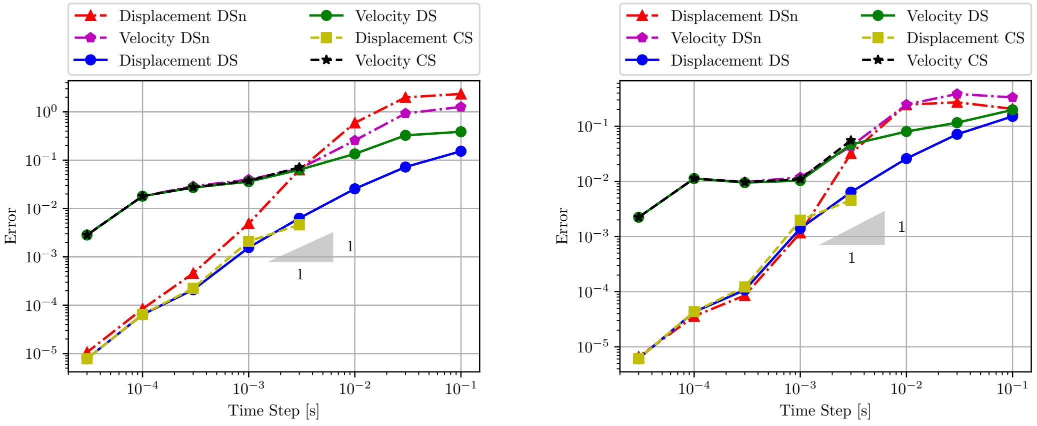
(b) Maximum number of iterations.

FIGURE 3.4: Horizontal impact of an elastic bar: number of iterations taken by the DS and CS solvers.

at 0.45 m from the tip of the bar, Fig. 3.6(b). When neglecting the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms for $h = 0.03$ s, the computed solution satisfies the non-penetration condition at the tip node, Fig. 3.6(a), but the node at 0.45 m from the tip penetrates the wall, see Fig. 3.6(b).

3.3.3 Bouncing of a flexible pendulum

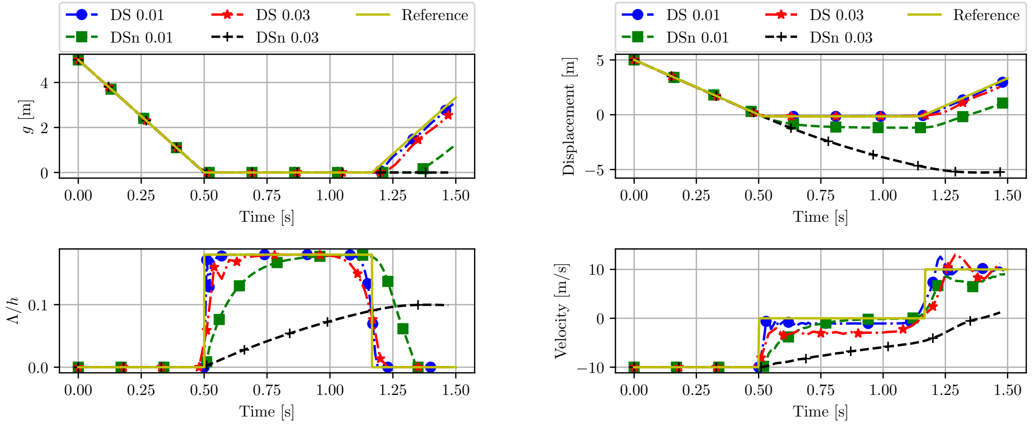
The bouncing of a flexible beam pendulum hitting an obstacle is next studied, Fig. 3.1(c). This test allows to assess the performance of the integrators for problems involving nonlinear flexible beams. The pendulum is constrained to swing in the x-y plane around a pivot located at the origin. The properties of the beam are: undeformed length $L = 1$ m, cross-section area $A = 10^{-4}$ m^2, cross-section inertia $I = 8.33\ 10^{-10}$ m^4, shear section area $A_s = 5/6A$, Young modulus $E = 2.1\ 10^{11}$ N/m^2, density $\rho = 7800$ kg/m^3 and Poisson ratio $\nu = 0.3$. The beam is in horizontal position at the initial configuration with zero velocity, and begins to fall under the action



(a) Convergence analysis performed for the time window $t = (0, 1.5)$ s for the horizontal displacement and velocity of the node located at 0.45 m from the tip of the bar.

(b) Convergence analysis performed for the time window $t = (0, 1.5)$ s for the horizontal displacement and velocity of the tip node of the bar.

FIGURE 3.5: Convergence analysis for the horizontal impact of an elastic bar.



(a) Gap g and contact force Λ/h at the tip of the bar.

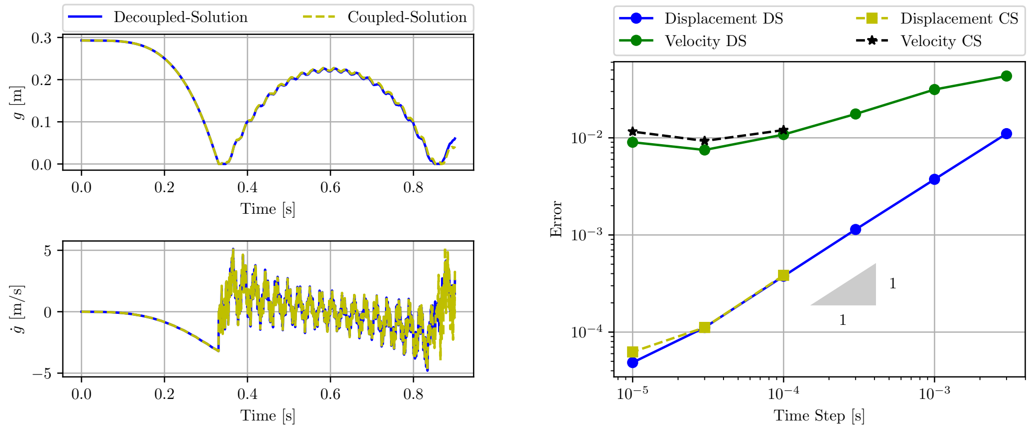
(b) Displacement and velocity of the node located at 0.45 m from the tip of the bar.

FIGURE 3.6: Comparison of the results obtained for different stepsizes using the decoupled solver with (DSn) and without (DS) neglecting the $\boldsymbol{f}^p$ and $\boldsymbol{f}^v$ terms.

of gravity $a_g = 9.81$ m/s^2. The tip of the beam hits a rigid wall located at $x = \sqrt{2}/2$, with a coefficient of restitution $e = 0$.

The beam is discretized into 8 equally-spaced nonlinear beam elements. The unilateral constraint representing the impact condition is $0 \leq g^{\mathcal{U}} = x - \sqrt{2}/2 \perp \mathrm{d}i^{\mathcal{U}} \geq 0$ and is enforced at the tip node. The reference solution is computed with a stepsize of $h = 10^{-6}$ s by using the original NSGA (CS) solver.

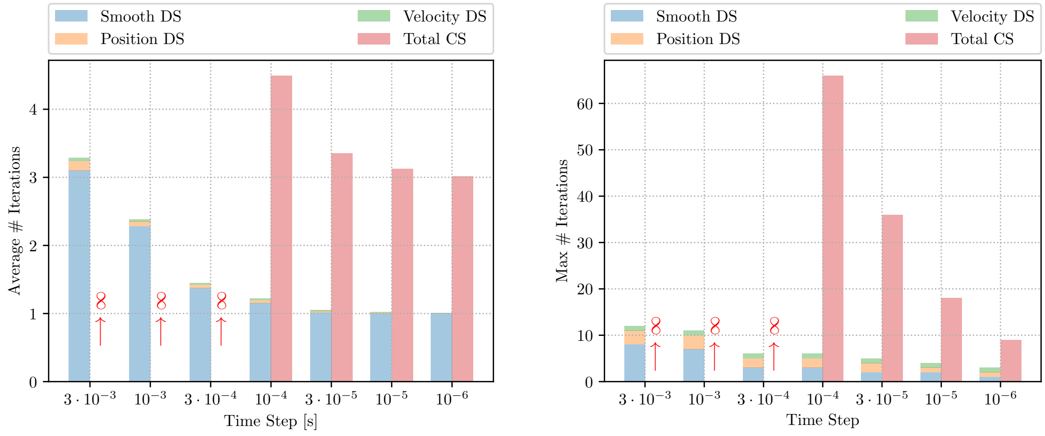
The y component of the displacement and the velocity at the tip of the beam calculated using $h = 10^{-4}$ s for both solvers are plotted in Fig. 3.7(a). The average and



(a) Solution obtained for both solvers for $h = 10^{-4}$ s. Showing the y component of the displacement and the velocity fields at the tip of the beam.

(b) Convergence analysis performed for the time window $t = (0, 0.35)$ s for the y component of the displacement and the velocity of the tip node of the pendulum.

FIGURE 3.7: Results obtained for the bouncing flexible pendulum.



(a) Average number of iterations.

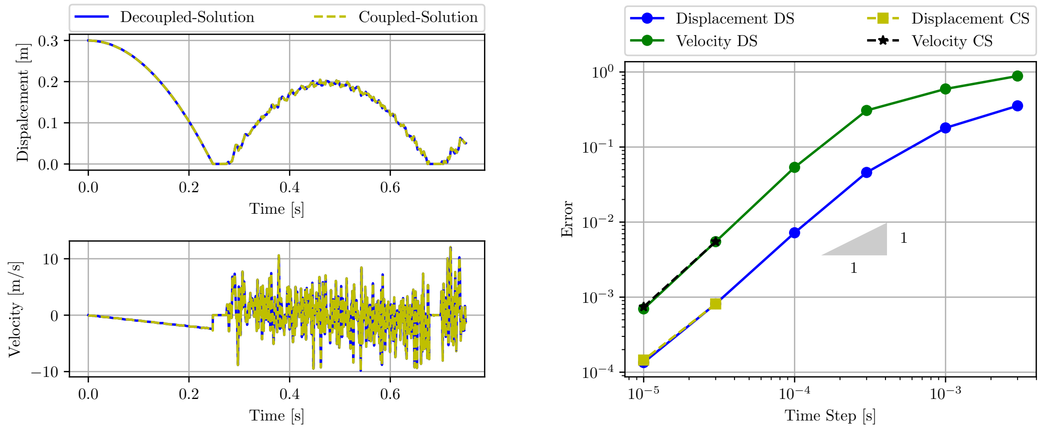
(b) Maximum number of iterations.

FIGURE 3.8: Bouncing flexible pendulum: number of iterations taken by the DS and CS solvers.

maximum number of iterations are shown in Figs. 3.8(a-b). It can be observed that for the three largest stepsizes, the original NSGA does not converge, Fig. 3.8. In addition, when it converges for smaller stepsizes, it takes a large number of iterations at impact events (see the maximum number of iterations). On the contrary, the proposed decoupled solver (DS) requires a small number of iterations for convergence.

3.3.4 Bouncing of a 3D flexible cube

In what follows a 3D example is considered. It consists of a 3D flexible cube which bounces against a rigid plane due to the action of gravity $a_g = 9.81$ m/s^2, Fig. 3.1(d). The cube is discretized into 125 equal tri-linear hexahedral geometrically nonlinear

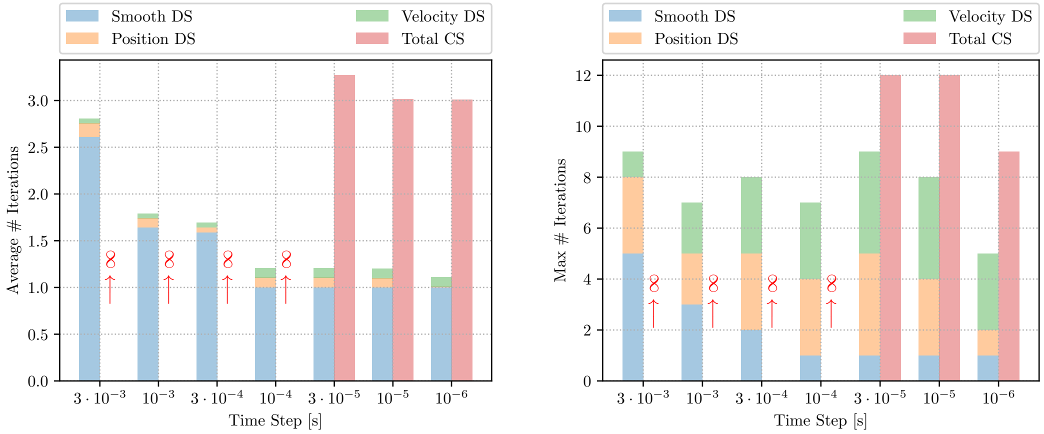


(a) Solution obtained for both solvers for $h = 3 \cdot 10^{-5}$ s. Showing the z component of the displacement and the velocity fields at node $\boldsymbol{x}_1$, see Fig. 3.1(d).

(b) Convergence analysis performed for the time window $t = (0, 0.3)$ s.

FIGURE 3.9: Results obtained for the bouncing of a 3D flexible cube.

finite elements. It is assumed that the material responds linearly. The cube has an undeformed side length $L_1 = 0.1$ m, the Young modulus is $E = 90$ Pa, the density is $\rho = 1$ kg/m^3, the Poisson ratio is $\nu = 0.3$ and the centroid of the cube is initially located at 0.35 m from the floor with a zero initial velocity. The restitution coefficient is $e = 0$. Contact elements are defined between the floor plane and each node on the cube face.



(a) Average number of iterations.

(b) Maximum number of iterations.

FIGURE 3.10: Bouncing 3D flexible cube: number of iterations taken by the DS and CS solvers.

The z component of the displacement and the velocity at node $\boldsymbol{x}_1$, computed using $h = 3 \cdot 10^{-5}$ s for both solvers, are plotted in Fig. 3.9(a). The convergence analysis for both solvers is shown in Fig. 3.9(b). As it can be observed, a convergence rate close to order 1 is achieved for the proposed solver (DS). The original NSGA

(CS) algorithm also exhibits order 1 with the same accuracy, but only for stepsizes smaller than or equal to $3 \cdot 10^{-5}$ s. For large stepsizes, the CS solver does not converge (Figs. 3.10(a-b)). The proposed DS solver deals well with flexible problems characterized by large displacements for any time stepsize. On the contrary, the original NSGA (CS) requires adopting a very small stepsize to get convergence.

3.4 Summary and concluding remarks

In this chapter, a fully decoupled nonsmooth generalized-α integration method was presented. Like its predecessor, it does not suffer from any drift phenomenon as it imposes the constraints both at position and at velocity levels. Additionally, it is well suited for problems with vibration effects as it integrates the smooth component of the motion with the second order accurate generalized-α method.

The algorithm was implemented as a sequence of three sub-problems to be solved at each time step. The most distinctive feature of the new algorithm was that the sub-problem defining the smooth part of the motion is strictly independent of the position correction and of the velocity jump, so that the solution of the three sub-problems could be performed in a purely decoupled sequential manner.

Four numerical examples were presented, showing that the proposed method improves the robustness for problems involving nonlinear bilateral constraints and/ or flexible elements, without deteriorating the accuracy of the original NSGA method. The number of iterations was reduced and much larger time steps could be adopted.

A variant of the new method, in which the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms were neglected, was analyzed in the examples. The computed results showed that neglecting those terms could lead to results of bad quality if sufficiently small stepsizes were not adopted. Hence, it was recommended to take into account those terms in the implementation of the decoupled algorithm. The application of this new algorithm to deal with frictional contact problems is studied in the next chapter.

Chapter 4

A nonsmooth frictional contact formulation for MBS dynamics

This chapter presents a new node-to-face frictional contact element to simulate the nonsmooth dynamics of systems composed of rigid and flexible bodies connected by kinematic joints. Its content is adapted from the paper [81]. The equations of motion are integrated using the NSGA time integration scheme presented in Chapter 3. The frictional contact problem is formulated using a mixed approach, based on an augmented Lagrangian technique proposed by Alart and Curnier for quasi static problems [43] and a Coulomb's friction law. The numerical results are independent of the user-defined penalty parameter for the normal or tangential components. The bilateral and the unilateral constraints are exactly fulfilled both at position and velocity levels. Finally, the robustness and the performance of the proposed element is demonstrated by solving several numerical examples of nonsmooth mechanical systems involving frictional contact.

4.1 Frictional contact formulation

The nonlinear Finite Element Method (FEM) is used in this work to analyze multibody systems composed of rigid and/or flexible elements and kinematic joints with different types of loading. By using the FEM approach, there is no distinction between the global coordinates of the rigid or flexible bodies. Therefore, the nonlinear effects are described with respect to a unique inertial frame [5].

As shown in Section 2.7, impacts and frictional phenomena in dynamic systems can be concisely represented by differential inclusions. Following this approach, for systems involving frictional impacts, Eqs. (3.37, 3.41) at position and velocity levels can be rewritten as

$$
\begin{aligned}
\boldsymbol{M}(\tilde{\boldsymbol{q}}_{n+1})\boldsymbol{U}_{n+1} - h^2 \boldsymbol{f}^{p}_{n+1} - \boldsymbol{g}^{T}_{\boldsymbol{q},n+1}\, \boldsymbol{\nu}_{n+1} &= \mathbf{0} && \text{(4.1a)}\\
-\boldsymbol{g}^{\overline{\mathcal{U}}}(\boldsymbol{q}_{n+1}) &= \mathbf{0} && \text{(4.1b)}\\
-g^{j}_{N}(\boldsymbol{q}_{n+1}) &\in \partial\psi_{\mathbb{R}^+}(\nu^{j}_{N,n+1}), \forall j \in \mathcal{U} && \text{(4.1c)}\\
-\boldsymbol{g}^{j}_{T}(\boldsymbol{q}_{n+1}) &\in \partial\psi_{C(\nu^{j}_{N,n+1})}(\boldsymbol{\nu}^{j}_{T,n+1}), \forall j \in \mathcal{U} && \text{(4.1d)}
\end{aligned}
$$

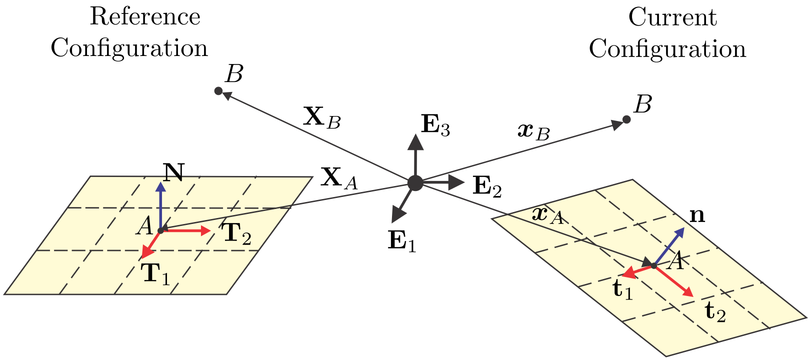


FIGURE 4.1: Node-to-face contact element.

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q}_{n+1})\boldsymbol{W}_{n+1} - h\boldsymbol{f}^*_{n+1} - \boldsymbol{g}^T_{\boldsymbol{q},n+1}\boldsymbol{\Lambda}_{n+1} &= \mathbf{0} && \text{(4.2a)}\\
-\boldsymbol{g}^{\overline{\mathcal{U}}}_{\boldsymbol{q},n+1}\boldsymbol{v}_{n+1} &= \mathbf{0} && \text{(4.2b)}\\
\text{if } g^j_N(\boldsymbol{q}_{n+1}) \leq 0 \text{ then } -\overset{\circ}{g}{}^j_{N,n+1} &\in \partial\psi_{\mathbb{R}^+}(\Lambda^j_{N,n+1}), \forall\, j \in \mathcal{U} && \text{(4.2c)}\\
\text{if } g^j_N(\boldsymbol{q}_{n+1}) \leq 0 \text{ then } -\overset{\circ}{\boldsymbol{g}}{}^j_{T,n+1} &\in \partial\psi_{C(\Lambda^j_{N,n+1})}(\boldsymbol{\Lambda}^j_{T,n+1}), \forall\, j \in \mathcal{U} && \text{(4.2d)}
\end{aligned}$$

By observing Eqs. (3.39, 3.45), the terms that are related with the frictionless unilateral constraints can be compiled in the following global generalized internal force vectors at position and velocity levels, respectively,

$$\boldsymbol{F}^p_G(\boldsymbol{q}_{n+1},\ \boldsymbol{\nu}_{n+1}) = \begin{Bmatrix} -\boldsymbol{g}^{\mathcal{A},T}_{\boldsymbol{q},n+1}\,\boldsymbol{\xi}^{\mathcal{A}}_{n+1} \\ -k_p\boldsymbol{g}^{\mathcal{A}} \\ -\frac{k_p^2}{p_p}\boldsymbol{\nu}^{\overline{\mathcal{A}}}_{n+1} \end{Bmatrix} \tag{4.3}$$

$$\boldsymbol{F}^v_G(\boldsymbol{v}_{n+1},\ \boldsymbol{\Lambda}_{n+1}) = \begin{Bmatrix} -\boldsymbol{g}^{\mathcal{B},T}_{\boldsymbol{q},n+1}\,\boldsymbol{\sigma}^{\mathcal{B}}_{n+1} \\ -k_v\overset{\circ}{\boldsymbol{g}}{}^{\mathcal{B}} \\ -\frac{k_v^2}{p_v}\boldsymbol{\Lambda}^{\overline{\mathcal{B}}}_{n+1} \end{Bmatrix} \tag{4.4}$$

From Eqs. (3.51, 3.52) the global tangent matrix at position and velocity levels are

$$\boldsymbol{K}^p_G = \begin{bmatrix} -\boldsymbol{g}^{\mathcal{A},T}_{\boldsymbol{qq},n+1}\,\boldsymbol{\xi}^{A}_{n+1} & -k_p\boldsymbol{g}^{\mathcal{A},T}_{\boldsymbol{q},n+1} & \mathbf{0} \\ -k_p\boldsymbol{g}^{\mathcal{A}}_{\boldsymbol{q},n+1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{k_p^2}{p_p}\boldsymbol{I}^{\overline{\mathcal{A}}} \end{bmatrix} \tag{4.5}$$

$$\boldsymbol{K}^v_G = \begin{bmatrix} \boldsymbol{0} & -k_v\boldsymbol{g}^{\mathcal{B},T}_{\boldsymbol{q},n+1} & \mathbf{0} \\ -k_v\boldsymbol{g}^{\mathcal{B}}_{\boldsymbol{q},n+1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{k_v^2}{p_v}\boldsymbol{I}^{\overline{\mathcal{B}}} \end{bmatrix} \tag{4.6}$$

In a similar way, we consider that the frictional contact element does not contribute to the smooth motion and only participates to the position and velocity equations through vectors $\boldsymbol{F}^p_G(\boldsymbol{q}_{n+1}, \boldsymbol{\nu}_{n+1})$ and $\boldsymbol{F}^v_G(\boldsymbol{v}_{n+1}, \boldsymbol{\Lambda}_{n+1})$ that will be detailed below.

The proposed node-to-face frictional contact element is formulated in the framework of finite displacements and rotations. It is composed of a master planar rigid face in space with reference node A in body $\mathcal{B}^1$, and a slave node B belonging to body $\mathcal{B}^2$, see Fig. 4.1. For conciseness, this study focuses on the contact of a node with a planar external surface of a body.

The coordinates of the contact element $\boldsymbol{q}$ include the positions $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ of nodes A and B, respectively, and the rotation operator $\boldsymbol{R}_A$ of node A. For the treatment of the rotation operator, we follow a similar formulation and solution strategy as described in [78] and in Section 2.9. The rotational velocity of node A is thus represented in the velocity vector $\boldsymbol{v}$ by the three components of the angular velocity vector $\boldsymbol{\Omega}_A$ in the local frame of node A. Therefore

$$\dot{\boldsymbol{R}}_A = \boldsymbol{R}_A \boldsymbol{S}(\boldsymbol{\Omega}_A) \tag{4.7}$$

where the operator $\boldsymbol{S}(\boldsymbol{u}) : \mathbb{R}^3 \to \mathbb{R}^3 \otimes \mathbb{R}^3$ returns a 3×3 skew-symmetric matrix such that $\boldsymbol{u} \times \boldsymbol{v} = \boldsymbol{S}(\boldsymbol{u})\boldsymbol{v} = -\boldsymbol{v} \times \boldsymbol{u}$, $\forall\ \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^3$. Following the principle of the Lie group solver [78], the rotation at time step $n+1$ is represented as an increment vector $\boldsymbol{\Psi}_{A,n+1} \in \mathbb{R}^3$ with respect to the rotation at time step n as

$$\boldsymbol{R}_{A,n+1} = \boldsymbol{R}_{A,n} \exp(\boldsymbol{S}(\boldsymbol{\Psi}_{A,n+1})) \tag{4.8}$$

which involves the exponential map on the rotation group. The nonsmooth generalized-α time integration formulae are then expressed in terms of the unknowns $\boldsymbol{\Psi}_{A,n+1}$ and $\boldsymbol{\Omega}_{A,n+1}$ at position and velocity levels, respectively, as discussed in Section 2.9

The inertial frame is defined by a set of orthonormal vectors $\boldsymbol{E}_1$, $\boldsymbol{E}_2$ and $\boldsymbol{E}_3$, see Fig 4.1. During motion, node B undergoes a displacement, meanwhile the rigid plane rotates and translates. The normal to the contact surface in the reference configuration is given by $\boldsymbol{N}$, meanwhile the vectors $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are tangent to the contact surface. The positions of nodes A and B in the reference configuration are $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$, respectively, and the positions of nodes A and B in the current configuration are given by vectors $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$, respectively. During motion, the new orientations of $\boldsymbol{N}$, $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ at the current configuration, are denoted by vectors $\boldsymbol{n}$, $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$, respectively:

$$\boldsymbol{n}_{n+1} = \boldsymbol{R}_{A,n+1}\boldsymbol{N} \tag{4.9}$$

$$\boldsymbol{t}_{1,n+1} = \boldsymbol{R}_{A,n+1}\boldsymbol{T}_1 \tag{4.10}$$

$$\boldsymbol{t}_{2,n+1} = \boldsymbol{R}_{A,n+1}\boldsymbol{T}_2 \tag{4.11}$$

The normal gap between nodes A and B in the current configuration is

$$g_{N,n+1} = \boldsymbol{N}^T \boldsymbol{R}^T_{A,n+1} \left(\boldsymbol{x}_{B,n+1} - \boldsymbol{x}_{A,n+1}\right) \tag{4.12}$$

The normal gap is used to evaluate if the interacting bodies come in contact with each other (a zero gap indicates contact). From a geometrical point of view, the normal gap g_N represents the shortest distance from node A to the contact surface at the current configuration. Thus, $g_N > 0$ implies that the bodies are not in contact. In order to account for friction when the bodies are in contact, i.e if $g_N = 0$ in the time interval (t_n, t_{n+1}), the incremental tangential movement with components $g_{T1,n+1}, g_{T2,n+1}$ referred to the material frame $\boldsymbol{T}_1, \boldsymbol{T}_2$, is computed as follows

$$\begin{aligned} g_{T1,n+1} &= \boldsymbol{T}_1^T \left[\boldsymbol{R}_{A,n+1}^T \left(\boldsymbol{x}_{B,n+1} - \boldsymbol{x}_{A,n+1} \right) - \boldsymbol{R}_{A,n}^T \left(\boldsymbol{x}_{B,n} - \boldsymbol{x}_{A,n} \right) \right] \\ g_{T2,n+1} &= \boldsymbol{T}_2^T \left[\boldsymbol{R}_{A,n+1}^T \left(\boldsymbol{x}_{B,n+1} - \boldsymbol{x}_{A,n+1} \right) - \boldsymbol{R}_{A,n}^T \left(\boldsymbol{x}_{B,n} - \boldsymbol{x}_{A,n} \right) \right] \end{aligned} \tag{4.13}$$

This tangential movement defines if the contact state is stick or slip. Stick state not only means that the normal gap distance between the node and the face remains constant along two consecutive time steps, but also that there is no relative movement between the contact node B and the contact surface which is attached to node A. The surface rotation is taken into account projecting the relative position of the nodes in the material frame at time $n+1$ and n, and checking its variation over the time step according to Eq. (4.13). For better understanding imagine the case represented in Fig. 4.2 where the positions of nodes A and B are constant, but the contact surface attached to node A rotates around its normal direction. In this case, since the tangential movement is not zero and the contact state is slip even if the position of the nodes is constant.

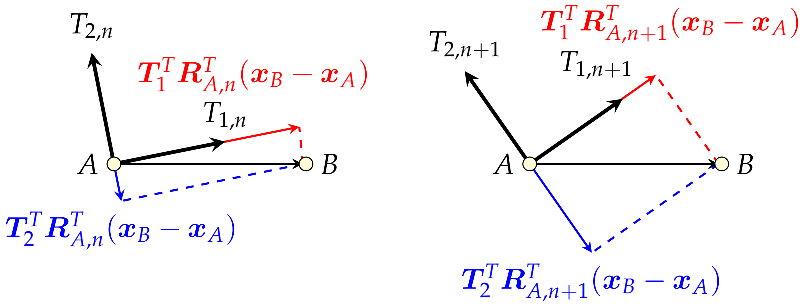

$T_{2,n}$
$\boldsymbol{T}_1^T \boldsymbol{R}_{A,n}^T(\boldsymbol{x}_B - \boldsymbol{x}_A)$
$T_{1,n}$
A
B
$\boldsymbol{T}_2^T \boldsymbol{R}_{A,n}^T(\boldsymbol{x}_B - \boldsymbol{x}_A)$
$\boldsymbol{T}_1^T \boldsymbol{R}_{A,n+1}^T(\boldsymbol{x}_B - \boldsymbol{x}_A)$
$T_{2,n+1}$
$T_{1,n+1}$
A
B
$\boldsymbol{T}_2^T \boldsymbol{R}_{A,n+1}^T(\boldsymbol{x}_B - \boldsymbol{x}_A)$

FIGURE 4.2: Tangential gap representation for a rotation contact surface

Finally, the generalized gap vector is defined as

$$\boldsymbol{g}_{n+1} = \left\{ \begin{array}{c} g_{N,n+1} \\ \boldsymbol{g}_{T,n+1} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{N}^T \boldsymbol{R}_{A,n+1}^T \left(\boldsymbol{x}_{B,n+1} - \boldsymbol{x}_{A,n+1} \right) \\ \boldsymbol{T}_1^T \left[\boldsymbol{R}_{A,n+1}^T \left(\boldsymbol{x}_{B,n+1} - \boldsymbol{x}_{A,n+1} \right) - \boldsymbol{R}_{A,n}^T \left(\boldsymbol{x}_{B,n} - \boldsymbol{x}_{A,n} \right) \right] \\ \boldsymbol{T}_2^T \left[\boldsymbol{R}_{A,n+1}^T \left(\boldsymbol{x}_{B,n+1} - \boldsymbol{x}_{A,n+1} \right) - \boldsymbol{R}_{A,n}^T \left(\boldsymbol{x}_{B,n} - \boldsymbol{x}_{A,n} \right) \right] \end{array} \right\} \tag{4.14}$$

where the first component is the normal gap, while the second and the third components are the tangential displacements. In what follows, we will adopt the subscripts T and N for denoting the components of a given quantity in the tangential and in the normal directions, respectively.

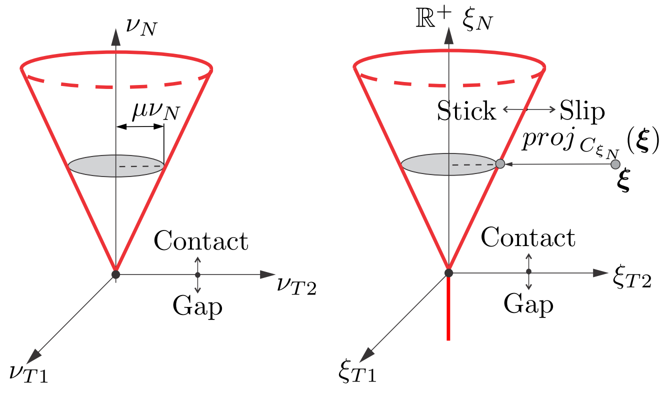
4.1.1 Frictional contact problem at position level

The modeling of friction represents an additional complexity with respect to the frictionless problem, as a consequence of the non-conservative character of the friction forces. A popular model is given by the Coulomb friction law. In this model, the bodies in contact can be in two different states: *stick* or *slip*. In stick, there is no relative displacement between the bodies, meanwhile in slip the bodies slide on each other. At position level, the friction law should be expressed in terms of the main unknowns $\boldsymbol{U}_{n+1}$ and $\boldsymbol{\nu}_{n+1}$ (with the splitting $\boldsymbol{q}_{n+1} = \widetilde{\boldsymbol{q}}_{n+1} + \boldsymbol{U}_{n+1}$). The restrictions of gap, contact, stick or slip, described before, are represented by the following frictional contact conditions,

$$
\begin{aligned}
& g_{N,n+1} \geq 0, \quad \nu_{N,n+1} \geq 0, \quad g_{N,n+1}\nu_{N,n+1} = 0; && \text{(4.15a)}\\
& \|\boldsymbol{g}_{T,n+1}\| \geq 0, \quad \|\boldsymbol{\nu}_{T,n+1}\| \leq \mu\nu_{N,n+1}, \\
& \qquad \|\boldsymbol{g}_{T,n+1}\| \left(\|\boldsymbol{\nu}_{T,n+1}\| - \mu\nu_{N,n+1}\right) = 0; && \text{(4.15b)}\\
& \boldsymbol{g}_{T,n+1} = -\|\boldsymbol{g}_{T,n+1}\| \frac{\boldsymbol{\nu}_{T,n+1}}{\|\boldsymbol{\nu}_{T,n+1}\|} && \text{(4.15c)}
\end{aligned}
$$

where ν_N and $\boldsymbol{\nu}_T$ are the normal and the tangential Lagrange multipliers at position level in the normal and tangential directions, respectively, and collinearity between the tangential displacement and the tangential contact force is assumed with the tangential force opposed to motion. The first condition in Eq. (4.15a) indicates the impenetrability restriction; the second one is the non-traction condition (only compression is allowed) and the third one is the complementarity equation. The second set of restrictions, Eq. (4.15b), represents the conditions for friction. The first inequality, that is always satisfied, states that the tangential displacement is positive or zero; the second one establishes that the maximum value of the density of the tangential contact force is $\mu\nu_{N,n+1}$, where μ is the friction coefficient; the third one is the complementarity equation, which indicates that $\|\boldsymbol{g}_{T,n+1}\|$ and $\|\boldsymbol{\nu}_{T,n+1}\| - \mu\nu_{N,n+1}$ cannot be simultaneously different from zero. Hence, when $\|\boldsymbol{\nu}_{T,n+1}\| < \mu\nu_{N,n+1}$ and $\|\boldsymbol{g}_{T,n+1}\| = 0$ the contact status is in stick, and when $\|\boldsymbol{g}_{T,n+1}\| \neq 0$, the body slips and the tangential force is equal to $\|\boldsymbol{\nu}_{T,n+1}\| = \mu\nu_{N,n+1}$.

It is important to remark that, in the proposed framework, it is mandatory to impose the Coulomb friction constraint at position level. In order to understand this, recall that the time integration scheme is characterized by three decoupled subproblems, and that the correction at position level is *blind* to any correction done at velocity level for the same time step where the terms are independent. Therefore, if the friction constraints are imposed only at velocity level some non-physical behaviour can be observed at position level, as shown in more detail in the application example in Section 4.2.1.



(a) Coulomb isotropic friction cone

(b) Extension of the augmented Coulomb isotropic friction cone

FIGURE 4.3: Three dimensional representation of the Coulomb friction cone.

The set of constraints given in Eq. (4.15) could have been expressed in the form of an inclusion as in Eq. (4.1). In this work, from a convex analysis as in Alart and Curnier [43], the frictional contact problem is solved by transforming the normal and tangential inclusions into equivalent equations by using the proximal point algorithm [82]. For the sake of conciseness, in what follows, we only work with the part of the augmented Lagrangian corresponding to the unilateral constraints of a specific contact element. A similar approach was used in [80]. The adopted form of the augmented Lagrangian approach can be derived from the following function expressed in terms of the variables $\boldsymbol{q}_{n+1}$ and $\boldsymbol{\nu}_{n+1}$ defined in the global frame of reference.

$$\begin{aligned}\mathcal{L}^p(\boldsymbol{q}_{n+1},\boldsymbol{\nu}_{n+1}) = & -k_p g_{N,n+1}\nu_{N,n+1} + \frac{p_p}{2}\left(g_{N,n+1}\right)^2 - \\ & \frac{1}{2p_p}\mathrm{dist}^2\left[k_p\nu_{N,n+1} - p_p g_{N,n+1}, \mathbb{R}^+\right] - k_p \boldsymbol{g}_{T,n+1}\cdot\boldsymbol{\nu}_{T,n+1} + \\ & \frac{p_p}{2}\|\boldsymbol{g}_{T,n+1}\|^2 - \frac{1}{2p_p}\mathrm{dist}^2\left[k_p\boldsymbol{\nu}_{T,n+1} - p_p\boldsymbol{g}_{T,n+1}, C_{\xi_N}\right]\end{aligned} \tag{4.16}$$

where it is understood that the gap function depends on $\boldsymbol{q}$, though this is not explicitly mentioned for the sake of conciseness.

In order to facilitate the presentation of the problem, the augmented Lagrangian is split in two terms: the normal and the tangential contributions

$$\mathcal{L}^p(\boldsymbol{q}_{n+1},\boldsymbol{\nu}_{n+1}) = \mathcal{L}^p_N(\boldsymbol{q}_{n+1},\nu_{N,n+1}) + \mathcal{L}^p_T(\boldsymbol{q}_{n+1},\boldsymbol{\nu}_{T,n+1}) \tag{4.17}$$

where the augmented Lagrangian in the normal direction is

$$\begin{aligned}\mathcal{L}_N^p(\boldsymbol{q}_{n+1}, \nu_{N,n+1}) = -k_p g_{N,n+1}\nu_{N,n+1} + \frac{p_p}{2}\left(g_{N,n+1}\right)^2 - \\ \frac{1}{2p_p}\mathrm{dist}^2\left[\xi_{N,n+1}, \mathbb{R}^+\right]\end{aligned} \tag{4.18}$$

while in the tangential direction it is given by

$$\begin{aligned}\mathcal{L}_T^p(\boldsymbol{q}_{n+1}, \boldsymbol{\nu}_{T,n+1}) = -k_p \boldsymbol{g}_{T,n+1} \cdot \boldsymbol{\nu}_{T,n+1} + \frac{p_p}{2}\|\boldsymbol{g}_{T,n+1}\|^2 - \\ \frac{1}{2p_p}\mathrm{dist}^2\left[\boldsymbol{\xi}_{T,n+1}, C_{\xi_N}\right]\end{aligned} \tag{4.19}$$

The function $\mathrm{dist}(\boldsymbol{z}, C)$ represents the distance between a point $\boldsymbol{z} \in \mathbb{R}^n$ and a convex set C, see [28]. The normal part of the Lagrangian is based on the definition of the distance to the set $\mathbb{R}^+$:

$$\mathrm{dist}(\xi_N, \mathbb{R}^+) = \begin{cases} -\xi_N & \text{if} \quad \xi_N < 0 \qquad\qquad \text{Gap} \\ 0 & \text{if} \quad \xi_N \geq 0 \qquad\qquad \text{Contact} \end{cases} \tag{4.20}$$

The Coulomb isotropic friction law, in 3D problems, is represented by a cone as shown in Fig. 4.3(a). A section of the Coulomb cone of radius $\mu\nu_N$ is defined by

$$C(\nu_N) = \{\boldsymbol{\nu}_T \text{ s.t. } ||\boldsymbol{\nu}_T|| \leq \mu\nu_N\} \tag{4.21}$$

which represents the set of admissible tangential friction forces. Then, the extended augmented cone C_{ξ_N} is the convex set defined by extension of the friction cone to the half line $\mathbb{R}^-(\xi_N)$, i.e., the set of negative values of the normal augmented multiplier, see Fig. 4.3(b). The tangential part of the Lagrangian is based on the definition of the distance to the extended augmented friction cone:

$$\mathrm{dist}(\boldsymbol{\xi}_T; C_{\xi_N}) = \left\{ \begin{array}{ccccccc} \boldsymbol{\xi}_T & \text{if} & \xi_N & < & 0 & \text{Gap} \\ \boldsymbol{\xi}_T - \mu\xi_N\boldsymbol{\tau}_p & \text{if} & \|\boldsymbol{\xi}_T\| & \geq & \mu\xi_N & \text{Slip} \\ \boldsymbol{0} & \text{if} & \|\boldsymbol{\xi}_T\| & < & \mu\xi_N & \text{Stick} \end{array} \right. \tag{4.22}$$

where $\boldsymbol{\tau}_p = \boldsymbol{\xi}_T / \|\boldsymbol{\xi}_T\|$ is a unit vector pointing in the direction of the tangential contact force.

Force vector in the normal direction

The generalized internal force vector in the normal direction is obtained by taking variations of Eq. (4.18),

$$\delta\mathcal{L}_N^p(\boldsymbol{q}, \nu_N) = -k_p\delta\nu_N g_N - \xi_N\delta g_N - \frac{1}{p_p}\delta\left[\frac{1}{2}\mathrm{dist}^2(\xi_N, \mathbb{R}^+)\right] \tag{4.23}$$

where $\xi_N = k_p \nu_N - p_p g_N$ is the augmented multiplier in the normal direction. In this equation, the subindex $n+1$ is omitted for conciseness. This notation is adopted for the rest of the chapter. The variation of $(1/2)\text{dist}^2(\xi_N, \mathbb{R}^+)$ in Eq. (4.23) gives

$$\delta\left[\frac{1}{2}\text{dist}^2(\xi_N, \mathbb{R}^+)\right] = \begin{cases} \xi_N \delta\xi_N & \text{if} \quad \xi_N < 0 \quad \text{Gap} \\ 0 & \text{if} \quad \xi_N \geq 0 \quad \text{Contact} \end{cases} \tag{4.24}$$

The virtual variation of the normal gap in Eq. (4.12), is written as follows

$$\delta g_N = \boldsymbol{n}^T(\delta\boldsymbol{x}_B - \delta\boldsymbol{x}_A) + \boldsymbol{N}^T\boldsymbol{S}(\boldsymbol{R}_A^T(\boldsymbol{x}_B - \boldsymbol{x}_A))\delta\boldsymbol{\Theta}_A \tag{4.25}$$

where the variation of the normal $\delta\boldsymbol{n} = -\boldsymbol{R}_A\tilde{\boldsymbol{N}}\delta\boldsymbol{\Theta}_A$ was used. Here, $\delta\boldsymbol{\Theta}_A$ is the variation of the incremental material rotations at node A such that $\delta\boldsymbol{R}_A = \boldsymbol{R}_A\boldsymbol{S}(\delta\boldsymbol{\Theta}_A)$, [5]. In order to obtain a compact expression for the normal gap variation, Eq. (4.25) can be written as follows

$$\delta g_N = g_{N\boldsymbol{q}}\delta\boldsymbol{q} \tag{4.26}$$

where the gradient matrix in the normal direction is

$$g_{N\boldsymbol{q}} = [-\boldsymbol{n}^T \quad \boldsymbol{N}^T\boldsymbol{S}(\boldsymbol{d}_{AB}) \quad \boldsymbol{n}^T] \tag{4.27}$$

with $\boldsymbol{d}_{AB} = \boldsymbol{R}_A^T(\boldsymbol{x}_B - \boldsymbol{x}_A)$; $\delta\boldsymbol{q}$ is the variation of the nodal coordinates for a contact element with nodes A, B, where the positions and rotations are ordered in the following way

$$\delta\boldsymbol{q} = [\delta\boldsymbol{x}_A^T \ \delta\boldsymbol{\Theta}_A^T \ \delta\boldsymbol{x}_B^T]^T \tag{4.28}$$

Replacing Eqs. (4.24, 4.26, 4.27) into Eq. (4.23), the generalized internal forces vector of the contact element $\boldsymbol{F}_N^p$ can then be readily identified as conjugated to the variation of generalized coordinates of the element $\boldsymbol{\Phi} = [\boldsymbol{q}^T \ \nu_N \ \boldsymbol{\nu}_T^T]^T$ as follows,

$$\delta\mathcal{L}_N^p(\boldsymbol{\Phi}) = \delta\boldsymbol{\Phi}^T\boldsymbol{F}_N^p(\boldsymbol{\Phi}) = \begin{Bmatrix} \delta\boldsymbol{q} \\ \delta\nu_N \\ \delta\boldsymbol{\nu}_T \end{Bmatrix}^T \begin{cases} \begin{Bmatrix} \boldsymbol{0} \\ -\frac{k_p^2}{p_p}\nu_N \\ \boldsymbol{0} \end{Bmatrix} & \xi_N < 0 \qquad \text{Gap} \\ \begin{Bmatrix} -g_{N\boldsymbol{q}}^T\xi_N \\ -k_p g_N \\ \boldsymbol{0} \end{Bmatrix} & \xi_N \geq 0 \qquad \text{Contact} \end{cases} \tag{4.29}$$

Force vector in the tangential direction

The generalized internal force vector in the tangential direction is computed in a similar way. By taking variations of Eq. (4.19) the following equation is obtained

$$\delta\mathcal{L}_T^p(\boldsymbol{q},\boldsymbol{\nu}_T) = -k_p\delta\boldsymbol{\nu}_T\cdot\boldsymbol{g}_T - \boldsymbol{\xi}_T\cdot\delta\boldsymbol{g}_T - \frac{1}{p_p}\delta\left[\frac{1}{2}\mathrm{dist}^2(\boldsymbol{\xi}_T,C_{\xi_N})\right] \tag{4.30}$$

The variation of $(1/2)\mathrm{dist}^2(\boldsymbol{\xi}_T;C_{\xi_N})$ is

$$\delta\left[\frac{1}{2}\mathrm{dist}^2(\boldsymbol{\xi}_T;C_{\xi_N})\right] = \begin{cases} \boldsymbol{\xi}_T\cdot\delta\boldsymbol{\xi}_T & \text{if} \quad \xi_N < 0 \quad \text{Gap} \\ (\|\boldsymbol{\xi}_T\| - \mu\xi_N)\boldsymbol{\tau}_p\cdot\delta\boldsymbol{\xi}_T & \text{if} \quad \|\boldsymbol{\xi}_T\| \geq \mu\xi_N \quad \text{Slip} \\ 0 & \text{if} \quad \|\boldsymbol{\xi}_T\| < \mu\xi_N \quad \text{Stick} \end{cases} \tag{4.31}$$

where $\boldsymbol{\tau}_p = \boldsymbol{\xi}_T/\|\boldsymbol{\xi}_T\|$ is the direction of the tangential contact force at position level. Then, the virtual variation of the tangential displacement $\boldsymbol{g}_T$ (Eq. (4.13)) is given by

$$\delta\boldsymbol{g}_T = \boldsymbol{g}_{T\boldsymbol{q}}\delta\boldsymbol{q} \tag{4.32}$$

with the gradient matrix in the tangential direction defined as

$$\boldsymbol{g}_{T\boldsymbol{q}} = \begin{bmatrix} -\boldsymbol{t}_1^T & \boldsymbol{T}_1^T\boldsymbol{S}(\boldsymbol{d}_{AB}) & \boldsymbol{t}_1^T \\ -\boldsymbol{t}_2^T & \boldsymbol{T}_2^T\boldsymbol{S}(\boldsymbol{d}_{AB}) & \boldsymbol{t}_2^T \end{bmatrix} \tag{4.33}$$

Finally, replacing Eqs. (4.31, 4.32) in Eq. (4.30), the generalized internal forces vector of the contact element $\boldsymbol{F}_T^p$ for the status of gap, slip or stick, is given by

$$\delta\mathcal{L}_T^p(\boldsymbol{\Phi}) = \delta\boldsymbol{\Phi}^T\boldsymbol{F}_T^p(\boldsymbol{\Phi}) = \begin{Bmatrix} \delta\boldsymbol{q} \\ \delta\nu_N \\ \delta\boldsymbol{\nu}_T \end{Bmatrix}^T \begin{cases} \begin{Bmatrix} \boldsymbol{0} \\ 0 \\ -\frac{k_p^2}{p_p}\boldsymbol{\nu}_T \end{Bmatrix} & \xi_N < 0 \quad \text{Gap} \\ \begin{Bmatrix} -\mu\xi_N\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\tau}_p \\ 0 \\ \frac{k_p}{p_p}(-k_p\boldsymbol{\nu}_T + \mu\xi_N\boldsymbol{\tau}_p) \end{Bmatrix} & \|\boldsymbol{\xi}_T\| \geq \mu\xi_N \quad \text{Slip} \\ \begin{Bmatrix} -\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\xi}_T \\ 0 \\ -k_p\boldsymbol{g}_T \end{Bmatrix} & \|\boldsymbol{\xi}_T\| < \mu\xi_N \quad \text{Stick} \end{cases} \tag{4.34}$$

Force vector and Hessian matrix at position level

The final expression of the internal contact force vector for the different contact status are obtained by adding the normal and tangential contributions of the internal forces vector from Eqs. (4.29, 4.34).

$$\boldsymbol{F}^p = \left\{ \begin{array}{ll} \left\{ \begin{array}{c} \mathbf{0} \\ -\frac{k_p^2}{p_p}\nu_N \\ -\frac{k_p^2}{p_p}\boldsymbol{\nu}_T \end{array} \right\} & \xi_N < 0 \quad \text{Gap} \\ \left\{ \begin{array}{c} -g_{N\boldsymbol{q}}^T\xi_N - \mu\xi_N\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\tau}_p \\ -k_p g_N \\ \frac{k_p}{p_p}(-k_p\boldsymbol{\nu}_T + \mu\xi_N\boldsymbol{\tau}_p) \end{array} \right\} & \|\boldsymbol{\xi}_T\| \geq \mu\xi_N \quad \text{Slip} \\ \left\{ \begin{array}{c} -g_{N\boldsymbol{q}}^T\xi_N - \boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\xi}_T \\ -k_p g_N \\ -k_p\boldsymbol{g}_T \end{array} \right\} & \|\boldsymbol{\xi}_T\| < \mu\xi_N \quad \text{Stick} \end{array} \right. \tag{4.35}$$

The linearization of the internal force vector gives the contact Hessian matrix. The contributions to the Hessian matrix for the different contact conditions are expressed as follows:

$$\Delta\boldsymbol{F}^p = \left\{ \begin{array}{ll} \left\{ \begin{array}{c} \boldsymbol{0} \\ -\frac{k_p^2}{p_p}\Delta\nu_N \\ -\frac{k_p^2}{p_p}\Delta\boldsymbol{\nu}_T \end{array} \right\} & \xi_N < 0 \quad \text{Gap} \\ \left\{ \begin{array}{c} -\Delta g_{N\boldsymbol{q}}^T\xi_N - g_{N\boldsymbol{q}}^T\Delta\xi_N - \mu\Delta\xi_N\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\tau}_p - \mu\xi_N\Delta\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\tau}_p - \\ \qquad \mu\xi_N\boldsymbol{g}_{T\boldsymbol{q}}^T\Delta\boldsymbol{\tau}_p \\ -k_p\Delta g_N \\ \frac{k_p}{p_p}(-\Delta\boldsymbol{\nu}_T + \mu\Delta\xi_N\boldsymbol{\tau}_p + \mu\xi_N\Delta\boldsymbol{\tau}_p) \end{array} \right\} & \|\boldsymbol{\xi}_T\| \geq \mu\xi_N \quad \text{Slip} \\ \left\{ \begin{array}{c} -\Delta g_{N\boldsymbol{q}}^T\xi_N - g_{N\boldsymbol{q}}^T\Delta\xi_N - \Delta\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\xi}_T - \boldsymbol{g}_{T\boldsymbol{q}}^T\Delta\boldsymbol{\xi}_T \\ -k_p\Delta g_N \\ -k_p\Delta\boldsymbol{g}_T \end{array} \right\} & \|\boldsymbol{\xi}_T\| < \mu\xi_N \quad \text{Stick} \end{array} \right. \tag{4.36}$$

The contact/friction Hessian matrices are computed by replacing in Eq. (4.36) the Eqs. (4.26, 4.32) together with the derivatives of the matrices of constraints of gap and tangential displacements, which are next given:

$$\Delta g_{N\boldsymbol{q}}^T = \frac{\partial g_{N\boldsymbol{q}}^T}{\partial\boldsymbol{q}}\Delta\boldsymbol{q} = \begin{bmatrix} \mathbf{0} & \boldsymbol{R}_A\boldsymbol{S}(\boldsymbol{N}) & \mathbf{0} \\ -\boldsymbol{S}(\boldsymbol{N})\boldsymbol{R}_A^T & \boldsymbol{S}(\boldsymbol{N})\boldsymbol{S}(\boldsymbol{d}_{AB}) & \boldsymbol{S}(\boldsymbol{N})\boldsymbol{R}_A^T \\ \mathbf{0} & -\boldsymbol{R}_A\boldsymbol{S}(\boldsymbol{N}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x}_A \\ \Delta\boldsymbol{\Theta}_A \\ \Delta\boldsymbol{x}_B \end{bmatrix} \tag{4.37}$$

$$\Delta\boldsymbol{g}_{T\boldsymbol{q}}^T\boldsymbol{\tau} = \frac{\partial\boldsymbol{g}_{T\boldsymbol{q}}^T}{\partial\boldsymbol{q}}\boldsymbol{\tau}\Delta\boldsymbol{q} = \begin{bmatrix} \mathbf{0} & \boldsymbol{R}_A\boldsymbol{S}(\boldsymbol{\tau}) & \mathbf{0} \\ -\boldsymbol{S}(\boldsymbol{\tau})\boldsymbol{R}_A^T & \boldsymbol{S}(\boldsymbol{\tau})\boldsymbol{S}(\boldsymbol{d}_{AB}) & \boldsymbol{S}(\boldsymbol{\tau})\boldsymbol{R}_A^T \\ \mathbf{0} & -\boldsymbol{R}_A\boldsymbol{S}(\boldsymbol{\tau}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x}_A \\ \Delta\boldsymbol{\Theta}_A \\ \Delta\boldsymbol{x}_B \end{bmatrix} \tag{4.38}$$

The derivative of $\boldsymbol{\tau}_p$ is given by

$$\Delta\boldsymbol{\tau}_p = (\boldsymbol{I} - \boldsymbol{\tau}_p \otimes \boldsymbol{\tau}_p)\frac{\boldsymbol{\xi}_T}{\|\boldsymbol{\xi}_T\|} \tag{4.39}$$

and finally, the rotation increment $\Delta\boldsymbol{\Theta}_A$ is such that

$$\Delta\boldsymbol{R}_A = \boldsymbol{R}_A\boldsymbol{S}(\Delta\boldsymbol{\Theta}_A) \tag{4.40}$$

4.1.2 Frictional contact problem at velocity level

The augmented Lagrangian which regularizes the frictional contact problem at velocity level is given by

$$\begin{aligned}\mathcal{L}^v(\boldsymbol{v}_{n+1}, \boldsymbol{\Lambda}_{n+1}) = -k_v\mathring{g}_{N,n+1}\Lambda_{N,n+1} + \frac{p_v}{2}\left(\mathring{g}_{N,n+1}\right)^2 - \\ \frac{1}{2p_v}\mathrm{dist}^2\left[k_v\Lambda_{N,n+1} - p_v\mathring{g}_{N,n+1}, \mathbb{R}^+\right] - k_v\mathring{\boldsymbol{g}}_{T,n+1}\cdot\boldsymbol{\Lambda}_{T,n+1} + \\ \frac{p_v}{2}\|\mathring{\boldsymbol{g}}_{T,n+1}\|^2 - \frac{1}{2p_v}\mathrm{dist}^2\left[k_v\boldsymbol{\Lambda}_{T,n+1} - p_v\mathring{\boldsymbol{g}}_{T,n+1}, C_{\sigma_N}\right]\end{aligned} \tag{4.41}$$

where $\mathring{g}_N$ and $\mathring{\boldsymbol{g}}_T$ are the Newton impact law in the normal and tangential direction, respectively, and defined as follows:

$$\mathring{g}^j_{N,n+1} = g^j_{N\boldsymbol{q},n+1}\boldsymbol{v}_{n+1} + e^j_N g^j_{N\boldsymbol{q},n}\boldsymbol{v}_n \tag{4.42}$$

$$\mathring{\boldsymbol{g}}^j_{T,n+1} = \boldsymbol{g}^j_{T\boldsymbol{q},n+1}\boldsymbol{v}_{n+1} + e^j_T\boldsymbol{g}^j_{T\boldsymbol{q},n}\boldsymbol{v}_n \tag{4.43}$$

and C_{σ_N} is a section of radius $\mu\sigma_N$ of the augmented Coulomb friction cone expressed in terms of variables at velocity level with the generalized velocity vector given by

$$\boldsymbol{v}_{n+1} = \left[\dot{\boldsymbol{x}}^T_{A,n+1}\ \boldsymbol{\Omega}^T_{A,n+1}\ \dot{\boldsymbol{x}}^T_{B,n+1}\right]^T \tag{4.44}$$

Additionally, the dependence of the impact law in terms of $\boldsymbol{v}$ is not explicitly specified in Eq. (4.41) for the sake of conciseness.

Then, by following a similar reasoning as that presented in Section 4.1.1, the internal force vectors at velocity level are given by

$$\boldsymbol{F}^{v} = \left\{ \begin{array}{ll} \left\{ \begin{array}{c} \boldsymbol{0} \\ -\frac{k_v^2}{p_v} \Lambda_N \\ -\frac{k_v^2}{p_v} \boldsymbol{\Lambda}_T \end{array} \right\} & \sigma_N < 0 \quad \text{Gap} \\ \left\{ \begin{array}{c} -g_{N\boldsymbol{q}}^T \sigma_N - \mu \sigma_N \boldsymbol{g}_{T\boldsymbol{q}}^T \boldsymbol{\tau}_v \\ -k_p \mathring{g}_N \\ \frac{k_p}{p_p}(-k_p \boldsymbol{\Lambda}_T + \mu \sigma_N \boldsymbol{\tau}_v) \end{array} \right\} & \|\boldsymbol{\sigma}_T\| \geq \mu \sigma_N \quad \text{Slip} \\ \left\{ \begin{array}{c} -g_{N\boldsymbol{q}}^T \sigma_N - \boldsymbol{g}_{T\boldsymbol{q}}^T \boldsymbol{\sigma}_T \\ -k_v \mathring{g}_N \\ -k_v \mathring{\boldsymbol{g}}_T \end{array} \right\} & \|\boldsymbol{\sigma}_T\| < \mu \sigma_N \quad \text{Stick} \end{array} \right. \tag{4.45}$$

The unit vector $\boldsymbol{\tau}_v$ pointing in the direction of the contact force is defined as $\boldsymbol{\tau}_v = \boldsymbol{\Lambda}_T / \|\boldsymbol{\Lambda}_T\|$.

Finally, the linearization of the internal force vector gives the Hessian matrix. Note that in Step 3 of the algorithm presented in Section 3.2.3, where the projection on velocity constraints are calculated, the variable $\boldsymbol{q}$ coming from the solution of the position sub-problem is fixed. Thus, the gradient matrices $g_{N\boldsymbol{q}}$ and $\boldsymbol{g}_{T\boldsymbol{q}}$ are constant at this instance and do not contribute in the calculation of the Hessian matrix. Therefore, the Hessian matrix for the different contact conditions is as follows:

$$\Delta \boldsymbol{F}^{v} = \left\{ \begin{array}{ll} \left\{ \begin{array}{c} \boldsymbol{0} \\ -\frac{k_v^2}{p_v} \Delta \Lambda_N \\ -\frac{k_v^2}{p_v} \Delta \boldsymbol{\Lambda}_T \end{array} \right\} & \sigma_N < 0 \quad \text{Gap} \\ \left\{ \begin{array}{c} -g_{N\boldsymbol{q}}^T \Delta \sigma_N - \mu \Delta \sigma_N \boldsymbol{g}_{T\boldsymbol{q}}^T \boldsymbol{\tau}_v - \mu \sigma_N \boldsymbol{g}_{T\boldsymbol{q}}^T \Delta \boldsymbol{\tau}_v \\ -k_v \Delta \mathring{g}_N \\ \frac{k_v}{p_v}(-\Delta \boldsymbol{\Lambda}_T + \mu \Delta \sigma_N \boldsymbol{\tau}_v + \mu \sigma_N \Delta \boldsymbol{\tau}_v) \end{array} \right\} & \|\boldsymbol{\sigma}_T\| \geq \mu \sigma_N \quad \text{Slip} \\ \left\{ \begin{array}{c} -g_{N\boldsymbol{q}}^T \Delta \sigma_N - \boldsymbol{g}_{T\boldsymbol{q}}^T \Delta \boldsymbol{\sigma}_T \\ -k_v \Delta \mathring{g}_N \\ -k_v \Delta \mathring{\boldsymbol{g}}_T \end{array} \right\} & \|\boldsymbol{\sigma}_T\| < \mu \sigma_N \quad \text{Stick} \end{array} \right. \tag{4.46}$$

The contact/friction Hessian matrices are computed similarly to the contact/friction Hessian matrices at the position level. The internal forces vectors and the Hessian matrices at the position and velocity levels contribute to the global tangent matrices and to the generalized internal forces vectors by a standard assembly procedure.

4.2 Numerical examples

Several numerical examples are studied to evaluate the robustness and accuracy of the proposed frictional contact model. Each of the examples has been chosen to highlight different characteristics that help to evaluate the performance of the proposed friction algorithm. In the first example, a point mass undergoes slipping and sticking motion on an inclined plane, showing the ability of the proposed method to handle a friction force different from zero at equilibrium state. The second example consists of a rocking rod impacting two supports, which allows us to assess the performance of the algorithm when the contact surface does not remain still. Then, in order to study problems with flexible components, the motion of a sliding chain of springs and masses and the oblique impact of a beam with a rigid wall are considered. The last example consists in the simulation of a pendulum impacting a rigid plane attached to flexible supports, showing the ability of the algorithm for handling flexible elements, a moving contacting surface and 3D trajectories which imply a 3D frictional behaviour.

The frictional contact algorithm proposed in this work has been implemented in the finite element code Oofelie [66] as will be described in Chapter 5. The convergence of the numerical solution obtained with the proposed methodology is analyzed in all examples. For that purpose, as in the previous chapter, the error for each value of the time increment h is evaluated using the L^1 norm:

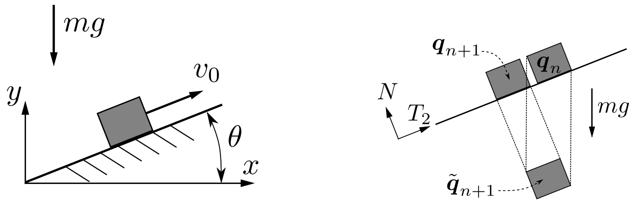
$$\mathrm{Error}(h) = \frac{\sum_{i=0}^{N} |f_i - f(t_i)|}{\sum_{i=0}^{N} |f(t_i)|} \tag{4.47}$$

where N is the number of time steps, f_i is the numerical solution obtained using the nonsmooth generalized-α method (NSGA) and $f(t_i)$ is the reference solution. The analytical solution is adopted as a reference solution in all cases in which it is available, otherwise the numerical solution corresponding to a very small time increment is taken as reference solution.

4.2.1 A point mass sliding and sticking on an inclined plane

This example intends to show the ability of the proposed algorithm to handle the switch between the sliding and sticking states, and to handle cases with a friction force different from zero at equilibrium state. The problem consists of a point mass which is initially in sliding state with an initial tangential velocity of $v_0 = 2\,\mathrm{m/s}$ with respect to the inclined plane, Fig. 4.4(a). Under the action of gravity and friction, the mass looses velocity. The final state depends on the friction coefficient and on the plane angle values. Here, the friction coefficient is larger than the tangent of the slope angle. Therefore, the mass reaches a state of sticking.

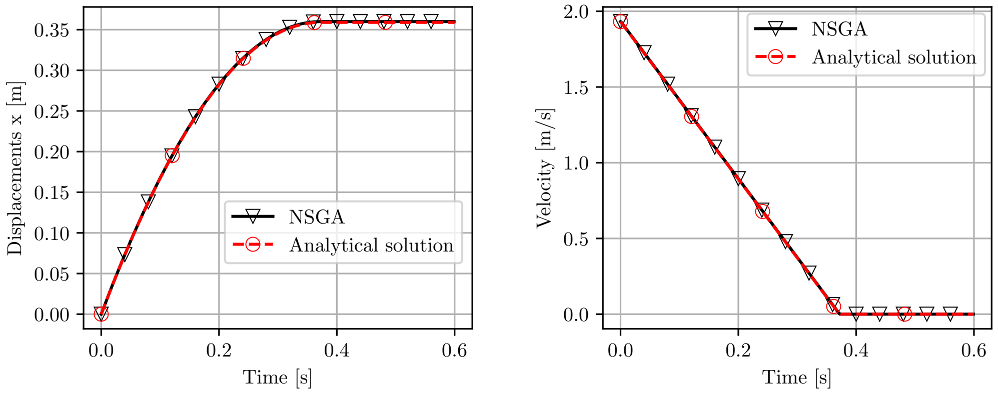
The parameters used for the simulation are defined as follows: the mass of the point is $m = 1$ kg, the angle of the inclined plane is $\pi/12$, the gravity acceleration is $g = 9.81\ \mathrm{m/s^2}$, the friction coefficient is $\mu = 0.3$ and the restitution coefficients for



(a) (b)

FIGURE 4.4: (a) A point mass sliding and sticking on an inclined plane. (b) Position correction when Coulomb's friction law is not imposed at position level.

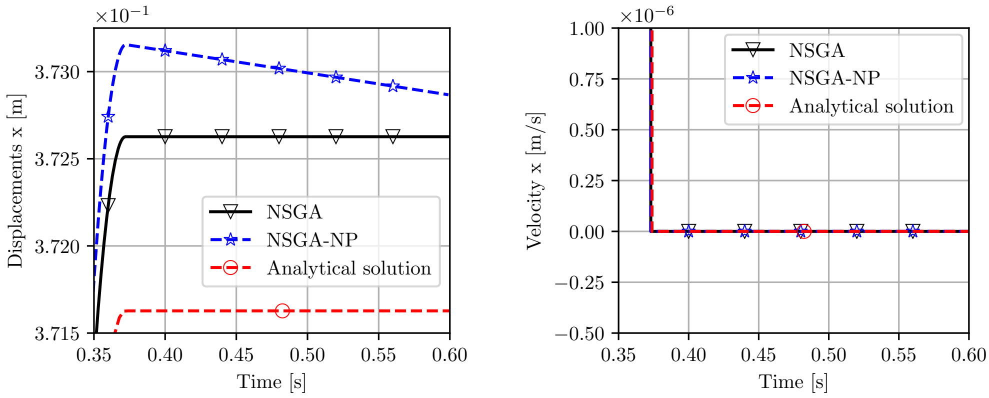
the normal and tangential directions are zero. The numerical solution is computed with a spectral radius of $\rho_\infty = 0.8$ and a total simulation time of 2s.



(a) Displacement in x direction.

(b) Velocity in x direction.

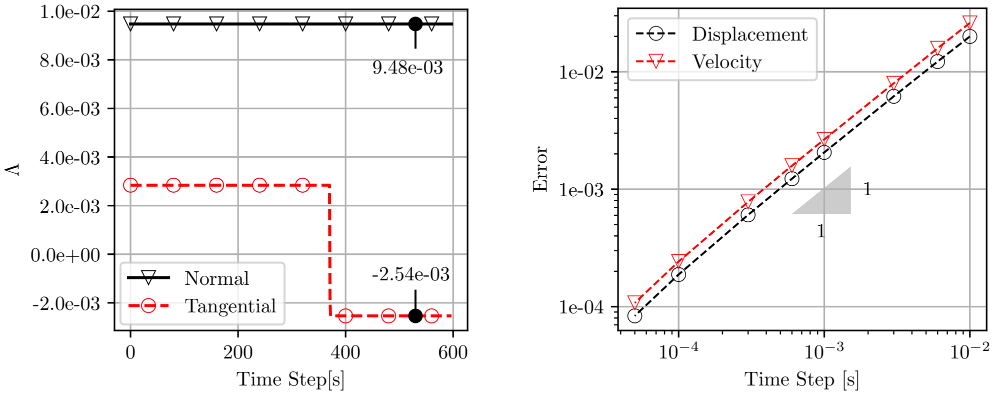
FIGURE 4.5: Point mass sliding and sticking on an inclined plane: results for a time step of 10^{-3} s.



(a) Displacement in x direction.

(b) Velocity in x direction.

FIGURE 4.6: Zoom of the Fig. 4.5, adding also the results for the case of NSGA-NP



(a) Lagrange multipliers.

(b) Convergence rate curves.

FIGURE 4.7: Point mass sliding and sticking on an inclined plane: results for a time step of 10^{-3} s.

A comparison between the numerical and analytical solutions for the displacement and the velocity fields is shown in Figs. 4.5(a,b), respectively, where a good agreement between them can be noticed. As expected, it can be observed that a slip motion takes place from 0 s to 0.374 s, followed by a constant sticking phase in which the block remains at rest.

As already mentioned in Section 4.1.1, the Coulomb's friction law must also be imposed at position level, due to the fact that both $\tilde{\boldsymbol{q}}$ and $\boldsymbol{U}$ are *blind* to any correction done at velocity level during the time step. In Fig. 4.6(a), the results obtained for the x component of the position and the velocity for the case in which the Coulomb's friction law is not imposed at position level (NSGA-NP) are shown.

As it can be observed in the NSGA-NP solution, when the contact is in stick mode, the position still evolves even though the velocity is zero. This non-physical phenomenon of the NSGA-NP algorithm is explained in Fig. 4.4(b), where it is made clear that the smooth prediction of the position $\tilde{\boldsymbol{q}}$ is only corrected in the normal direction, and, therefore, the position of the block ends up artificially sliding down in a kind of numerical *drift*. This problem is reduced when the time step of the NSGA-NP algorithm decreases and fully disappears when imposing in the NSGA algorithm as the Coulomb's law is then imposed at position level, see Figs. 4.5 and 4.6.

In Fig. 4.7(a) the Lagrange multipliers $\boldsymbol{\Lambda}$ in the normal and tangential directions can be observed. As expected from the physics, in the tangential direction a sign change takes place and it represents the change of the contact status, from sliding to sticking. Additionally, once the mass is at rest the numerical solution agrees with the analytical values for the normal reaction impulse Λ_N and the tangential friction impulse Λ_T which are in this case 9.48×10^{-3} Ns and -2.54×10^{-3} Ns, respectively. During the sliding phase the value of the tangential impulse is $\Lambda_T = 2.84 \times 10^{-3}$ Ns, which is equal to $\mu\Lambda_N$, satisfying the Coulomb's frictional law. It is remarked that in order to obtain the constraint force, the Lagrange multiplier $\boldsymbol{\Lambda}$ must be divided by the time step h, because $\boldsymbol{\Lambda}$ represents the impulse of the unilateral constraints which is the integral of the reaction force in the time interval $(t_n, t_{n+1}]$. Lastly, Fig. 4.7(b) shows the convergence obtained for the displacement and velocity fields, where, as expected, $\mathcal{O}(h)$ is achieved.

4.2.2 A rocking rod impacting two supports

Let us study a rigid rocking rod impacting two supports, see Fig. 4.8, which has already been studied by several authors [83, 84, 85]. In this example, the contact occurs between the support points and the face represented by the rod, meaning that the contact surface does not remain still. The parameters of the model and the initial conditions are taken from Zander *et al.* [83]. The rocking rod initially falls from a height $h_1 = 0.104$ m with an initial angle $\theta = 12°$ under the action of gravity $g = 9.8$ m/s^2. Then, it impacts against two rigid obstacles separated by a horizontal distance a of 0.4 m and a vertical height h_2, as depicted in Fig. 4.8. The rod is considered rigid and slender; it has a length $l = 1$ m, an inertia moment $I_z = 4 \times 10^{-2}$ kg m^2 and a mass $m = 0.48$ kg. Zander *et al.* [83] made a numerical study to find the restitution coefficients that best approximate the results obtained using a flexible model of the rod, and they found the values for the normal and tangential restitution coefficients to be $e_N = 0.6262$ and $e_T = 0$, respectively. In the following simulations, the time increment is 10^{-3} s, the spectral radius is $\rho_\infty = 0.8$ and the total simulation time is 0.3 s.

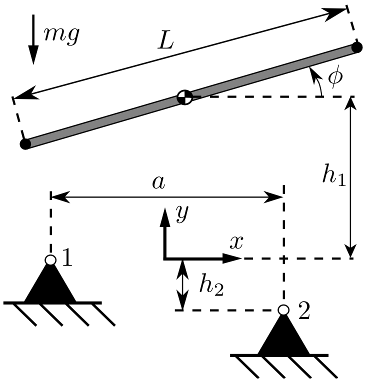
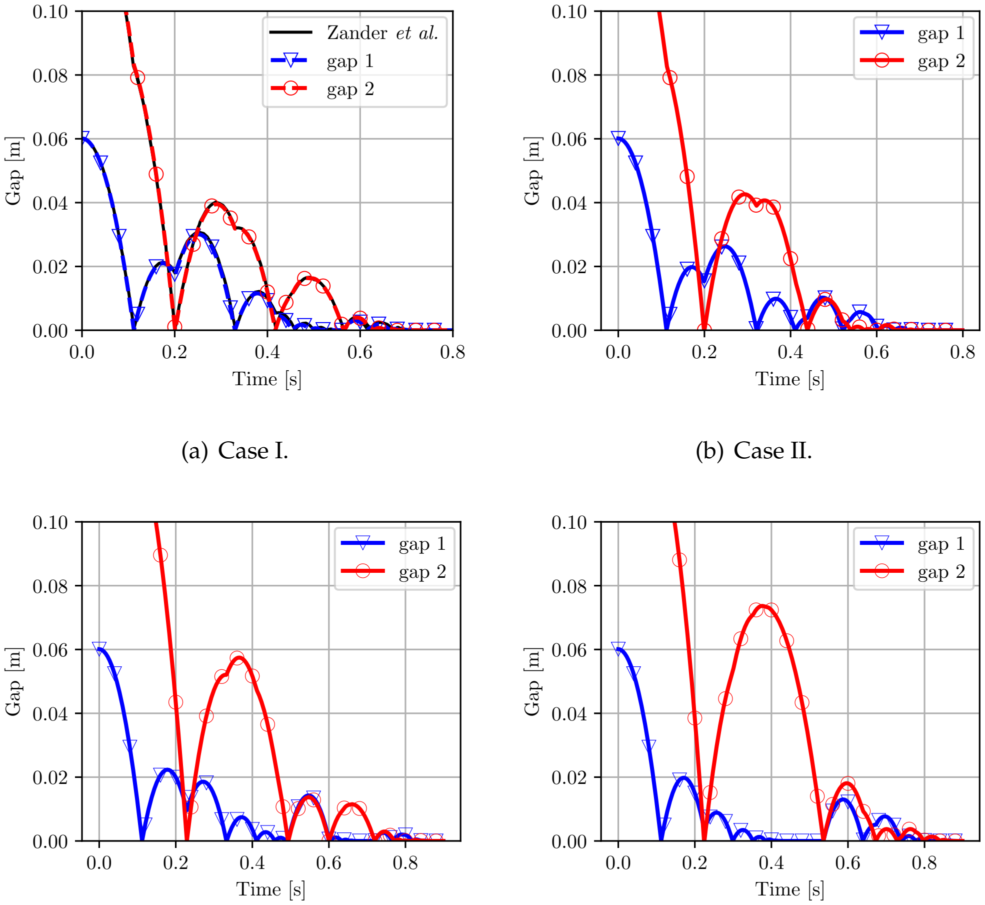


FIGURE 4.8: Rocking rod dimensions.

In order to evaluate the robustness of the proposed methodology, four different configurations for the supports are considered. The vertical position of the supports and the friction coefficients are selected as specified in Table 4.1, where h_2 is the vertical distance between supports and, μ_1 and μ_2 are the friction coefficients at support 1 and 2, respectively.

Case	h_2 [m]	μ_1	μ_2	a [m]
I	0	0.1	0.1	0.4
II	0	0	0.1	0.4
III	0.4	0.3	0.3	0.4
IV	0.4	0	0.3	0.4

TABLE 4.1: Rocking rod parameters.



(a) Case I.

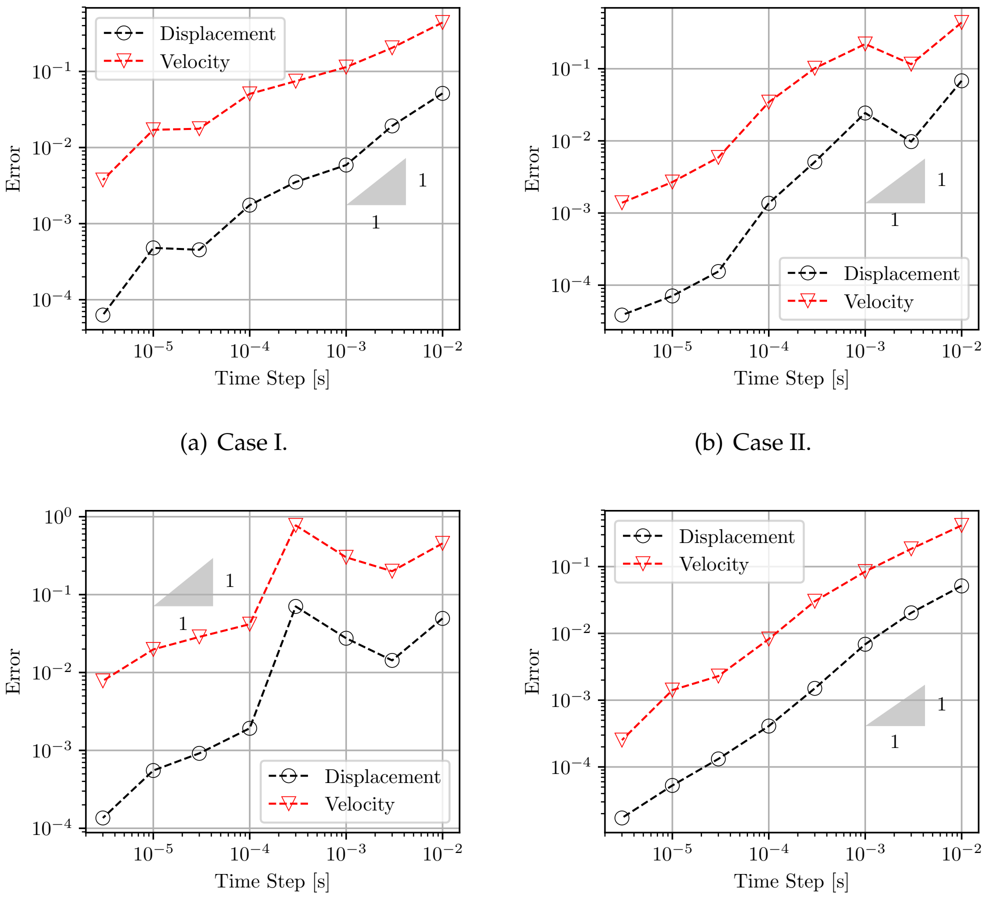
(b) Case II.

(c) Case III.

(d) Case IV.

FIGURE 4.9: Normal gap in the contacts for the rocking rod example.

The numerical solution of Case I is depicted in Fig. 4.9(a), where the normal gap distance between the supports and the rod are compared to the solution computed by Zander *et al.* [83]. Both solutions are in a good agreement. Figure 4.9(b) presents the results for Case II, where the normal gap distance at support 1 is assumed frictionless, whilst friction effects are considered in the support 2. By analyzing this figure, after the first impact, the normal gap distance at support 1 shows different bounces from the frictionless case (take Fig. 4.9(a) as reference). On the other hand, the solution for the normal gap distance at support 2 is not very affected until impacting for the second time. This happens because during this period of time the right end of the rod is in a free fall motion. Then, after the third impact, the solutions for Case I and II show a different behaviour.



(a) Case I.

(b) Case II.

(c) Case III.

(d) Case IV.

FIGURE 4.10: Convergence rate for the rocking rod example.

The cases III and IV, in which the supports are not aligned horizontally and the friction coefficients are different, are also tested. It can be observed that the solutions in Figs. 4.9(c,d) are quite different compared to Cases I and II since we are using a greater frictional coefficient. The differences between cases III and IV start to be slightly perceptible even after the first impact, and become obvious after the second impact. In all the studied cases, exact satisfaction of the constraints at position level is attained, see Fig. 4.9.

The case III, where the supports are not aligned and friction is present on both sides is a good example to explain the phenomenon of redundancies in multiple constraints, which has been studied by several authors [86, 87, 88]. This phenomenon often arises in simulations involving rigid bodies with frictional contact [89]. For a rigid body with several frictional contacts, like the rocking rod, if more than one contact is in stick state at the same time, dependent constraints are imposed. In this case, the solution is not unique.

In Fig. 4.11, the free body diagram of the rocking rod at the end of the simulation, once it is resting on the supports is presented. Let us remark that it is not perfectly centered: the COM rod is at the coordinate position (0.0134, -0.0213) m instead of (0, 0.02) m. It is clear that the normal reaction forces N_1 and N_2 are properly defined, and can be analytically computed resulting a value of $N_1 = 2.19$ N and $N_2 = 2.50$ N, which are in agreement with the values observed in Fig. 4.12. Nevertheless, since the rod is not moving, the contacts are in stick state, which implies that frictional forces F_1 and F_2 are equal or smaller than the maximum frictional force, $\|F_i\| \leq \mu \boldsymbol{N}_i$, for $i = 1, 2$. In this case there is an infinite number of possible solutions F_1, F_2 such that the condition $F_1 + F_2 - P_T = 0$ is satisfied. Figure 4.12 shows the Lagrangian multipliers divided by the time step, which are the reaction forces. In the tangential direction only one of the possible solution is obtained, which satisfies the condition $F_1 + F_2 - P_t = 0$ with the tangential weight $P_T = 0.47$ N.

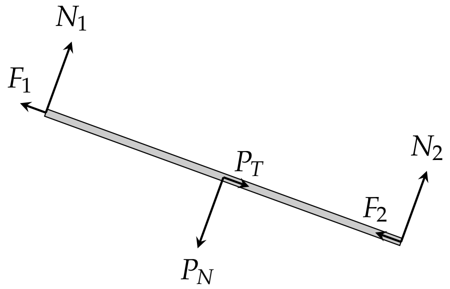
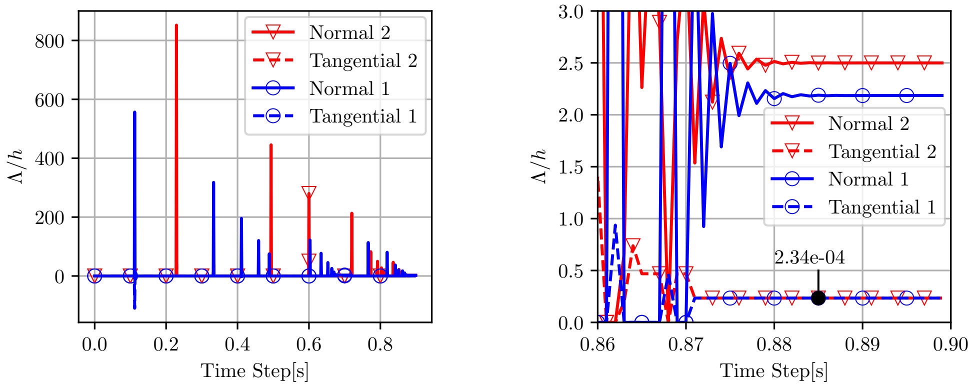


FIGURE 4.11: Free body diagram of the rod at rest



(a) Full time simulation

(b) Zoom

FIGURE 4.12: Case III: Lagrange multipliers at velocity level $\boldsymbol{\Lambda}$

The convergence of the error with the time step is shown in Fig. 4.10, the reference solution is obtained with $h = 10^{-6}$ s. A linear convergence rate is obtained for all cases.

4.2.3 Block sliding on a belt

This section deals with the dynamic modeling of a planar mechanism composed by a mass m attached to a spring with stiffness k and natural length l to a inertial fixed point, where $m = 3$ kg, $k = 50$ N/m and $l = 1$ m. As shown in Fig. 4.13, the mass starts from rest with an initial angle between the spring and the horizontal of $\pi/12$ radians. Due to the gravity effect it falls until it collides with a belt that has a constant velocity $v_o = 1$ m/s. The friction coefficient at the contact is $\mu = 1$, the restitution coefficients are $e_N = 0.5$ in the normal direction and $e_T = 0$ in the tangential direction. The numerical solution is computed with a spectral radius of $\rho_\infty = 0.8$ and the analyzed time interval is $t \in [0, 3]$ s. with a time step $h = 10^{-3}$ s.

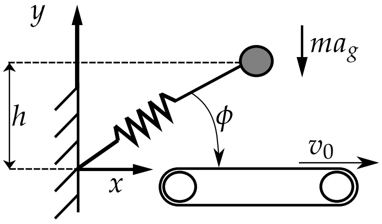
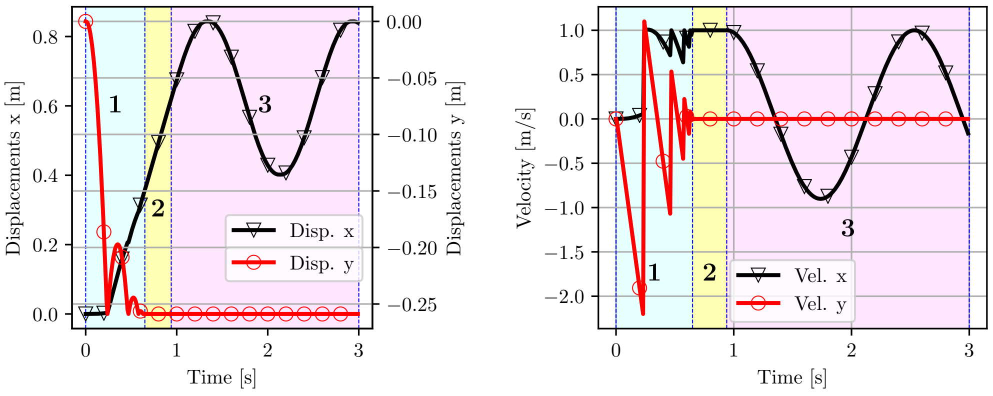


FIGURE 4.13: Block sliding in belt dimensions

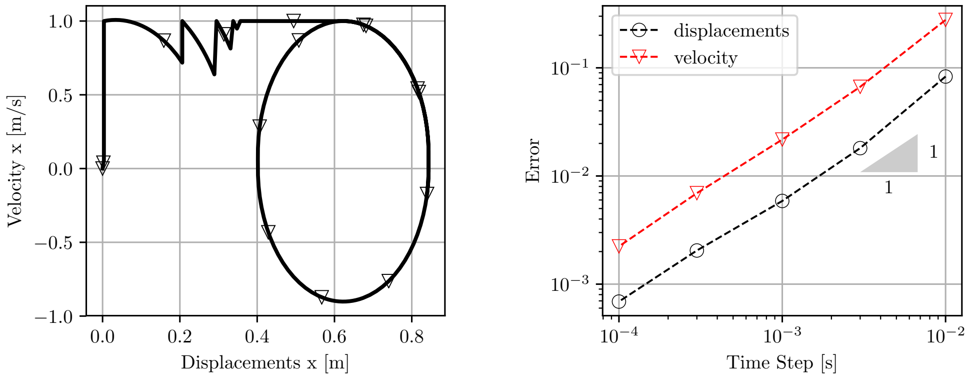
The mechanism has three different stages highlighted with colors in the charts presented in Fig. 4.14. The first one goes from the initial time until time $t = 0.65$ s. During this stage the mass is bouncing and dissipates energy in each bounce until it remains in contact with the belt. When the mass stops bouncing the second stage starts, in which the contact is in stick state. Therefore the mass moves with constant velocity along with the belt. During this movement the spring is elongated until the instant $t = 0.94$ s when the spring force overpasses the maximum frictional force and the mass slides over the belt, starting the third stage. These effect can be observed in Fig. 4.14 where the displacements and velocities of the mass are shown. In the second and third stage, displacements and velocity in the vertical direction become zero, while in the horizontal direction during the second stage the velocity is constant and in the third it becomes oscillatory. Fig. 4.15(a) shows the phase portrait for the x direction, where the the limit cycle of the third phase is shown. Finally Fig. 4.15(b) shows the convergence rate curves that, as in the previous examples, reveals a linear convergence. This example shows the capabilities of the method to account for all the possible contact states (gap, slip and stick) and the transition between them.



(a) Block displacements

(b) Block velocities

FIGURE 4.14: Block sliding in Belt



(a)

(b)

FIGURE 4.15: Block sliding in belt: (a) Phase portrait x, (b) Convergence rates curve

4.2.4 Sliding masses connected by springs

In this example, the capability of the proposed methodology to deal with flexible components is studied. It consists of 21 identical masses, $m = 0.0025$ kg, connected by 20 equal springs of length $\ell = 0.05$ m and rigidity $k = 20$ N/m, as shown in Fig. 4.16. The system has zero initial velocity and displacement, with an axial force $F = 0.4$ N acting on the first node from $t = 0$ s to $t = 0.1$ s. A friction coefficient $\mu = 0.3$ and a gravity acceleration of $g = 9.81$ m/s^2 are adopted. The normal and tangential restitution coefficients are zero, and the analyzed time interval is $t \in [0, 1.5]$ s. In the initial configuration, the length of each spring is its natural length.

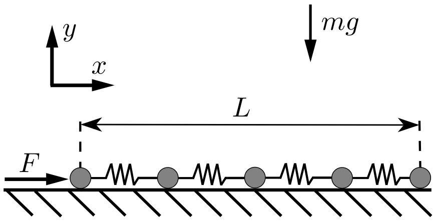
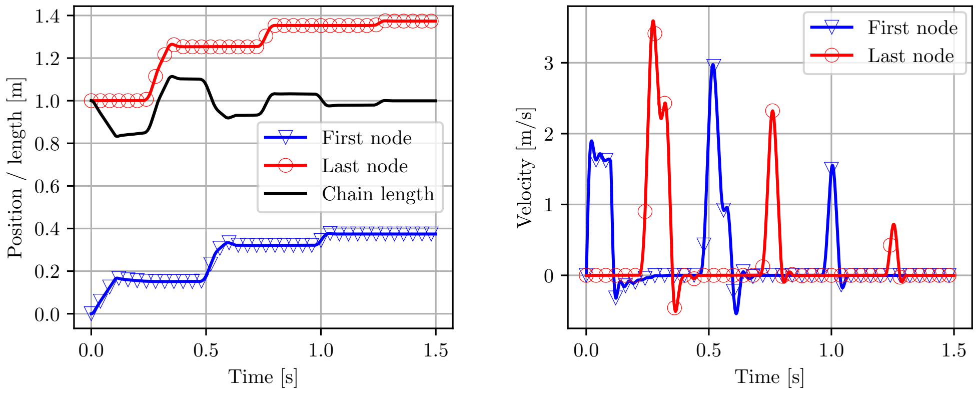


FIGURE 4.16: System of sliding masses connected by springs.

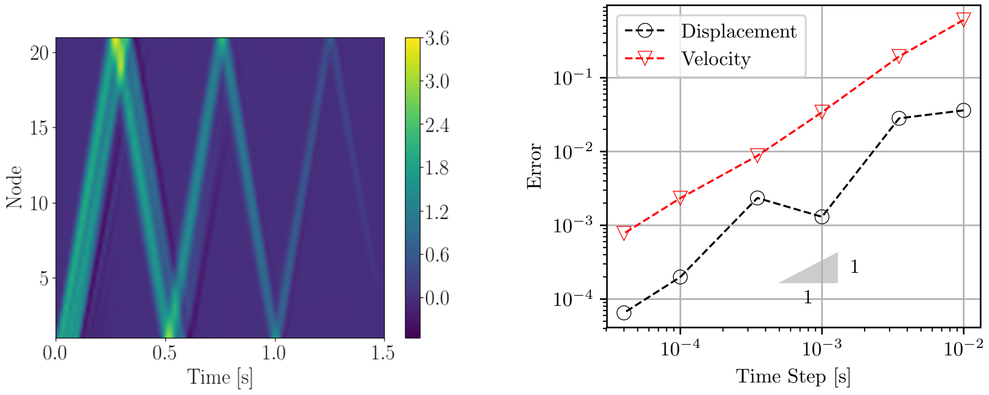
First, a time step of $h = 10^{-3}$ s is used. Fig. 4.17(a) shows the positions of the first and last nodes, and the total length of the system. The first node starts to move in response to the action of the applied force and a wave front propagates. The last node starts to move after 0.22 s, with the subsequent shrinkage of the system and a wave reflection. Thanks to friction, the wave propagates in several cycles but with a decreasing amplitude up to the point where the system remains still without moving and with its initial length recovered. The wave propagation effect can be clearly observed in Fig. 4.17(b) as well as in Fig. 4.18(a) where the velocity for each node is shown as a function of time. Finally, the results for a convergence study can be observed in Fig. 4.18(b) where, as expected, $\mathcal{O}(h)$ is achieved.



(a) Nodes position.

(b) Nodes velocity.

FIGURE 4.17: Sliding masses for a time step $h = 10^{-3}$s.



(a) Plot of nodal velocity amplitudes in time, illustrating wave propagation ($h = 10^{-3}$s).

(b) Convergence study.

FIGURE 4.18: Sliding masses results.

4.2.5 Oblique impact of an elastic beam against a rigid wall

This example deals with the oblique impact of an elastic beam against a rigid wall (Fig. 4.19), which allows us to evaluate the robustness of the method for problems involving friction and nonlinear flexible elements such as beams. Initially, García Orden and Goicolea [90] studied this problem using truss elements and only axial deformation effects. Then, Lens and Cardona [91] proposed a modification taking into account the flexural behaviour by using beam elements in the framework of an energy preserving time integration scheme.

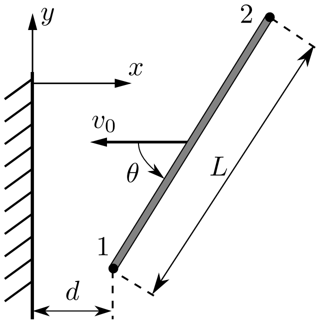
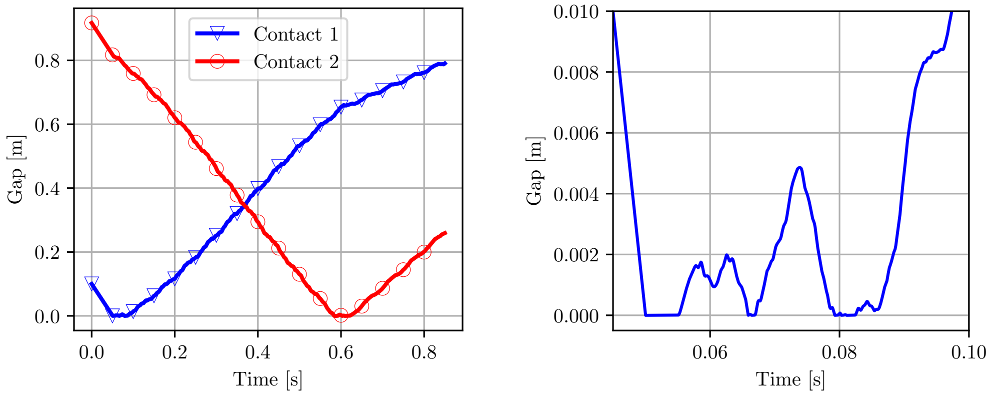


FIGURE 4.19: Oblique impact of a flexible beam with a rigid wall.

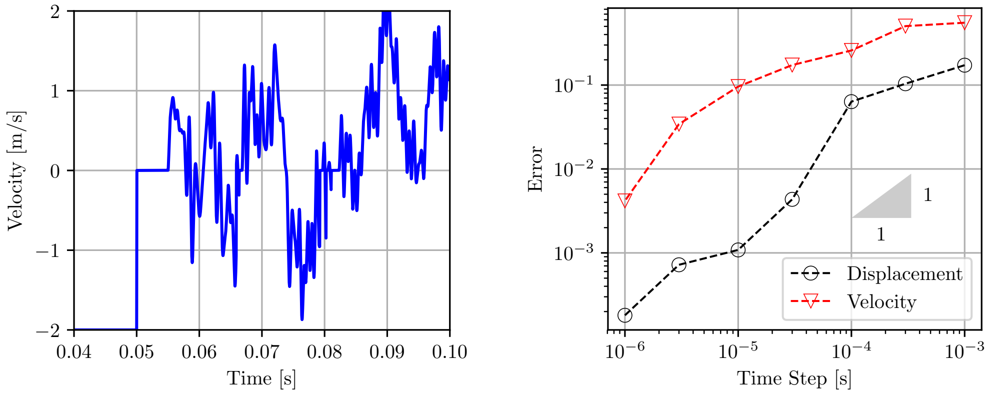
The elastic beam impacts against a vertical rigid wall with an angle of incidence θ. It moves horizontally until the lower tip of the beam impacts with the rigid wall. Then, it rotates resulting in a second impact at the upper end of the beam. The initial configuration is defined by $\theta = 35.2°$ and $v_0 = 2$ m/s. The beam is 1 m long with a transverse area of 2.5477×10^{-3} m^2 and inertia moment of $I = 5.40897 \times 10^{-7}$ m^4. The Young modulus is 1×10^9 N/m^2, the Poisson ratio is 0.3 and the mass density is $\rho = 7850$ kg/m^3. The friction coefficient is $\mu = 0.5$ and the normal and tangential restitution coefficients are zero. The beam is discretized with eight equally spaced nonlinear beam finite elements.



(a) Normal gap.

(b) Zoom of the normal gap for contact 1.

FIGURE 4.20: Oblique impact of a beam results.



(a) Horizontal velocity of the lower end of the beam.

(b) Convergence study.

FIGURE 4.21: Oblique impact of a beam results.

The normal gap for both contacts is shown in Fig. 4.20(a), where the results are computed with a time step $h = 10^{-4}$. It can be observed that the first impact takes place in the lower end of the beam at 0.05 s. The second impact occurs at the other end of the beam at 0.5878 s. A detail of the motion involving the dynamics of the first impact is shown in 4.20(b) and 4.21(a). It is observed that the impact phase has a duration of 82.5 ms. During the first 5 ms the contact remains closed, after which the tip of the beam clatters until the contact is completely released. This vibration effect is more evident at the velocity level, as it can be appreciated in Fig. 4.21(a), where the x-component of the velocity field during the impact phase is shown. The convergence analysis is shown in Fig. 4.21(b), it was computed for a total time of 0.85 s with different time step sizes.

The numerical solutions presented in this paper can be compared to those presented by Lens and Cardona [91] or García Orden and Goicolea [90]. However, we should take into account that García Orden and Goicolea only considered axial deformation effects and friction was not modeled. On the other hand, in the model of Lens and Cardona [91] the flexural behaviour was modeled using beam elements; however, the impact was frictionless. Furthermore, in both cases, the impact equations were regularized by the penalty method, and a high dependency of results on the penalty factor was shown. Hence, owing to these facts, the dynamic response obtained with the current model is not exactly the same to the ones found in [90, 91].

4.2.6 A pendulum impacting a rigid plane attached to flexible supports

As shown in Fig. 4.22 a pendulum impacting a rigid plane attached to flexible supports has been simulated. The plane is of square shape with a side length of $l = 5$ m. Its center of mass, located at its geometrical center, has its translations fixed, therefore it can only rotate. Each of the four corners of this plane is attached to a geometrically nonlinear spring-damper element, where the stiffness constant and the damping coefficient are 100 N/m and 1 Ns/m, respectively, and the initial length is $L_0 = 0.2$ m. The mass and the principal moments of inertia of the plate are equal to $m_s = 0.2$ kg and $I_s = m_s L^2/6$. At a height $h_0 = 1$ m from the center of the plane, there is attached a pendulum of 1.3 m length with mass and principal moments of inertia equal to $m_p = 5$ kg and $I_p = m_p l^2$. The fixed end of the pendulum is free to rotate, and initially the free end is in the position $(0; 1.3; 1)$ m. Both, the pendulum and the plane are under the action of the gravity equal to 9.81 m/s^2. Initially, the pendulum has an angular velocity of $\boldsymbol{\Omega} = (0; 0; 1.5)$ rad/s and the free end a velocity of $\boldsymbol{v} = (-1.95; 0; 0)$ m/s. Friction is modeled adopting a friction coefficient equal to $\mu = 0.3$, and the normal and tangential restitution coefficients are taken equal to zero. This setting allows us to show the ability of the proposed algorithm for handling flexible elements and a 3D frictional behaviour.

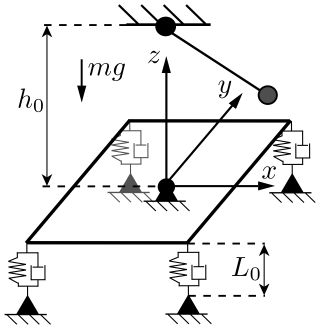
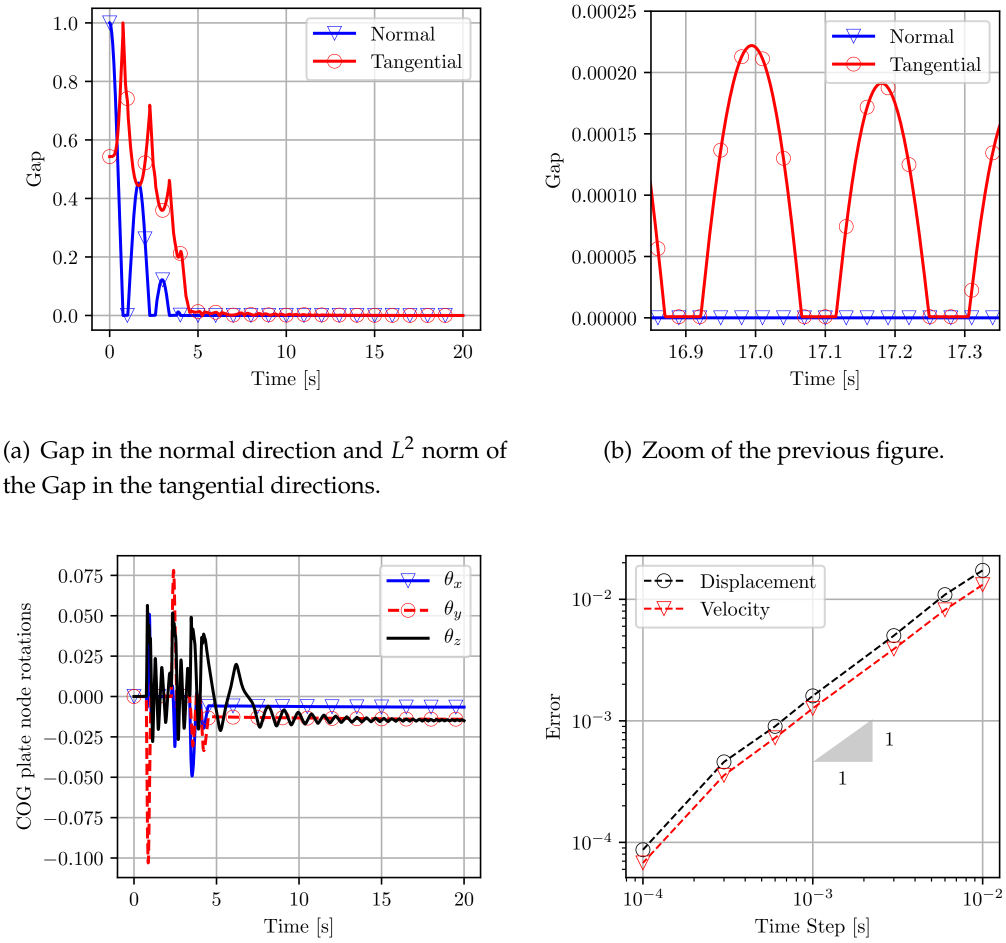


FIGURE 4.22: Pendulum impacting a rigid plane attached to flexible supports: Configuration of the problem.



(a) Gap in the normal direction and L^2 norm of the Gap in the tangential directions.

(b) Zoom of the previous figure.

(c) Rotation vector of the plane.

(d) Convergence study.

FIGURE 4.23: Pendulum impacting a rigid plane attached to flexible supports (results for $h = 10^{-3}$).

The pendulum starts a downwards trajectory in response to the effect of gravity, until colliding with the plane attached to flexible supports. Due to the flexibility of the system, the pendulum bounces adopting states of sliding and sticking as the orientation of the plane changes in response to the collisions with the pendulum. This can be observed in Fig. 4.23(a-b), where the bounces of the pendulum can be reckoned by analyzing the normal component of the gap, and the sticking and sliding phases can be inferred from the graph of the norm of the gap in the tangential direction. In Fig. 4.23(c) the history of the rotation vector of the plane can be appreciated. The results for a convergence study are shown in Fig. 4.23(d) where, as expected, $\mathcal{O}(h)$ is achieved.

4.3 Summary and concluding remarks

A new frictional contact algorithm for nonsmooth multibody systems is presented. The integration of the equations of motion of the frictional contact problem is performed using the nonsmooth generalized-α time integration scheme based on an augmented Lagrangian approach. The smooth contributions are integrated using a second-order scheme, whereas a first order scheme is used for the impulsive contributions. Compared with the classical Moreau-Jean method, the proposed method leads to qualitatively better numerical solutions with less numerical dissipation. The proposed contact model satisfies exactly the contact constraints at position level, which means that no penetration is observed. Impacts are also properly solved with the exact satisfaction of constraints at velocity level. Then, the complementarity problem for the normal and frictional contact problem is solved at each time step using a Newton semi-smooth method in a fully implicit approach with fast convergence. The proposed methodology has been successfully applied to study different mechanism configurations consisting of flexible and/or rigid bodies and to observe the friction effects on the dynamic response of the system.

The presented numerical examples demonstrate the ability of the method to represent frictional contact problems with large displacements and rotations in a two and three dimensions. The algorithm has three main features: i) the final solution is independent of the penalty parameters both in the tangential and normal directions, their values mostly affect the convergence rate, ii) the algorithm does not require modifications in existing finite elements of the library, in other words it is minimally intrusive and iii) the smooth motion of a mechanism which captures a large part of elastic and vibration phenomena in the flexible bodies, is solved using a generalized–α method with second-order accuracy and with controlled dissipation, something that is not possible with the first order integrators which are usually used to solve this kind of problems and introduce a much higher numerical dissipation. The equations for the analytical computation of the residual forces and tangent matrices of the frictional contact algorithm are provided. Five numerical examples of mechanisms with unilateral friction constraints are presented. Finally, the results of these examples were compared to analytical equations and previous numerical solutions obtained by other authors showing good agreement and convergence rate.

Chapter 5

Finite Element implementation

This chapter introduces the procedure to embed the nonsmooth generalized-α solver and the previously presented frictional contact formulation into a general purpose finite element code, in this case Oofelie::Multiphysics (Object Oriented Finite Elements Led by Interactive Executor) [66] software. Compared to other general purpose codes for nonsmooth dynamic systems, such as, LMGC90 [30], Siconos [63] and Chrono [64], the nonlinear finite element approach is particularly well-suited to deal with complex articulated systems including flexible components.

The chapter starts with an introduction to the Oofelie software, explaining the different types of classes, the scripting procedure and the data base used to store information. The description of the implementation for the new solver and the frictional contact presented in Chapters 3 and 4 is given. Finally the woodpecker toy example is simulated, which is a multibody dynamic system involving frictional contacts and nonlinear dynamic interactions. This example demonstrates the possibility to solve this type of problems in a systematic way using the finite element method.

5.1 Oofelie Multiphysics

Oofelie is a multiphysics finite element software designed to study coupled thermal, mechanical, acoustic and electrical fields problems. It is written in C++ making use of the object oriented philosophy. The structure of the software was firstly presented by Cardona et al. [66]. The first version of the software was developed in the *Laboratoire de Techniques Aéronautiques et Spatiales* (LTAS) at the University of Liège and at CIMEC-INTEC in Argentina. Later on in 2001 the spin-off Open Engineering was created, in order to further develop and industrialize the software. The use of Oofelie can be done through an associated interpreter, which uses a specialized high-level language with a syntax close to C++. This interpreter allows to access the most relevant classes and methods in the Oofelie code. It is used to define the necessary physical data of the problem, and set up the desired numerical options to solve it. Most classes in Oofelie can be classified in the next groups:

- **MTK** classes contain mathematical tools, such as different types of matrices and vectors, most of them inheriting from the virtual class `MotherMatrix` or

MotherVector. The idea is to optimize the matrix operations for each case, e.g. the Vect3 and Matr3 are defined to deal with 3D space operations.

- **Analysis** classes contain all the necessary classes to manage the data base (DB) where the results are stored.
- Each of the **FEM** classes represents a different type of mechanical finite element, e.g. linear flexible beams, nonlinear flexible beams, rigid bodies, hinges and so on.
- **Materials** classes are used to model different material behaviors.
- **Solvers** classes include different numerical algorithms techniques to solve distinct types of problems.
- In **Kernel** and **Main** are the main general classes of Oofelie used to define the work flow of the program and its data structure, for example, the Dof, Dofset, Partition, Connection, VectorStr, MatStr and Domain.

Any class which needs to be accessed from the interpreter needs an extra class, which has the same name preceded by oeI_. These classes define the wrapper between the interpreter and Oofelie classes.

An important contribution of this thesis to Oofelie is the implementation of the nonsmooth NSGA method and the frictional contact element presented in Chapters 3 and 4 respectively. The solver class is called oeNonSmoothGASolver2 and the contact element class oeContactNodeToFaceWithFrictionAndRotation.

5.1.1 Input data: Interpreter scripts

As aforementioned, the data used to carry out a simulation in Oofelie is introduced through the interpreter. This information is typed in a text file with the Oofelie extension, which is .e. In an Oofelie simulation file, the physical data to describe the system is stored inside an object of the class Domain. In order to create an object Domain with the necessary data for a simulation, at least one instance of the class PositionSet, ElementSet and FixationSet is required. Even though it is not mandatory, the ExcitationSet is also needed when external loads are applied. Finally to solve the problem, an instance of a solver class is needed. In our case the solver class is the NonSmoothGASolver2. The creation of a solver instance requires an associated domain. Finally some commands are required to save the data results, e.g. the class ToParaView can be used to save the data in the format of the post processing software Paraview, or the class ToUNV can be used to visualize the results in Siemens NX, provided that a dedicated plug-in is available. For a better understanding an example of the script for a falling ball is shown in Code 5.1. In line 8, the four arguments are the node number followed by the x, y and z components of its initial position.

CODE 5.1: Oofelie scritp example: Falling ball

```
Domain domain_1("fallingBall");
domain_1.setAnalysis(DYNAMIC_PO);

scalar mass = 1;
Propelem point_element(PointMass_E) { MASS, mass; }

PositionSet position_set_1();
position_set_1.define(1, 0.0, 1, 0.0);
domain_1.add(position_set_1);

ElementSet element_set_1();
element_set_1.define(1,point_element, 1);
domain_1.add(element_set_1);

FixationSet fixation_set_1();
domain_1.add(fixation_set_1);

ExcitationSet excitation_set_1();
excitation_set_1.define(1, TY|GF, (-9.81 * mass));

domain_1.dom_add(excitation_set_1);
domain_1.setAnalysis(DYNAMIC_PO);
domain_1.setStep(1);

NonSmoothGASolver2 solver (domain_1);
solver.setFinalTime (4);
solver.setTimeStep (2e-3);
solver.compute();

ToParaView toPV(domain_1);
toPV.exportCodeNode(DISPLACEMENT, "Displacement");
toPV.writeResults(DYNAMIC_PO, domain_1.getUserName());
```

5.1.2 Data structure and assembly procedure

Continuum fields are discretized in space using nodal coordinates [92], it means that different nodes in the space are connected by elements. Depending on which element connects the nodes with its neighborhood nodes, different DOF may be defined for each node, e.g. displacements, rotations or temperatures. When the elements are initialized it is checked if the necessary DOFs at their nodes already exist in the DB, if it is the case, then the existing DOFs are associated to the node element, therefore the contribution of this element is added to these associated DOFs. If the DOFs do not exist, they are created and associated. Also some DOFs can be associated to an element instead of a node, e.g. the Lagrange multipliers. Each of these DOFs are one instance of the class Dof and the list of DOFs is managed by the class DofSet.

The DOFs status can be saved for all the time steps and at different differentiation levels, such as the generalized displacements, generalized velocities and generalized accelerations. This information is stored in the DB that belongs to the domain, and it can be globally accessed from the solver but also locally from each element, which simplifies the exchange of information between the solver and the elements.

An intuitive way to imagine the Oofelie DB is as a set of tables storing information with the same structure. In these tables each row can be understood as the information related to a specific DOF and each column as the information for all the DOFs of one type or differentiation level, i.e. displacement, velocity, force or any oder relevant information depending on the analysis type. This concept is illustrated in Fig. 5.1, for a case where the displacements, velocities, accelerations and forces are stored. Each of these tables is related to a different instant for a given category and these categories in Oofelie are called Analysis, for example DYNAMIC_PO or NONLINEARSTATIC_PO. An Analysis is a compilation of data for different instants, each instant is called a step. At each step a table with all the data is stored, similarly to the one showed in 5.1. Depending on the Analysis type the meaning of these steps can be different. For example in the DYNAMIC_PO are the time discretization steps for a dynamic analysis, on the other hand in the NONLINEARSTATIC_PO Analysis are the incremental load steps. The Analysis type is chosen through the domain as can be seen in Code 5.2 and 5.3. In Fig. 5.2 an example of DB is represented, where Code 5.2 points to the table highlighted in red and Code 5.3 to the one highlighted in blue in Fig. 5.2.

CODE 5.2: Code used to set the domain analysis to dynamic

```
        dom->setAnalysis(DYNAMIC_PO);
        dom->setStep(1);
```

CODE 5.3: Code used to set the domain analysis to nonlinear static

```
        dom->setAnalysis(NONLINEARSTATIC_PO);
        dom->setStep(2);
```

In order to access the DB the class VectorStr can be used. This class not only allows one to access the DB but also to execute the assembly procedure. An instance of the class VectorStr is a vector where the size can be the number of DOF of the system or less, in which every element vector is a pointer to the corresponding DOF in the database. For a better understanding of this concept an example is shown in

Fig. 5.3. The example consists of a system with one node and one instance of the `VectorStr` which is related to the rotations of the node. In this case the DOF are the displacements, the rotations and the Lagrange multipliers.

In Oofelie, the contribution of an element is not computed inside one unique function, normally this contribution is splitted in different functions related to its nature. For example a spring-damper element could have different functions to compute the residue, one to compute the internal forces and other ones to compute the dissipative forces, called `fill_inter_forc` and `fill_dissipation_forc` respectively. The same philosophy applies for the tangent matrix: one function computes the damping contribution and another the stiffness contribution, called `fill_tang_damping` and `fill_tang_stiffness` respectively. These functions can be called from the `VectorStr` and `MatrixStr` classes, computing and assembling the contribution of all elements.

Additionally, another way to access the DB is by making direct requests to get the value of a DOF or several values for one node, for example to get the displacements in a three dimension space of one specific node. This type of calls is the one used inside the elements.

All these possibilities to access the DB give flexibility to implement different types of solvers, being useful, for example, to implement an implicit solver, for instance the NSGA, with almost no modification to the existing elements.

Finally, it is worth mentioning that even though the DB is organized in an efficient way, in some problems the number of nodes can be very large, and storing the data for all the time steps can be overwhelming in the sense that the computer can run out of memory. Therefore in Oofelie a memory manager tool allows one to save the information into the hard drive periodically every certain number of time steps in order to release memory.

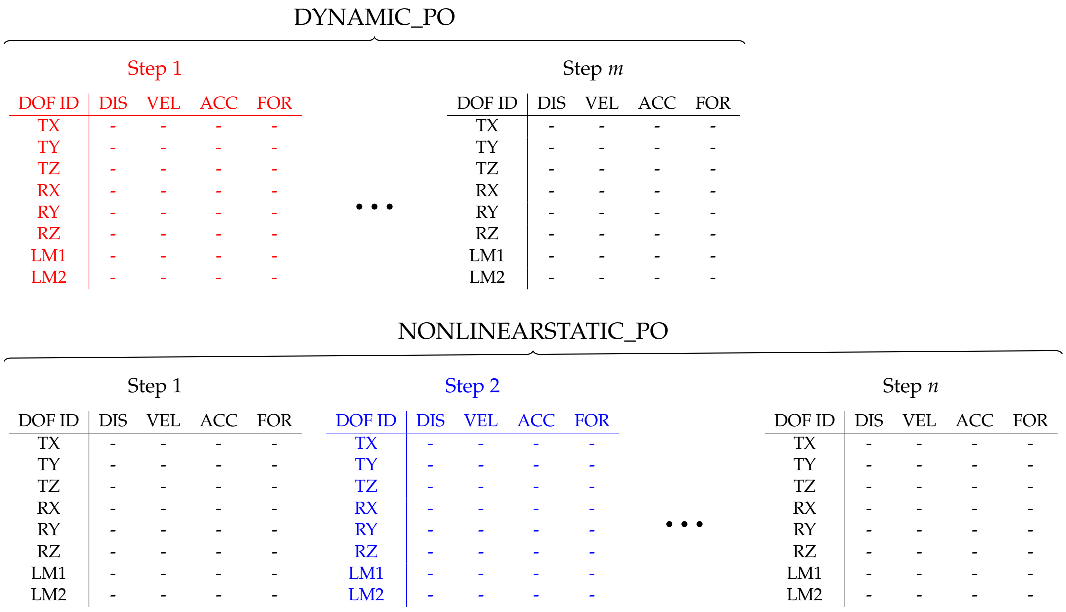
DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

FIGURE 5.1: Data base table representation

5.2 NSGA solver and frictional contact implementation

5.2.1 Solver classes

As introduced in the Chapter 3 the NSGA is based on three different problems to be solved at each time step: for the smooth variables, the position correction and the



DYNAMIC_PO

Step 1

DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

. . .

Step m

DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

NONLINEARSTATIC_PO

Step 1

DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

Step 2

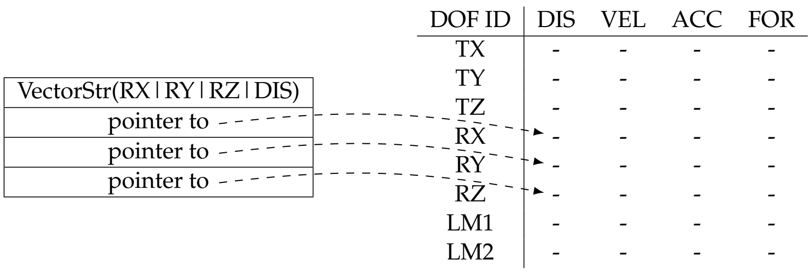
DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

. . .

Step n

DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

FIGURE 5.2: Data base representation



VectorStr(RX | RY | RZ | DIS)
pointer to
pointer to
pointer to

DOF ID	DIS	VEL	ACC	FOR
TX	-	-	-	-
TY	-	-	-	-
TZ	-	-	-	-
RX	-	-	-	-
RY	-	-	-	-
RZ	-	-	-	-
LM1	-	-	-	-
LM2	-	-	-	-

FIGURE 5.3: Data base representation

velocity jump. All of them have a similar structure. Therefore the solver has been implemented using different classes to make it flexible, efficient, and maintainable. The classes for the NSGA solver are:

- `NonSmoothGASolver`: It is the main class, which contains the definition of the data structure, and solves the complete system.
- `NonLinearSolver`: It is an abstract class, which is used as an interface for the classes `PositionNonLinearSolver`, `SmoothNonLinearSolver` and `VelocityNonLinearSolver`. It also contains methods to solve nonlinear systems.
- `SmoothNonLinearSolver`: Concrete class, which contains the methods to compute the smooth contribution.
- `PositionNonLinearSolver`: Concrete class, which contains the methods to compute the position correction.
- `VelocityNonLinearSolver`: Concrete class, which contains the methods to compute the velocity jump.

The NSGA solver is implemented in the `NonSmoothGASolver2` class. The method called `computeSolution` solves the full system following the Algorithm 1. In order to exchange the information between the different classes a data structure called `NonLinearSolverContext` has been created. The members of this structure are mainly `VectorStr`, one `BlockMatrixStr` and one `TypeMatStr` that are linked to the Oofelie DB. It also contains the numerical parameters of the generalized-α method. This data structure is instantiated in the `NonSmoothGASolver2` and its memory address is taken as an input by the constructors of the three concrete classes, in that way they can directly assemble their contribution.

The `NonLinearSolver` is a class with numerical tools to solve a non linear system of the general form $\boldsymbol{r}(\boldsymbol{q}) = \boldsymbol{0}$ where $\boldsymbol{r}(\boldsymbol{q})$ is the residual function. The two implemented methods are `newtonRaphson` and `lineSearch`. The Newton-Raphson method is a root-finding algorithm which produces successively better approximations to the roots of a real-valued function. Despite the need of a Newton semi-smooth method to solve the frictional contact problem, in the `newtonRaphson` function a standard Newton-Raphson is implemented because all the computations related with the activation status of the unilateral constraints are managed at the element level, completly separated from the solver. In that way the methods implemented in the `NonLinearSolver` are easy to mantain and reusable for other functionalities.

The Newton-Raphson method only converges locally, which can cause convergence problems in some cases, for example if the starting point is out of the convergence area or the derivate of the function whose root is sought is discontinuous [93]. Therefore in the second method a Line Search (LS) algorithm has been implemented following the approach presented by Wriggers [94]. The line search is a modification of the Newton-Raphson method. When the residue of iteration $i+1$ is greater than the value obtained in the previous iteration i the line search is applied, which can be seen as a damping in the corrections to avoid to go outside of the attractive region. This reduction is represented by α_i; which is a numeric value between 0 and 1, as it is shown in Eq. (5.1).

$$\boldsymbol{q}_{n+1}^{i+1} = \boldsymbol{q}_{n+1}^{i} + \alpha_i \Delta \boldsymbol{q}_{n+1}^{i+1} \tag{5.1}$$

This parameter α_i could be chosen heuristically, but to improve the performance it is convenient to compute its optimal value. Since the associated solution for $\boldsymbol{r}(\boldsymbol{q})^T \boldsymbol{r}(\boldsymbol{q})$ is also solution for $\boldsymbol{r}(\boldsymbol{q})$, it makes sense to express this as a function only dependent on α_i

$$\boldsymbol{f}(\alpha_i) = -\boldsymbol{r}(\boldsymbol{q}_i + \alpha_i \Delta \boldsymbol{q}_{i+1})^T \boldsymbol{r}(\boldsymbol{q}_i + \alpha_i \Delta \boldsymbol{q}_{i+1}) \tag{5.2}$$

and minimize it to obtain the optimal α_i. The minimization of Eq. (5.2) is done using the secant method, which is easy to implement since it only involves the evaluation of $\boldsymbol{r}$.

The Algorithms 2 and 3 show how these two methods have been implemented,

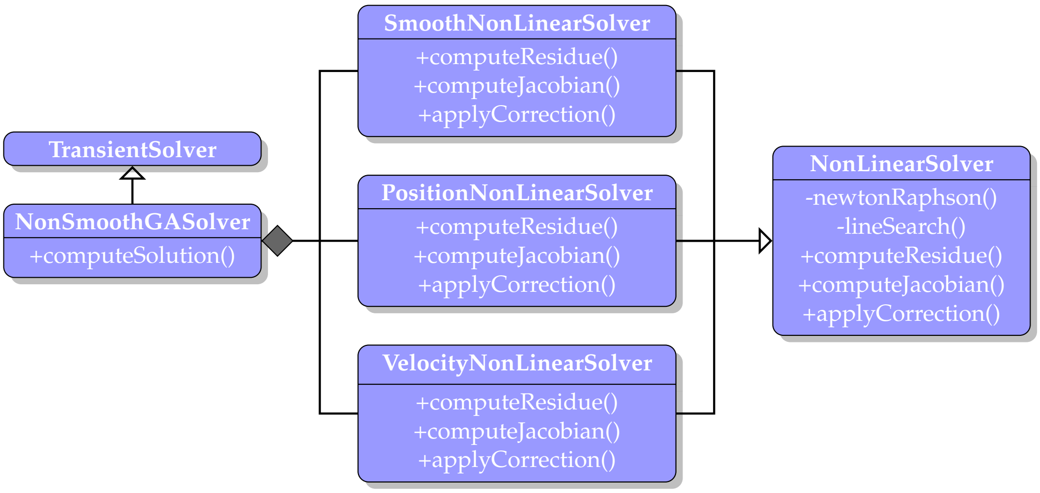


FIGURE 5.4: UML diagram: Nonsmooth Generalized-α Solver

where e is the relative error defined as the ratio between the norm of the residue and the norm of the residue plus a certain reference value, $e = \|\boldsymbol{r}_i\| / \|\boldsymbol{r}_i + reference\|$, e_{ls} is the relative error used in the line search method, i_{max} is the maximum number of allowed iterations and the criterion expression used for checking the convergence is the one presented in Eq. (3.55).

These two methods call the three defined virtual methods `computeResidue`, `computeJacobian` and `applyCorrection`, which are fully defined in the concrete classes that inheritate from the `NonLinearSolver` class, having a different behaviour for each subproblem, smooth, position and velocity. The full interaction between these classes is shown in the UML diagram in Fig. 5.4, where the standard notation $+$ and $-$ denotes public and private methods, respectively.

Algorithm 2 NonLinearSolver::newtonRaphson() implementation

```
$\boldsymbol{q}_{n+1}^0 = \boldsymbol{q}_n$
$\mathbf{r}^0 = \mathbf{r}(\boldsymbol{q}_{n+1}^0)$
$i = 0$
Compute $e(\boldsymbol{q}_{n+1}^0)$
**while** $(i < i_{max}$ and $e > tol)$ **do**
    Compute Jacobian matrix $\boldsymbol{J}$
    Compute corrections $\Delta\boldsymbol{q}_{n+1}^{i+1}$: Solve $\boldsymbol{J}(\boldsymbol{q}_{n+1}^i)\Delta\boldsymbol{q}_{n+1}^i = -\boldsymbol{r}(q_{n+1}^i)$
    Apply correction: $\boldsymbol{q}_{n+1}^{i+1} = \boldsymbol{q}_{n+1}^i + \Delta\boldsymbol{q}_{n+1}^{i+1}$
    Compute residue $\boldsymbol{r}(\boldsymbol{q}_{n+1}^{i+1})$
    Compute $e(\boldsymbol{q}_{n+1}^{i+1})$
    $i = i + 1$
**end while**
```

The contribution of the three different problems can be computed using the existing methods implemented in the element library. Equations (3.34), (3.39) and (3.41) to compute the residue for the smooth motion, position correction and velocity jumps, respectively, present a similar structure, the same occurs for the tangent matrix presented at Eq. (3.50), (3.51) and (3.52). This allows us to call the existing

Algorithm 3 NonLinearSolver::lineSearch() implementation

```
$\boldsymbol{q}^0_{n+1} = \boldsymbol{q}_n$
$\mathbf{r}^0 = \mathbf{r}(\boldsymbol{q}^0_{n+1})$
$i = 0$
Compute $e(\boldsymbol{q}^0_{n+1})$
while $(i < i_{max}$ and $e > tol)$ do
    Compute Jacobian matrix $\boldsymbol{J}$
    Compute $\Delta\boldsymbol{q}^{i+1}_{n+1}$: Solve $\boldsymbol{J}(\boldsymbol{q}^i_{n+1})\Delta\boldsymbol{q}^i_{n+1} = -\boldsymbol{r}(q^i_{n+1})$
    Apply corrections: $\boldsymbol{q}^{i+1}_{n+1} = \boldsymbol{q}^i_{n+1} + \Delta\boldsymbol{q}^{i+1}_{n+1}$
    Compute residue $\boldsymbol{r}(\boldsymbol{q}^{i+1}_{n+1})$
    $e_{LS} = e$
    Compute $e(\boldsymbol{q}^{i+1}_{n+1})$
    if $e > e_{LS}$ then
        Compute $\alpha_{optimal}$
        Apply correction: $\boldsymbol{q}^{i+1}_{n+1} = \boldsymbol{q}^i_{n+1} + \alpha_{optimal}\Delta\boldsymbol{q}^{i+1}_{n+1}$
        Compute residue $\boldsymbol{r}(\boldsymbol{q}^{i+1}_{n+1})$
        Compute $e(\boldsymbol{q}^{i+1}_{n+1})$
    end if
    $i = i + 1$
end while
```

methods using as arguments pointers to different locations in the DB to compute their contribution to the residue or the tangent matrix at each level (smooth, position or velocity). For example, the `nsmth_inert_forces` is called at smooth, position and velocity levels using $\dot{\tilde{\boldsymbol{v}}}$, $\boldsymbol{U}$, and $\boldsymbol{W}$ as argument, respectively.

It is important to remark that not all the results are stored in the DB for all the time steps. The only variables that are stored for all the time steps are the positions $\boldsymbol{q}_n$, velocities $\boldsymbol{v}_n$, accelerations $\dot{\tilde{\boldsymbol{v}}}_n$ and the Lagrange multipliers $\tilde{\boldsymbol{\Lambda}}_n$, $\boldsymbol{\nu}_n$ and $\boldsymbol{\Lambda}_n$. Once convergence is achieved these variables are stored in the DB `Analysis` `DYNAMIC_PO` for the current time step. The other variables $\tilde{\boldsymbol{q}}_{n+1}$, $\boldsymbol{U}_{n+1}$, $\tilde{\boldsymbol{v}}_{n+1}$ and $\boldsymbol{W}_{n+1}$ are only stored during the iteration procedure and are linked to the `Analysis` `CURRITERATION_PO`.

The method `computeResidue` uses the standard methods `inert_forces`, `dissipation_forces` and `int_forces` which are already implemented in the elements and also the new ones `nsmth_int_forces_smo`, `nsmth_int_forces_pos` and `nsmth_int_forces_vel`. The new frictional contact element is the only one in which these three new methods are explicitly implemented. For other existing elements, since these new functions are not implemented, the standard function of the elements `fill_inter_forc` is called. Only the elements involving bilateral constraints have been slightly modified to introduce these three functions, which call the function `fillConstraintsAtPositionLevel` or `fillConstraintsAtVelocityLevel`.

5.2.2 Frictional contact element class

The friction model is added at the element level, using a frictional contact element as presented in the Chapter 4. This element has been implemented in Oofelie in an element class called `oeContactNodeToFaceWithFrictionAndRotation`. It follows the same structure as the rest of Oofelie elements. In the constructor, the nodes are defined, and the standard method `toDofSet` associates or creates and associates the DOFs regarding to this element. The frictional contact element involves twelve DOFs, three translations of the slave node, three translations and three rotations of the node attached to the master surface, and three Lagrange multipliers which represent the reaction forces due to the impacts. One of these Lagrange multipliers is in the normal direction and the other two to acount for the frictional effects in the tangential direction.

The contribution of the element is split in three different parts, following the idea of the solver, the smooth variables, the position corrections and the velocity jumps. As presented in the section 4.1 the frictional contact only contributes to the residue when the unilateral constraints are active, otherwise it has no influence on the motion. Therefore the contribution to the residue is done by two methods: `fill_nsmth_int_forc_pos` and `fill_nsmth_int_forc_vel`, computing the Eqs. (4.35) and (4.45) respectively.

In the case of the tangent matrix a contribution is evaluated for the three levels. The methods: `fill_nsmth_tang_pos_ns` and `fill_nsmth_tang_vel_ns` compute the Eqs. (4.36) and (4.46) respectively, and `fill_nsmth_tang_smo_ns` computes the contribution to the smooth variables. This last contribution is the negative identity matrix for the components related to the Lagrange multipliers DOFs. Indeed, even if there is no unilateral constraint, the number of DOFs is the same for all three subproblems (smooth variables, position corrections and velocity jumps) to simplify the implementation. Inside these methods some mathematical operations are repeated. Therefore several auxiliary methods have been implemented to simplify the code.

5.3 NSGA and FEM applied to the woodpecker toy example

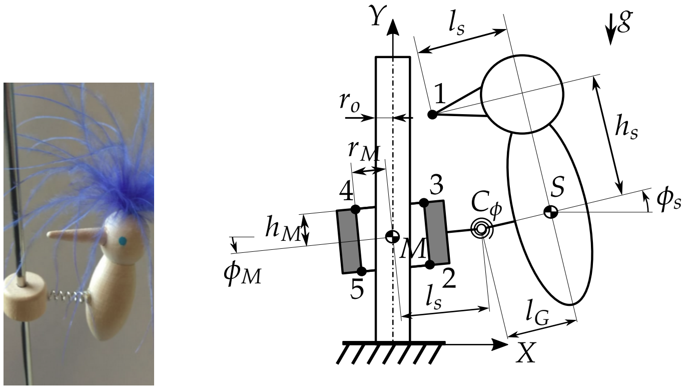
5.3.1 Introduction to the woodpecker toy

The study of the woodpecker toy dynamics sets up a good example to show the capabilities of the nonsmooth generalized-α implemented in a general purpose finite element code to solve multibody dynamics problems involving nonsmooth phenomena.

The woodpecker toy is a frequent example in papers dealing with nonsmooth contact dynamics methodologies. It involves several rigid bodies, which undergoes impacts and dynamic transitions between the gap, stick and slip states. The toy consists of a pole, a sleeve with an internal diameter slightly larger than the pole diameter, a woodpecker body with the beak and a spring which attaches the bird's body with the sleeve, see Fig. 5.5(a). It is a non conservative system where the gravity acts as the only energy source. The gap between the sleeve and the pole is small so that, the potential energy is not only transformed into kinetic energy but also dissipated through the frictional impacts, ending in a stable limit cycle. The example was first introduced by Pfeiffer [95] where the loss of energy due to the impacts is represented by an equivalent heuristic model. Glocker and Pfeiffer [96] studied the problem including a contact model for the beak. The equations of motion were based on a Coulomb friction model and a Poisson impact law, and their solution was computed using the Moreau-Jean scheme. In an experiment, these authors measured a falling distance per cycle of 5.3 mm and a limit cycle frequency of $f = 9.2$ Hz. The problem was also studied in [97, 98, 99, 100]. In these references, a minimal set of coordinates is used to describe the woodpecker toy: the vertical position of the sleeve y, the sleeve rotation ϕ_M, and the bird rotation ϕ_S. Also, the sleeve displacement in the horizontal direction is neglected, so that symmetry can be used to simplify the example, and small rotations approximations are considered. Leine [99] made a bifurcation analysis of the system, and an exhaustive explanation of all the motion phases. Glocker [101] studied the problem with the same assumptions but using the Newton impact law. Slavic and Boltezar [102] adapted the methodology to consider unilateral constraints between bodies with arbitrary shapes, and they took the beak shape into account. Recently, Charles et. al [100] presented an algorithm for problems with impacts and friction, where the woodpecker toy model from Glocker [96] is used as a benchmark test.

All the previous works dealing with the nonsmooth dynamics of the woodpecker toy specifically deduced the equations of motion for the considered problem resulting in a model with a minimal set of coordinates. They assumed small rotations and neglected the horizontal displacement of the sleeve, except for Slavic and Boltezar [102]. For all of them, the numerical solution was also obtained using first-order time integrators, such as the Moreau-Jean scheme.

The use of a model with a minimal set of coordinates has the advantage that the equations of motion are compact and only unilateral constraints are involved. However, this approach can hardly be automatized in a general purpose software and the scalability to more complex mechanisms is hardly possible. Using a general MBS formulation, such as the finite element approach, the equations of motion can be assembled automatically considering large rotation. It should be remarked that with the adoption of this paradigm, in addition to unilateral constraints, bilateral constraints dealing with kinematic joints and rigid bodies will have to be managed. Using the nonsmooth generalized-α method, both types of constraints can be exactly satisfied without any drift at position and velocity levels, so the treatment of bilateral constraints does not represent any problem.



(a) Woodpecker toy.

(b) Model (not to scale).

FIGURE 5.5: Woodpecker example.

5.3.2 Problem definition for the woodpecker toy

The woodpecker toy is modeled as a mechanical planar system, as shown in Fig. 5.5(b). The translations in z direction and rotations around x and y axes are imposed to zero. Table 5.1 gives the parameters of the model used in the numerical simulation according to [95]. In Fig. 5.5(b), the letters M and S make reference to the sleeve and the woodpecker body respectively, and the numbers 1-5 are the IDs of the contact elements accordingly specified in Table 5.2: pole-beak (ID: 1), down right sleeve-pole (ID: 2), upper right sleeve-pole (ID: 3), upper left sleeve-pole (ID: 4) and down left sleeve-pole (ID: 5).

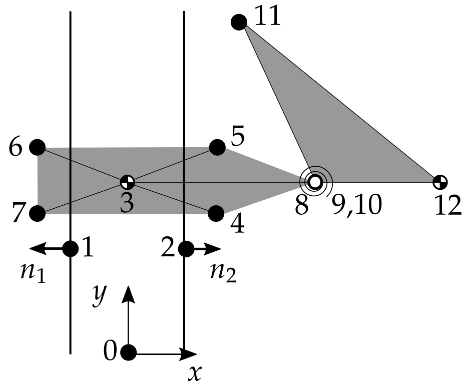


FIGURE 5.6: Woodpecker toy element discretization. Showing nodes ID.

5.3.3 Decomposition in elements

As finite element coordinates are used, the system is described through nodes and elements [5]. Each node defines a position and/or an orientation in the 3D space whilst the elements represent the interplay between them. In general, a system is modeled using a library of elements like rigid bodies, flexible beams, springs, hinges

TABLE 5.1: Woodpecker model parameters and initial conditions

Geometrical characteristics	
pole radius	$r_O = 0.0025$ m
sleeve radius	$r_M = 0.0031$ m
half sleeve height	$h_M = 0.0058$ m
distance: spring - COM sleeve	$l_M = 0.0100$ m
distance: spring - COM woodpeaker	$l_G = 0.0150$ m
distance: beak - COM woodpeaker	$h_S = 0.0200$ m
distance: beak - COM woodpeaker	$l_S = 0.0201$ m
Inertial properties	
sleeve mass	$m_M = 0.0003$ kg
woodpecker mass	$m_S = 0.0045$ kg
sleeve moment of inertia	$J_M = 5.0 \cdot 10^{-9}$ kg·m^2
woodpecker moment of inertia	$J_S = 7.0 \cdot 10^{-7}$ kg·m^2
Force elements	
angular stiffness	$c_\phi = 0.0056$ Nm/rad
gravity	$g = 9.81$ m/s^2
Contact parameters	
COR in normal direction	$\varepsilon_{N1} = 0.5$
COR in normal direction	$\varepsilon_{N2} = \varepsilon_{N3} = 0$
	$\varepsilon_{N4} = \varepsilon_{N5} = 0$
COR in tangential direction	$\varepsilon_{T1} = \varepsilon_{T2} = \varepsilon_{T3} = 0$
	$\varepsilon_{T4} = \varepsilon_{T5} = 0$
frictional coefficient	$\mu_1 = \mu_2 = \mu_3 = 0.3$
	$\mu_4 = \mu_5 = 0.3$
Initial conditions	
sleeve angular position	$\phi_{MO} = -0.1036$ rad
woodpecker angular position	$\phi_{SO} = -0.2788$ rad
woodpecker vertical velocity	$\boldsymbol{v}_o = -0.3411$ m/s
sleeve angular velocity	$\omega_{MO} = 0.0$ rad/s
woodpecker angular velocity	$\omega_{SO} = -7.4583$ rad/s
sleeve position X	$X_M = 0$ m

and so on. Then the contributions of the different elements are assembled numerically to obtain the discretized equations of motion of the system, which afterwards are integrated in time, in our case using the nonsmooth generalized-α method.

Flexible elements, like springs or beams, do not introduce constraints. These elements only contribute to the system with a force term that depends on the position, orientation and/or velocity of the element nodes. On the other hand, rigid bodies and hinges add bilateral constraints that connect the element nodes. Because of that, it is important to use an algorithm that imposes the constraints not only at velocity level but also at position level, otherwise a drift at position level can occur, as shown in [50]. Finally, contact conditions are modeled using a node-to-face frictional contact element.

In Fig. 5.6, the set of elements used to simulate the woodpecker example are shown, where the numbers refer to the nodes id, and the numbers inside a circle to the elements id. Two rigid elements, one hinge element, one torsional spring element and five frictional contact elements are used. In Table. 5.2 the list of elements and the nodes belonging to each one is shown. The rigid bodies connect all the element nodes with the center of mass of the element, which is the first node of the element. For each node that is connected to the center of mass, three bilateral constraints are introduced. In each contact element only two nodes are involved, the first one is the node attached to the contact surface. In this example the position of the surface contact node is fixed in time, but the element is capable to deal with the translation and rotation of the contact surface. The hinge element is composed of three nodes and introduces six bilateral constraints. The first two nodes are the position of the material points to be joined by the hinge, whilst the third node is used to store the relative angle of the hinge, which can be used to model a torsional spring or impose an external torque, to simulate the action of a motor for example. Three of the five bilateral constraints guarantee that the two nodes remain together. The next two fix the rotations about the two directions, that are perpendicular to the rotation axis of the hinge. The last constraint defines the value of the relative angle. A more extended explanation of the rigid, hinge and spring elements formulation can be found in Géradin and Cardona [5].

5.3.4 Elements contributions

This subsection presents the different contributions of the elements and shows that many of them can be evaluated from standard operators available in a nonlinear finite element code. The nonsmooth generalized-α method presents the advantage that the three sets of Eqs. (3.34), (3.37) and (3.41) have a similar structure to the smooth problem, allowing to reuse the implemented functions of the existing elements in the software. To assemble the set of equations presented in Eq. (3.34), (3.37) and (3.41), the functions of the elements are called several times with different input variables. The standard functions for the elements are:

TABLE 5.2: Finite elements description

Element ID	Element type	Description	Nodes
1	Friction Contact	Pole - Beak	2 11
2	Friction Contact	Pole - Sleeve	2 4
3	Friction Contact	Pole - Sleeve	2 5
4	Friction Contact	Pole - Sleeve	1 6
5	Friction Contact	Pole - Sleeve	1 7
6	Rigid Body	Sleeve	3 4 5 6 7 8
7	Rigid Body	Body bird	12 10 11
8	Hinge	Sleeve-Body	8 10 9
9	Spring	Torsional spring	0 9

- Bilateral internal forces (BIF): computes the reaction forces of the general form $\boldsymbol{g}_{\boldsymbol{q}}^{T}\boldsymbol{\lambda}$. , where $\boldsymbol{\lambda}$ is a vector with the same dimensions as the Lagrange multipliers. The input variables are the vector $\boldsymbol{\lambda}$ and the configuration variable $\boldsymbol{q}$.

- Flexible internal forces (FIF) $\boldsymbol{f}(\boldsymbol{q},\boldsymbol{v})$: computes internal forces of the elements that are not related with constraints, the inputs are the position and velocity vectors.

- Inertial forces (IF): computes $\boldsymbol{Ma}$, where $\boldsymbol{a}$ is a vector with the same dimensions as the acceleration. The input variables are $\boldsymbol{a}$ and the position vector.

- Constraints (C): computes the contribution of the position constraints, the input variable is the position vector.

- Velocity constraints (VC): computes the contribution of the velocity constraints. The input variables are the configuration variable and the velocity vectors at time n and $n+1$.

In the case of the spring element, which does not contribute to bilateral constraints or mass, the function FIF is only called three times, once for each subproblem, with the inputs $\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}$ at smooth level, $\boldsymbol{q}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}$ at position level and $\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}$ at velocity level.

In the hinge and rigid elements, the function BIF is called for the three subproblems, smooth, position and velocity with the input variables $\tilde{\boldsymbol{\lambda}}$, $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$ respectively. Also, to impose the constraints at position level, the function C is called with the position vector $\boldsymbol{q}_{n+1}$ as inputs. To impose the constraints at velocity level, the function VC is called for the smooth subproblem with $\tilde{\boldsymbol{q}}_{n+1}, \tilde{\boldsymbol{v}}_{n+1}$ as inputs, and it is called again for the velocity subproblem with the inputs $\boldsymbol{q}_{n+1}, \boldsymbol{v}_{n+1}$. Additionally, in the rigid element the function IF is called with the input values $\tilde{\boldsymbol{q}}_{n+1}, \dot{\tilde{\boldsymbol{v}}}_{n+1}$ for the smooth subproblem, $\boldsymbol{q}_{n+1}, \boldsymbol{U}_{n+1}$ for the position subproblem and $\boldsymbol{q}_{n+1}, \boldsymbol{W}_{n+1}$ for the velocity subproblem.

The contact element has been developed explicitly for this algorithm, therefore, since it is massless and it does not contribute to the smooth subproblem, its contribution is condensed in two functions:

- Nonsmooth residue position (NRP): computes all the contributions of the contact element to the position subproblem. The input variables are the position vector at time $n+1$ and the Lagrange multipliers $\boldsymbol{\nu}$,
- Nonsmooth residue velocity (NRV): computes all the contribution of the contact element to the velocity subproblem. The input variables are the velocity vector at time n and $n+1$, the position vector at time $n+1$ and the Lagrange multipliers $\boldsymbol{\Lambda}$,

5.3.5 Numerical results

In this section, the results for the woodpecker toy are presented. Two different cases are studied. In case I, the horizontal displacement of the sleeve center of mass (x_m) is fixed, whereas in case II this displacement is free. In case I since the displacement of the center of mass is fixed, it is possible to apply a symmetry condition and the contacts 4 and 5 can be removed [96, 101, 98, 102].

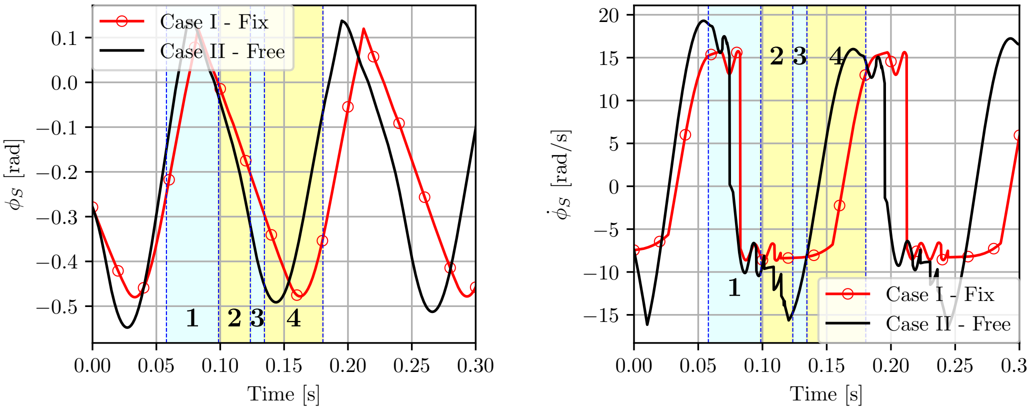
For both cases the convergence of the numerical solution obtained with the proposed methodology is analyzed. For that purpose, as in previous chapters, the error over the full time interval divided into N time increments h is evaluated using the L^1 norm, which is defined as shown in Eq. (4.47). Since no analytical solution is available, the numerical solution corresponding to a smaller time increment ($h = 10^{-5}$ s) is taken as reference solution.

All figures from 5.7(a) to 5.9(b) show the comparison of results between the cases I and II computed using a time step $h = 10^{-4}$ s. The rotation ϕ_S and angular velocity $\dot{\phi}_S$ for the woodpecker (node 12), are shown in Figs. 5.7(a) and 5.7(b), respectively. In Fig. 5.7(a) it can be observed that there is one dominant nonsmooth point per cycle, corresponding to the instant when the beak impact occurs.

Figs. 5.8(a)-5.9(b) compare the behavior of the sleeve for both cases. Figs. 5.8(a) and 5.8(b) show the rotations and the angular rotations of the sleeve, whilst Figs. 5.9(a) and 5.9(b) show the vertical displacements of the sleeve center of mass and its time derivative, respectively. In order to facilitate the understanding of the sleeve contact process for the case II, four zones are highlighted in blue and yellow in the charts. During the first phase the lower left corner of the sleeve (contact 5) impacts the rod. In the second phase the contacts 2 and 4 bounce against the pole entering in slip during short periods of time, until the moment where both remain closed, as can be seen in Figs. 5.10(a) and 5.10(b) where the gaps are shown. In this instant, the phase 3 starts and contacts 2 and 4 are active in slip mode, as can be seen in Fig. 5.9(a), i.e., the sleeve keeps falling. Finally, in phase 4 the sleeve contacts 2 and 4 are in stick mode, therefore, the sleeve does not have any motion and the curve remains flat, as can be observed in Figs. 5.8(a)-5.9(b). Besides in Figs. 5.10(b) it is possible to observe that for case II the upper right corner (contact 3) is never closed.

Comparing the two cases, we can observe that the decision to neglect the horizontal displacements of the sleeve center of mass has a considerable influence on the behavior of the woodpecker. The cycle frequency for case I is $f_I = 7.66$ Hz and $f_{II} = 7.96$ Hz for case II. The vertical displacements of the sleeve in each cycle is $h_I = 14.3$ mm and $h_{II} = 7.6$ mm for case I and II respectively. Despite their similar behavior and close frequencies, the restriction of the displacements extends the sticking face, slowing down the fall of the sleeve, a phenomenon that is clearly observable in Fig. 5.9(b). In case II, after tuning the friction coefficient, we observe that for $\mu = 0.35$ the frequency is $f^*_{II} = 8.3$ Hz and the sleeve vertical displacements $h^*_{II} = 5.3$ mm. These results are quite close to the experimental ones observed by Glocker and Pfeiffer [96], where the frequency is slightly bigger, $f_{exp} = 9.2$ Hz, and the sleeve vertical displacement is the same $h_{exp} = 5.3$ mm.

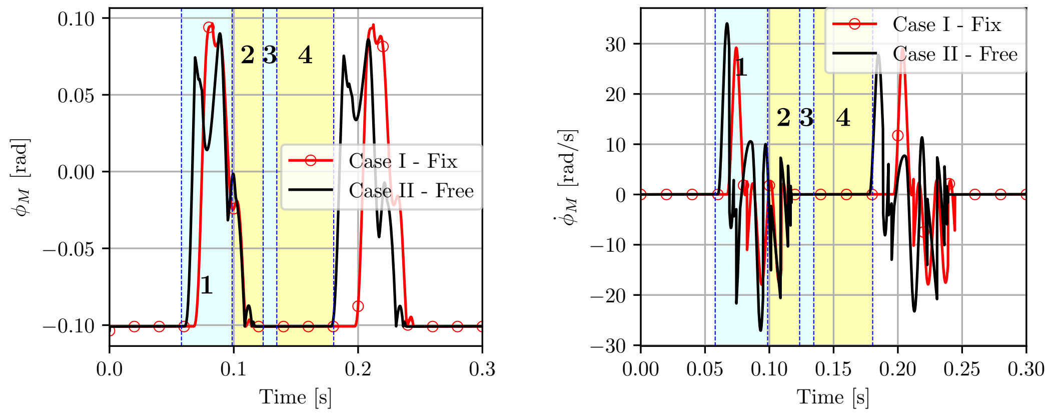
For case II, since the sleeve center of mass displacement in the horizontal direction is not fixed, its position and velocity are shown in Figs. 5.11(a) and 5.11(b) respectively. As expected, the bilateral constraints are satisfied at position level without any drift. In order to highlight this property, Fig. 5.12 shows the bilateral constraints introduced by the sleeve rigid body, for the node 4. The error obtained for this bilateral constraint is below the allowed tolerance.



(a)

(b)

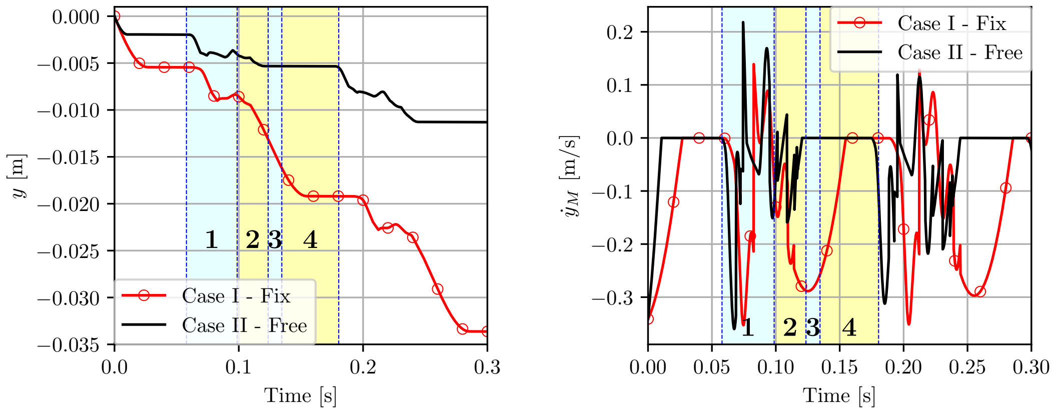
FIGURE 5.7: Woodpecker comparison: (a) Rotations (b) Angular velocities



(a)

(b)

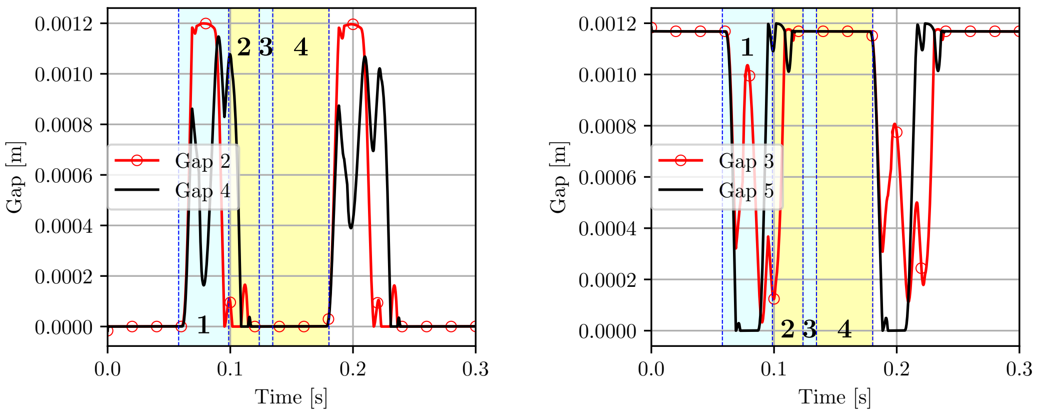
FIGURE 5.8: Sleeve comparison: (a) Rotations (b) Angular velocities



(a)

(b)

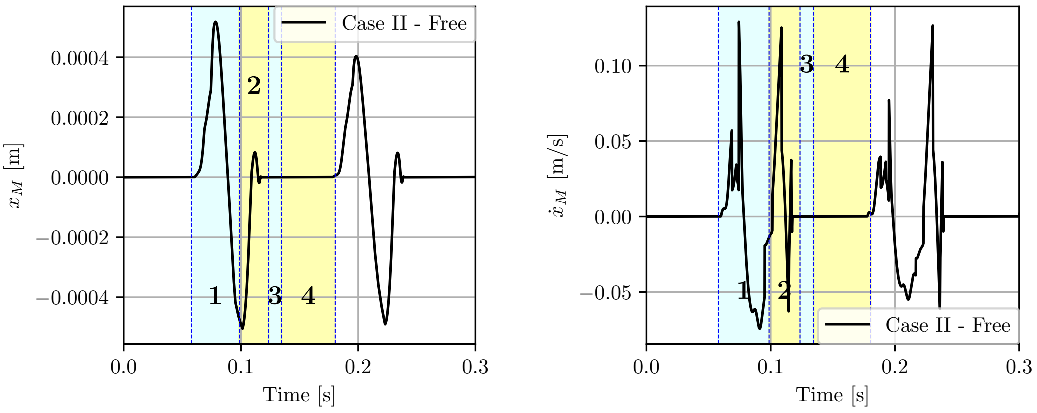
FIGURE 5.9: Sleeve comparison: (a) Vertical displacement (b) Vertical velocity



(a)

(b)

FIGURE 5.10: CASE II (Free): (a) Sleeve contact gaps 2 and 4. (b) Sleeve contact gaps 3 and 5.



(a)

(b)

FIGURE 5.11: CASE II (Free): (a) Sleeve horizontal displacement. (b) Sleeve horizontal velocity.

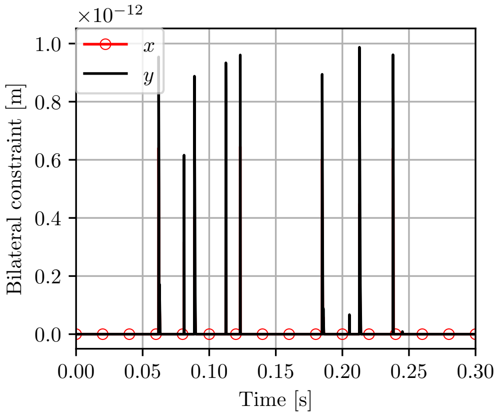
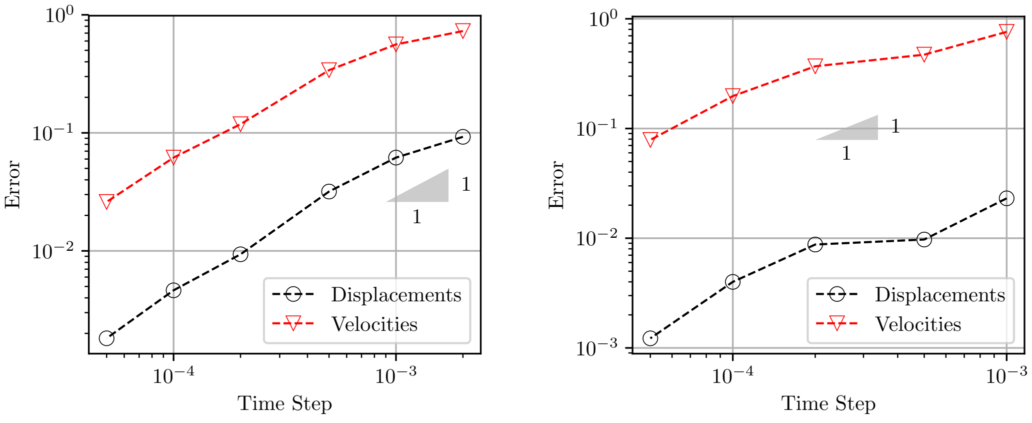


FIGURE 5.12: CASE II (Free) - Bilateral constraints for node 4.



(a)

(b)

FIGURE 5.13: Convergence study: (a) CASE I (Fixed). (b) CASE II (Free)

For both cases, Figs. 5.13(a) and 5.13(b) show the convergence rate obtained for the y component of the displacement and velocity fields of node 3. As it can be observed, an order 1 convergence is achieved.

Finally, an animation of the woodpecker toy simulation compares the results obtained for the case where the horizontal component of the sleeve displacement is fixed to zero (left side) and where it is totally free to move (right side). It can be found in the following link (**https://youtu.be/6wb1mCfF2t8**).

5.4 Summary and concluding remarks

In this chapter an introduction to the Oofelie code was done, explaining the scripting methodology and the data base. Oofelie has arisen as good choice to implement the methodology presented in this thesis due to its wide element library, its modularity, and the accessibility of the source code at the University of Liège. An explanation of the implementation of the NSGA solver and the frictional contact element has been done.

With this methodology, the woodpecker toy was simulated considering large rotations for two different cases, i) the horizontal displacement of the sleeve center of mass is fixed, and ii) this displacement is free. It was observed that this modeling assumption has a considerable influence as the vertical displacement of the sleeve is reduced from 14.3 mm in case I to 7.6 mm in case II, that is closer to the experimental measurement of 5.3 mm. The agreement with the experimental results could be further improved by an adjustment of the frictional coefficient.

Chapter 6

Conclusions

6.1 Summary

The modeling of mechanical systems dynamics involving rigid and flexible bodies is a mature field, with well established techniques available in several industrial simulation software. But, the dynamic simulation of mechanical systems involving frictional impacts is a challenging problem for the scientific community because of the high non-linearities and non-smoothness stemming from the Signorini's conditions and the Coulomb friction's law. There is an industrial demand to have reliable tools to simulate the aforementioned systems in an easy and systematic way. Therefore, this dissertation addresses the simulation of mechanical systems involving frictional contacts in the framework of multibody dynamics and presents a FEM solution based on nonsmooth techniques, and its implementation in the general purpose finite element software, Oofelie [66]

The presented methodology gathers the possibility to represent rigid bodies, flexible bodies and frictional contacts under the umbrella of the FEM approach. To achieve this, the rigid bodies are represented as a set of nodes, one of them being located at the COM and other nodes being rigidly linked to the COM by bilateral constraints. The connections between different rigid or flexible bodies is also done by bilateral constraints, for example to model hinges or spherical joints. The flexible bodies are discretized in space using flexible elements which allow deformation between their nodes. A node-to-face contact element is now included in the element library, which defines unilateral constraints between the master face and the slave node. Finally, the contribution of these elements are computed and numerically assembled together in an automatic way and the system is solved.

The original contribution of the thesis can be divided in two different parts. On the one side, a new version of the NSGA solver is developed as an evolution of the previous solver presented in [50]. On the other side, the development of a frictional contact element compatible with the NSGA solver is developed and tested. This element is based on a combination of a Newton impact law and a Coulomb frictional law and the methodology proposed by Alart and Curnier [43].

The proposed fully decoupled NSGA integration method was presented in Chap. 3 and has the following characteristics:

- The most distinctive feature of the new algorithm is that the sub-problem defining the smooth part of the motion is strictly independent of the position correction and of the velocity jump, so that the solution of the three sub-problems could be performed in a purely decoupled sequential manner. The algorithm is implemented as a sequence of three sub-problems to be solved at each time step.

- Like its predecessor, it does not suffer from any drift phenomena as it imposes the constraints both at position and at velocity levels.

- The algorithm is well suited for problems with vibration effects as it integrates the smooth component of the motion with the second order accurate generalized-α method, whereas a first order scheme is used for the impulsive contributions. This allows to finally control the numerical dissipation, something that is not possible with the first order integrators which are usually used to solve this kind of problems and introduce a much higher numerical dissipation.

- The algorithm does not require modifications in existing finite elements of the library, in other words it is minimally intrusive.

Four numerical examples were presented for frictionless problems, showing that the proposed method improves the robustness for problems involving nonlinear bilateral constraints and/or flexible elements, without deteriorating the accuracy of the original NSGA method. The number of iterations was reduced and much larger time steps could be adopted.

A variant of the new method, in which the $\boldsymbol{f}^p$ and $\boldsymbol{f}^*$ terms were neglected, was analyzed in the examples. The computed results showed that neglecting those terms could lead to results of bad quality if sufficiently small stepsizes were not adopted. Hence, it was recommended to take those terms into account in the implementation of the decoupled algorithm.

The application of this new algorithm to deal with frictional contact problems was presented in Chap. 4. The main features of the frictional contact element are:

- The enforcement of the unilateral constraints is done following an augmented Lagrangian approach, similarly to the one proposed Alart and Curnier [43] for the quasi-static problems.

- A Coulomb frictional law is used, then the complementarity problem for the normal and frictional contact problem is solved at each time step using a monolithic Newton semi-smooth method in a fully implicit approach, which allows to reproduce the slip and stick status of the contact.

- The jumps in the velocity field are modeled using a Newton impact law, which involves two restitution coefficients, one in the normal direction and another in the tangential direction.

- The final solution is independent of the value of the penalty parameters, both in the tangential and normal directions. Their values only affect the convergence rate.
- The proposed contact model also satisfies exactly the contact constraints at position level and velocity level, which means that no penetration is observed up to the tolerance of the solver.
- The node-to-face contact element can account for large displacements and rotations of the master face in three dimensions.

An important remark is the necessity to impose the Coulomb friction sticking constraints at position level in the proposed framework. In order to understand this, recall that the time integration scheme is characterized by three decoupled subproblems, and that the correction at position level is blind to any correction done at velocity level for the same time step. Therefore, if the friction constraints in the tangential direction are imposed only at velocity level some non-physical behaviour can be observed at position level, as it was shown with an example in Section 4.2.1.

The proposed methodology was successfully applied to study different mechanism configurations consisting of flexible and/or rigid bodies and to observe the friction effects on the dynamic response of the system. The results of these examples were compared with analytical or previous numerical solutions obtained by other authors showing good agreement and convergence rate, demonstrating the viability of the method and all the aforementioned features.

The possibility to embed these technique in an existing FEM software was proven in Chap. 5. An introduction to the Oofelie code was done, explaining the scripting methodology and the data base. Oofelie has arisen as good choice to implement the methodology presented in this thesis due to its wide element library and its modularity. This software and its finite element library were extended to nonsmooth frictional contact and impact problems, and due to the small intrusiveness of the algorithm many existing elements were reused without major modifications. An explanation of the implementation of the NSGA solver and the frictional contact element is included in the manuscript. It is important to remark that the Newton solver for problems involving frictional contact conditions suffers from convergence issues, resulting in a large number of iterations. In order to solve this problem, a line-search solver was used reducing the number of iterations needed to reach convergence.

Finally, with this methodology, the woodpecker toy example was simulated considering large rotations for two different cases, i) the horizontal displacement of the sleeve center of mass is fixed, and ii) this displacement is free. It was observed that this modeling assumption has a considerable influence as the vertical displacement of the sleeve is reduced from 14.3 mm in case I to 7.6 mm in case II, that is closer to the experimental measurement of 5.3 mm. The agreement with the experimental results could be further improved by an adjustment of the frictional coefficient. Therefore, the different examples of the thesis demonstrate the capacity of the proposed methodology to simulate MBS involving rigid bodies, flexible bodies, frictional contacts, including dynamic transitions between stick and slip contact states.

6.2 Further work

Let us discusses other points that could be further investigated in this field.

In a recent work [71] a new version of the NSGA method was proposed. The main difference is that the smooth motion is defined by imposing bilateral constraints and the active unilateral constraints at the acceleration level. This eliminates some spurious numerical oscillations of the constraints that generally occur after an impact, resulting in an increase of the robustness and stability of the proposed algorithm. It would be interesting to study the possibility to include these acceleration constraints in the presented formulation of the solver while preserving the decoupled formulation of the different sub-problems.

In this work an element for rigid to rigid or rigid to flexible contact was studied. From a geometric viewpoint, the element is described as slave node colliding against a master face. The method could be extended to the interaction between more complex geometries. This can be done by expressing analytically the geometries and attach them to the rigid body center of mass node. For example, basic geometries like sphere to sphere or a cylinder to surface contact element could be considered. A more general approach would be based on a geometrical description by spline functions that are used by computer-aided design (CAD) software packages.

In order to simulate a contact between two flexible bodies, some extra techniques should be implemented. For example, mortar technique proposed for quasi-static problems [37] could be extended to dynamic problems.

With the evolution of flexible multibody systems simulations, the industrial interest to study complex mechanical interactions between highly flexible bodies, such as cable wires, fibers braiding, hair, fibers in shield structures, among other has arisen [103, 104]. Therefore, it would be interesting to extend the presented methodology to the case of beam to beam contacts.

The equations of motion with bilateral and unilateral constraints formulated as a CP can be interpreted as optimization problems. Within the mathematical programming community, several techniques have been developed to solve such problems [21]. In this thesis, the augmented Lagrangian approach with a Newton semi-smooth solver was tested resulting in a feasible solution. But the use of alternative solvers could be further explored.

The methodology presented in this thesis proved the capability to deal with redundant compatible constraints, but some problems arise when redundant incompatible constraints are present. To better understand these incompatibilities let us imagine a rigid pendulum connected to a fixed point by a hinge, if during the rotation, the pendulum impacts a fixed surface and the friction coefficient is large enough to enter in stick mode, then the bilateral constraints of the rigid body and the unilateral constraint in the tangential direction due to friction may become incompatible. This incompatibility is observed when a numerical solver with finite time steps is used, but is not presented in the exact analytical solution. This is an open question where further research is needed.

As presented in this work, single-contact impacts are properly modeled using a Newton impact law, even accounting for frictional effects. Nevertheless, when multiple impacts occur at the same instant it is not possible to get an accurate prediction of the system evolution in agreement with experimental observations if the classical Newton impact law is used. This phenomenon is noticeably observed in the Newton's cradle pendulum example. Therefore, more sophisticated impact laws could be considered to model and simulate multiple collisions. For example, Winandy and Leine [105] propose an instantaneous impact law for completely elastic multi-collisions of the 3-ball Newton's cradle able to reproduce the outcomes of experimental observation. Nevertheless, they conclude that no dissipation function exists for the proposed impact law; therefore, it can not be used in a general manner for an arbitrary MBS. Other methods, e.g. the binary collision model or the one presented by Han and Gilmore [106], are able to reproduce simple examples, but also show limitations to model higher dimensional examples [9]. Thus, a general impact law able to account for multiple impacts in high dimensional problems is still an open question.

The methodology has been implemented in the software Oofelie, which has an easy and intuitive graphical interface embedded in the CAD software package Siemens NX, but the frictional contact element and the solver are not interfaced yet. Therefore, it is an important future task in order to transfer easily and fast this simulation technology to industrial applications.

Bibliography

[1] D. Vallejo. “Dinámica de sistemas multicuerpo rígido-flexibles en Coordenadas Absolutas”. PhD thesis. Universidad de Sevilla, 2006.

[2] B. Leon, A. Morales, and J. Sancho-Bru. *From Robot to Human Grasping Simulation*. Springer, 2014.

[3] F. Pfeiffer and Ch. Glocker. *Multibody dynamics with unilateral contacts*. Wiley series in nonlinear science. Wiley, 1996.

[4] V. Acary. “Projected event-capturing time-stepping schemes for nonsmooth mechanical systems with unilateral contact and Coulomb’s friction”. In: *Computer Methods in Applied Mechanics and Engineering* 256 (2013), pp. 224–250.

[5] M. Géradin and A. Cardona. *Flexible Multibody Dynamics: A Finite Element Approach*. Wiley, 2001.

[6] O. A. Bauchau. *Flexible Multibody Dynamics*. Springer, 2011.

[7] T. Wasfy and A. Noor. “Computational strategies for flexible multibody systems”. In: *Applied Mechanics Reviews* 56 (2003), pp. 553–613.

[8] P. Wriggers. *Computational Contact Mechanics*. John Wiley and Sons, 2002.

[9] N. S. Nguyen and B. Brogliato. *Multiple Impacts in Dissipative Granular Chains*. Lecture Notes in Applied and Computational Mechanics. Springer, 2014.

[10] G. Gilardi and I. Sharf. “Literature survey of contact dynamics modelling”. In: *Mechanism and Machine Theory* 37 (2002), pp. 1213–1239.

[11] A. Signorini. *Sopra alcune questioni di elastostatica*. Atti della Societa Italian per il Progresso della Scienza, 1933.

[12] H. Olsson, K.J. Aström, C. Canudas De Wit, M. Gäfvert, and P. Lischinsky. “Friction models and friction compensation”. In: *Control* 4 (1998), pp. 176–195.

[13] J.H. Jellet. *A treatise on the theory of friction*. Foster and Co., 1872.

[14] E. Rabinowicz. *Friction and Wear of Materials*. John Wiley & Sons, 1995.

[15] K.L. Johnson. *Contact Mechanics*. Cambridge University Press, 1987.

[16] P.R. Dahl. *A solid friction model*. Tech. rep. The Aerospace Corporation, 1968.

[17] R. Stribeck. “Die Wesentlichen Eigenschaften der Gieitund Rollenlager–the key qualities of sliding and roller bearings”. In: *Zeitschrift des Vereines Seutscher Ingenieur* 46 (1902), pp. 1432–1437.

[18] A. J. Morin. "New friction experiments carried out at Metz". In: *Proceedings of the French Royal Academy of Sciences* 4 (1833), pp. 1–128.

[19] O. Reynolds. "On the theory of lubrication and its application to Mr. Beauchamp Tower's experiments, including an experimental determination of the viscosity of olive oil". In: *Philosophical Transactions of the Royal Society* 177 (1886), pp. 157–234.

[20] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit. "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction*". In: *Automatica* 30 (1994), pp. 1083–1138.

[21] V. Acary, M. Brémond, and O. Huber. "On Solving Contact Problems with Coulomb Friction: Formulations and Numerical Comparisons". In: *Advanced Topics in Nonsmooth Dynamics: Transactions of the European Network for Nonsmooth Dynamics*. Springer International Publishing, 2018, pp. 375–457.

[22] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit. "A survey of models, analysis tools and compensation methods for the control of machines with friction". In: *Automatica* 30, no 7 (1994), pp. 1083–1138.

[23] M. Bostan and P. Hild. "Weak Formulations and Solution Multiplicity of Equilibrium Configurations with Coulomb Friction". In: *Mathematical Modelling of Natural Phenomena* 4 (2009), pp. 147–162.

[24] B. Brogliato. *Nonsmooth Impacts Mechanics - Models, Dynamics and Control*. Springer, 2016.

[25] F. Marques, P. Flores, J. C.P. Claro, and H. M. Lankarani. "Modeling and analysis of friction including rolling effects in multibody dynamics: a review". In: *Multibody System Dynamics* 45 (2019), pp. 223–244.

[26] E. Pennestrí, V. Rossi, P. Salvini, and P. P. Valentini. "Review and comparison of dry friction force models". In: *Nonlinear Dynamics* 83 (2016), pp. 1785–1801.

[27] V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and electronics*. Springer, 2008.

[28] R. I. Leine and N. van de Wouw. *Stability and Convergence of Mechanical Systems with Unilateral Constraints*. Springer, 2008.

[29] M. Fratarcangeli, V. Tibaldo, and F. Pellacini. "Vivace: a Practical Gauss-Seidel Method for Stable Soft Body Dynamic". In: *ACM Transactions on Graphics* 35 (2016), 214:1–214:9.

[30] F. Dubois, M. Jean, M. Renuf, R. Mozul, A. Martin, and M. Bagneris. *LMGC90*. CSMA. 2011.

[31] F. Jourdan, P. Alart, and M. Jean. "A Gauss-Seidel like algorithm to solve frictional contact problems". In: *Computer Methods in Applied Mechanics and Engineering* 155 (1998), pp. 31–47.

[32] T. Liu and M. Y. Wang. “Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping and Gauss-Seidel methods”. In: *IEEE Transactions on Automation Science and Engineering* 2 (2005), pp. 19–31.

[33] E. A. H. Vollebregt. “A Gauss-Seidel Type Solver for Special Convex Programs, with Application to Frictional Contact Mechanics”. In: *Journal of optimization theory and applications* 87 (1995), pp. 47–67.

[34] K. Erleben. “Rigid body contact problems using proximal operators”. In: Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 2017, pp. 1–12.

[35] N. Kikuchi and J.T. Oden. *Contact Problems in Elasticity: A Study of variational Inequalities Constrains and Finite Element Method*. SIAM, Philadelphia, 1988.

[36] T. Belytschko and M.O. Neal. “Contact-impact by the pinball algorithm with penalty and lagrangian-methods”. In: *International Journal for Numerical Methods in Engineering* 31 (1991), pp. 547–572.

[37] F.J. Cavalieri, V. Fachinotti, and A. Cardona. “A mortar contact algorithm for three-dimensional elasticity problems”. In: *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 28 (2012), pp. 80–92.

[38] H. Lankarani and P. Nikravesh. “Continuous contact force models for impact analysis in multibody analysis”. In: *Nonlinear Dynamics* 5 (1994), pp. 193–207.

[39] B. Brogliato. *Nonsmooth Mechanics*. Springer, 1999.

[40] V. Acary. “Higher order event capturing time-stepping schemes for nonsmooth multibody systems with unilateral constraints and impacts”. In: *Applied Numerical Mathematics* 62 (2012), pp. 1259 –1275.

[41] M.R. Hestenes. “Multiplier and gradient methods”. In: *Journal of Optimization Theory and Applications* 4 (1969), pp. 303–320.

[42] M. Powell. “A method for nonlinear constraints in optimization problems. In: Fletcher R. (ed) Optimization”. In: *Academic Press, London.* (1969), pp. 283–298.

[43] P. Alart and A. Curnier. “A mixed formulation for frictional contact problems prone to Newton like solution methods”. In: *Computer Methods in Applied Mechanics and Engineering* 92 (1991), pp. 353–375.

[44] P.M.A. Areias, J.M.A. César de Sá, and C. A. Conceicão António. “Algorithms for the analysis of 3D finite strain contact problems”. In: *International Journal for Numerical Methods in Engineering* 61 (2004), pp. 1107–1151.

[45] J.C. Simo and T.A. Laursen. “An augmented lagrangian treatment of contact problems involving friction”. In: *Computers and Structures* 42 (1992), pp. 97–116.

[46] N.M. Newmark. “A method for computation for structural dyanmics”. In: *ASCE Journal of the Engineering Mechanics Division* 85 (1959), pp. 67–94.

[47] H. Hilber, T. Hughes, and R. Taylor. “Improved numerical dissipation for time integration algorithms in structural dynamics”. In: *Earthquake Engineering and Structural Dynamics* 5 (1977), pp. 283–292.

[48] J. Chung and G. Hulbert. “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method”. In: *ASME Journal of Applied Mechanics* 60 (1993), pp. 371–375.

[49] T.A. Laursen. *Computational Contact and Impact Mechanics.* Springer: Berlin, Heidelberg, 2002.

[50] O. Brüls, V. Acary, and A. Cardona. “Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized-α scheme”. In: *Computer Methods in Applied Mechanics and Engineering* 281 (2014), pp. 131–161.

[51] J.-J. Moreau. “Standard inelastic shocks and the dynamics of unilateral constraints”. In: *G.D. Piero, F.M. Eds (Eds.), Unilateral Problems in Structural Analysis, Springer-Verlag* (1985), 173–221.

[52] J.-J. Moreau. “Bounded variation in time”. In: *Topics in Nonsmooth Mechanics* (1988), pp. 1–74.

[53] R.J. Le Veque. *Numerical Methods for Conservation Laws*. Birkhauser Verlag, Basel, 1990.

[54] Ch. Glocker. *Set-Valued Force Laws*. Lecture Notes in Applied and Computational Mechanics. Springer-Verlag Berlin Heidelberg, 2001.

[55] L. Paoli and M. Schatzman. “A Numerical Scheme for Impact Problems I: The One-Dimensional Case”. In: *SIAM Journal on Numerical Analysis* 40 (2002), pp. 702–733.

[56] L. Paoli and M. Schatzman. “A Numerical Scheme for Impact Problems II: The Multidimensional Case”. In: *SIAM Journal on Numerical Analysis* 40 (2002), pp. 734–768.

[57] M. Jean and J. J. Moreau. “Dynamics in the Presence of Unilateral Contacts and Dry Friction: A Numerical Approach”. In: *Unilateral Problems in Structural Analysis — 2*. Springer Vienna, 1987, pp. 151–196.

[58] J.J. Moreau and P. D. Panagiotopoulos. “Unilateral Contact and Dry Friction in Finite Freedom Dynamics”. In: *Nonsmooth Mechanics and Applications*. Springer Vienna, 1988, pp. 1–82.

[59] M. Jean. “The non-smooth contact dynamics method”. In: *Computer Methods in Applied Mechanics and Engineering* 177 (1999), pp. 235–257.

[60] Q-z. Chen, V. Acary, G. Virlez, and O Brüls. "A nonsmooth generalized-α scheme for flexible multibody systems with unilateral constraints". In: *International Journal for Numerical Methods in Engineering* 96 (2013), pp. 487–511.

[61] V. Acary. "Energy conservation and dissipation properties of time-integration methods for nonsmooth elastodynamics with contact". In: *Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 96 (2016), pp. 585–603.

[62] C.W. Gear, B. Leimkuhler, and G.K. Gupta. "Automatic integration of Euler-Lagrange equations with constraints". In: *Journal of Computational and Applied Mathematics* 12-13 (1985), pp. 77–90.

[63] V. Acary, O. Bonnefon, M. Brémond, O. Huber, F. Pérignon, and S. Sinclair. *An introduction to Siconos*. Tech. rep. RT-0340. INRIA, 2019, pp.97. inria-00162911v3.

[64] T. Heyn et al. "Chrono: A Parallel Physics Library for Rigid-Body, Flexible-Body, and Fluid Dynamics". In: *Mechanical Sciences* 4 (2013), pp. 49–64.

[65] A. Tasora, S. Benatti, D. Mangoni, and R Garziera. "A geometrically exact isogeometric beam for large displacements andcontacts". In: *Computer Methods in Applied Mechanics* 358 (2020).

[66] A. Cardona, I. Klapka, and M. Géradin. "Design of a new finite element programming environment". In: *Engineering Computations* 11 (1994), pp. 365–381.

[67] A. Cardona and M. Géradin. "Numerical Integration of Second Order Differential—Algebraic Systems in Flexible Mechanism Dynamics". In: *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*. Springer Netherlands, 1994, pp. 501–529.

[68] C. Lanczos. *The variational principles of mechanics*. Dover Books on Physics (Book 4). Dover Publications; 4th Revised ed. edition, 1986.

[69] J. J. Moreau. "Quadratic programming in mechanics: Dynamics of one-sided constraints". In: *SIAM Journal on Control and Optimization* 4 (1966), pp. 153–158.

[70] J. Pérès. *Mécanique Générale*. Masson, 1953.

[71] O. Brüls, V. Acary, and A. Cardona. "On the Constraints Formulation in the Nonsmooth Generalized-α Method". In: *Advanced Topics in Nonsmooth Dynamics*. Springer International Publishing, 2018, pp. 335–374.

[72] C. Studer. *Modeling and Numerical Time Integration in Non-Smooth Dynamics*. Springer, 2009.

[73] J.J. Moreau. "Numerical aspects of the sweeping process". In: *Computer Methods in Applied Mechanics and Engineering* 177 (1999), pp. 329–349.

[74] M. Géradin and D. Rixen. "Parametrization of finite rotations in computational dynamics: A review". In: *Revue Europeenne des Elements Finis* 4 (1995), pp. 497–553.

[75] O. Brüls, A. Martin, and A. Cardona. "Two Lie Group Formulations for Dynamic Multibody Systems With Large Rotations". In: *Proceedings of the ASME Design Engineering Technical Conference* 4 (2011), pp. 85–94.

[76] V. Sonneville. "A geometric local frame approach for flexible multibody systems". PhD thesis. Université de Liège, 2015.

[77] A. Mueller and Z. Terze. "The significance of the configuration space Lie group for the constraint satisfaction in numerical time integration of multibody systems". In: *Mechanism and Machine Theory* 82 (2014), 173–202.

[78] M. Arnold, A. Cardona, and O. Brüls. "Lie group generalized-α time integration of constrained flexible multibody systems". In: *Mechanism and Machine Theory* 48 (2012), pp. 121–137.

[79] M. Arnold and O. Brüls. "Convergence of the generalized-α scheme for constrained mechanical systems". In: *Multibody System Dynamics* 18 (2007), pp. 185–202.

[80] A. Cosimo, J. Galvez, F. J. Cavalieri, A. Cardona, and O." Brüls. "A robust nonsmooth generalized-α scheme for flexible systems with impacts". In: *Multibody System Dynamics* 48 (2020), pp. 127–149.

[81] J. Galvez, F. J. Cavalieri, A. Cosimo, O. Brüls, and A. Cardona. "A nonsmooth frictional contact formulation for multibody system dynamics". In: *International Journal for Numerical Methods in Engineering* (online since April 7, 2020).

[82] R.T. Rockafellar. "Augmented Lagrangians and applications of the proximal point algorithm in convex programming." In: *Mathematics of Operations Research* 1 (1976), pp. 97–116.

[83] R. Zander, M. Forg, and H. Ulbrich. "Impacts on beam structures: Interactions of wave propagation and global dynamics". In: *IUTAM Symposium on Multiscale Problems in Multibody System Contacts* 1 (2007), pp. 327–338.

[84] Ch. Glocker. "An Introduction to Impacts". In: *Nonsmooth Mechanics of Solids*. Ed. by Jaroslav Haslinger and Georgios E Stavroulakis. Vol. 485. Springer Vienna, 2006, pp. 45–101.

[85] M. Foerg, T. Geier, L. Neumann, and H. Ulbrich. "r-Factor Strategies for the Augmented Lagrangian Approach in Multi-Body Contact Mechanics". In: *III European Conference on Computational Mechanics* (2006), p. 316.

[86] M. Balaban and A. Maraee. "Removing redundant multiplicity constraints in UML class models". In: *Software and Systems Modeling* 18 (2018), pp. 2717–2751.

[87] F. Gonzalez and J. Kövecses. "Use of penalty formulations in dynamic simulation and analysis of redundantly constrained multibody systems". In: *Multibody System Dynamics* 29(1) (2012), pp. 57–76.

[88] M. Wojtyra. "Joint reactions in rigid body mechanisms with dependent constraints". In: *Mechanism and Machine Theory* 44(12) (2009), pp. 2267–2278.

[89] M. Wojtyra. "Modeling of static friction in closed-loop kinematic chains — Uniqueness and parametric sensitivity problems". In: *Multibody System Dynamics* 39 (2017), pp. 337–361.

[90] J.C. García Orden and J. Goicolea. "Conserving properties in constrained dynamics of flexible multibody systems". In: *Multibody System Dynamics* 4 (2000), pp. 225–244.

[91] E. Lens and A. Cardona. "A nonlinear beam element formulation in the framework of an energy preserving time integration scheme for constrained multibody systems dynamics". In: *Computers and Structures* 86 (2008), pp. 47–63.

[92] S. C. Chapra and R. P. Canale. *Numerical methods for engineers*. MC Graw Hill, 2015.

[93] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[94] P. Wriggers. *Nonlinear Finite Element Methods*. Springer, 2008.

[95] F. Pfeiffer. "Mechanische Systeme mit unstetigen Übergängen". In: *Ingenieur-Archiv* 54 (1984), pp. 232–240.

[96] Ch. Glocker and F. Pfeiffer. "Multiple impacts with friction in rigid multibody systems". In: *Nonlinear Dynamics* 7 (1995), pp. 471–497.

[97] F. Pfeiffer and Ch. Glocker. "Contacts in multibody systems". In: *Journal of Applied Mathematics and Mechanics* 64 (2000), pp. 773–782.

[98] R. I. Leine, D.H. Van Campen, and Ch. Glocker. "Nonlinear dynamics and modeling of various wooden toys with impacts and friction". In: *Journal of Vibration and Control* 9 (2003), pp. 25–78.

[99] R. I. Leine and D. H. Van Campen. "Bifurcation phenomena in non-smooth dynamical systems". In: *European Journal of Mechanics - A/Solids* 25 (2006), pp. 595–616.

[100] A. Charles, F. Casenave, and Ch. Glocker. "A catching-up algorithm for multibody dynamics with impacts and dry friction". In: *Computer Methods in Applied Mechanics and Engineering* 334 (2018), pp. 208–237.

[101] Ch. Glocker and Ch. Studer. "Formulation and preparation for numerical evaluation of linear complementarity systems in dynamics". In: *Multibody System Dynamics* 13 (2005), pp. 447–463.

[102] J. Slavic and M. Boltezar. "Non-linearity and non-smoothness in multi-body dynamics: Application to woodpecker toy". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical* 220 (2006), pp. 285–296.

[103] P. Neto A.and Pimenta and P. Wriggers. "A master-surface to master-surface formulation for beam to beam contact. Part II: Frictional interaction". In: *Computer Methods in Applied Mechanics and Engineering* 319 (2017), pp. 400–429.

[104] K. E. Dufva, J. T. Sopanen, and A. M. Mikkola. "Three-Dimensional Beam Element Based on a Cross-Sectional Coordinate System Approach". In: *Nonlinear Dynamics* 43 (2006), pp. 311–327.

[105] T. Winandy and R. I. Leine. "A maximal monotone impact law for the 3-ball Newton's cradle". In: *Multibody System Dynamics* 39 (2017), pp. 79–94.

[106] I. Han and B. J. Gilmore. "Multi-body impact motion with friction-analysis, simulation, and experimental validation". In: *Journal of Mechanical Design* 115.3 (1993), pp. 412–422.