UNIVERSITÉ GRENOBLE ALPES

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : 2MGE : Matériaux, Mécanique, Génie civil, Electrochimie

Arrêté ministériel : 7 août 2006

Présentée par

Mounia Haddouni

Thèse dirigée par **Bernard Brogliato** et
Co-encadrée par **Vincent Acary**

préparée au sein de **L'INRIA**
dans **l'École Doctorale IMEP2**

Algorithmes de résolution de la dynamique du contact avec impact et frottement

Thèse soutenue publiquement le **27 Mai 2015**,
devant le jury composé de :

Mr Olivier Brüls Professeur des universités	Rapporteur
Mr David Dureisseix Professeur des universités	Rapporteur
Mr Jean-Claude Léon Professeur des universités	Examinateur
Mr Jérôme Bastien Maître de conférences	Examinateur
Mr Fabrice Thouverez Professeur des universités	Examinateur
Mr Bernard Brogliato Directeur de recherche	Directeur de thèse
Mr Vincent Acary Chargé de recherche	Co-encadrant de thèse
Mr Jean-Daniel Beley Directeur Développement logiciel	Co-encadrant de thèse

Remerciements

C'est avec beaucoup de plaisir et d'émotions que je m'adresse à toutes les personnes qui ont participé à l'aboutissement de ce projet, afin de leur montrer toute ma gratitude pour leur soutien pendant cette thèse.

Je m'adresse tout d'abord à mes directeur et co-encadrant de thèse : Bernard Brogliato et Vincent Acary. Je souhaite les remercier pour leur grande rigueur scientifique, leur franc-parler, pour toutes les discussions scientifiques enrichissantes que nous avons eues et qui m'ont permis de clarifier mes idées sur de nombreux points, ainsi que pour leurs relectures rigoureuses et instructives.

Je remercie Jean-Daniel Beley qui m'a encadré pendant cette thèse, pour sa disponibilité et nos nombreux échanges malgré ses nombreuses charges de travail. Il a été pour moi un grand soutien moral et un mentor. Je remercie également Stéphane Garreau et Ramdane Lagha pour leur aide précieuse pendant ma thèse, pour leur patience et leur accompagnement pendant ma découverte du module Rigid Body Dynamics de Ansys. Je remercie ces trois personnes pour m'avoir donné l'envie de rejoindre Ansys.

J'exprime ma gratitude à Olivier Bruls et David Dureisseix d'avoir accepté d'être rapporteurs pour ma thèse, ainsi que Jean-Claude Léon, Jérôme Bastien et Fabrice Thouverez d'avoir accepté de faire partie de mon jury de thèse.

Enfin, je remercie mes parents Driss et Zineb qui m'ont toujours présenté un soutien infaillible et inconditionnel, mon mari Jérémy pour sa patience et son soutien pendant les moments de doute, et ma soeur Imane dont l'humour a toujours été un moyen efficace pour me remonter le moral pendant les moments difficiles. Je n'oublie pas mes collègues à l'INRIA, Narendra et Olivier, je n'oublierai jamais les échanges aussi bien scientifiques que non-scientifiques, qui ont garni ces trois années de thèse avec des moments très agréables et enrichissants.

Dedicated to my father Driss, "ibbanou laazizinou"

Contents

Introduction

This dissertation is concerned with the simulation of the dynamics of multibody systems with unilateral contact and friction. The dynamics of these mechanisms is modeled with the framework of nonsmooth mechanics. The nonsmooth mechanics field has emerged thanks to the pioneering and outstanding work of Jean-Jacques Moreau [98, 82, 83, 99, 100], considered to be the father of this discipline. This approach allows for discontinuities in the time evolution of the state of the system, this state being described by the positions and velocities of the system.

This PhD is a collaboration between the Bipop research team of INRIA Grenoble and the Ansys company. This thesis is intended to improve and build new efficient and robust time integration methods for the computation of the dynamics of systems with impacts and friction. Two major time integration families are used in the field of nonsmooth mechanics: event-driven schemes and time-stepping schemes. Event-driven methods accurately detect the events and integrate the dynamics with a classical differential algebraic equations (DAE) solver during the smooth period between two events. When an event occurs, it is handled and the integration is resumed starting from the time of occurrence of this event. Time-stepping schemes do not care about the exact detection of events and provide the opportunity of handling the smooth dynamics and the nonsmooth dynamics with the same framework. The work presented in this thesis is divided into two segments.

The first segment is concerned with the improvement of the event-driven method in the industrial context of Ansys. A bibliography on the numerical methods for DAEs and ODEs enabled to select several algorithms whose relevance has been theoretically proved and whose numerical efficiency has been previously addressed only in an academic context. Unfortunately, these methods are not (enough) known to be implemented in industrial software. The chosen methods have been compared on several academic and industrial benchmarks, on the basis of several criteria including: enforcement of the constraints, time efficiency and handling stiff dynamics. The outcome of this study allows for having a clear idea on the way to select a suitable time integration method knowing the characteristics of the system. The comparison work has been presented in two conference papers [64, 65] and in an article submitted to the journal *Multibody System Dynamics* in October 2014.

In the second segment, we examine several problems that are frequent in most practical industrial applications. Namely, accumulation of impacts, friction and edge transitions due to non-perfect geometry in a CAD description. After addressing the numerical difficulty raised by the aforementioned issues, we propose an algorithm to handle them. The proposed method is a hybrid event-driven/time-stepping scheme that enables to take advantage of both time integration methods (event-driven and time-stepping schemes). The idea is to use the event-driven scheme until a switch condition is satisfied, then the integration is resumed with a time-stepping method. The switch criteria are discussed and the algorithm is tested on several examples.

Chapter 1

The coordinates systems and the formulation of the dynamics of rigid multibody systems

1.1 The coordinates systems

The first step in the simulation of multibody systems is the choice of an adequate set of coordinates that describes unequivocally the position, velocity and acceleration of each body of the system. In the field of multibody dynamics, there exist two major types of coordinates: **independent** coordinates and **dependent** coordinates. The first type is composed of the degrees of freedom of the system and its main advantage is that it enables to formulate the motion equations as a minimal set of second order ordinary differential equations (ODEs) for which a multitude of efficient and robust numerical integrators exist. However, computing such coordinates is difficult and requires some more attention as discussed in Section 1.2.2.4. The latter type (dependent coordinates) is the most commonly used in the field of multibody dynamics. The number of dependent coordinates exceeds the number of the degrees of freedom of the system, thus algebraic constraints must be written which relate these coordinates. Dependent coordinates lead to motion equations formulated as differential-algebraic equations (DAEs). There are mainly three types of dependent coordinates: **reference point** coordinates (also called *absolute* coordinates), **relative** coordinates and **natural** coordinates. An extensive study of these sets of coordinates has been performed by J. García de Jalón [52] and E.E. Soellner and C. Führer

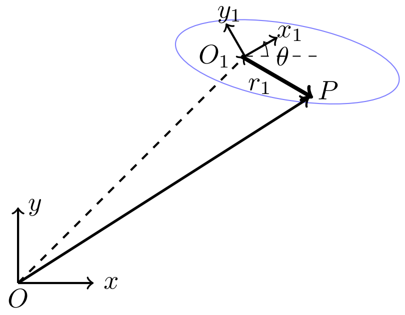


Figure 1.1: Cartesian coordinates

[42]. In this document we present the salient characteristics of each of the three types.

1.1.1 Reference point coordinates

In the framework of **reference point coordinates**, the absolute position of a given body is determined by the position of a point P of the body, and the orientation of the body with respect to a fixed reference frame $\mathbf{R}(O, x, y)$, as illustrated in Fig. 1.1. Therefore, the position of a point P in the reference frame can be written as

$$r = OP = r_{O_1} + r_1, \tag{1.1}$$

where O_1 is a *reference* point of the body (usually the center of mass is used), $r_{O_1} = OO_1$ and $r_1 = O_1P$ defines the position of point P with respect to the body fixed frame $\mathbf{R_1}(O_1, x_1, y_1)$. The position of the chosen point P can be defined using Cartesian coordinates for example. The orientation of the body is given by a rotation of R_1 expressed in R and can be described by one of the options discussed in Section 1.1.4. Let us consider a 4-bar mechanism. When the reference point coordinates are chosen, this system can be described in the reference frame $R(O, x, y)$ with the coordinates vector $q = [x_1, y_1, \varphi_1, x_2, y_2, \varphi_2, x_3, y_3, \varphi_3]$ as shown in Fig.1.2.

From the numerical standpoint, the reference point coordinates have the advantage that they lead to motion equations with sparse matrices, for which efficient algorithms make the computational effort less costly [52]. In addition, the constraint equations associated with a given kinematic joint are simpler. In turn, their main drawback is their large number.

1.1.2 Relative coordinates

In this representation, the position of a body is defined with respect to that of the neighboring one in the kinematic chain, which leads to a recursive formulation. Recursive algorithms improve the

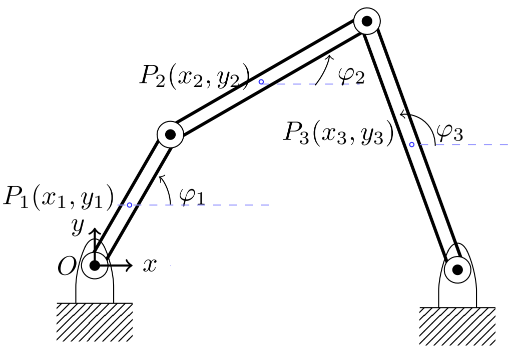


Figure 1.2: 4-bar mechanism with Cartesian coordinates

numerical efficiency of the calculation of the dynamics. Indeed, for holonomic systems, the inversion of the inertia matrix to compute the state of the system is costly. Recursive algorithms [44, 45], based on a graph topology of the multibody system, avoid this inversion. In a recursive formulation, the translation and rotation velocities of a body are related to those of its neighboring body. A graph is composed of nodes (or vertices), representing the bodies, and edges (or arcs) representing the joints or force elements connecting pairs of nodes. When a direction is given to each edge/arc, the graph is said to be *directed*. Wittenburg [132] proves that the kinematics description of a multibody system is made simple if the corresponding graph is tree-structured. In [132, Ch. 5, p. 95], Wittenburg defines a tree structure as a graph where "between any two vertices there exists a unique minimal chain of arcs and vertices connecting the two vertices". In other words, a tree is composed of the nodes of the graph, connected with edges without any closed loop. In this case, the kinematic chain is said to be open. When closed loops exist, as it is usually the case for multibody systems, they are cut at the "proper" joint or body to create an open loop. McPhee also made a pioneering work in the field of graph representation of multibody systems and automation of generation of kinematics and dynamics equations as in [95, 96], to cite but a few examples.

The choice of relative coordinates leads to a minimum number of dependent coordinates, which represents their main advantage by choosing the coordinates corresponding to the degrees of freedom allowed by the link. In addition, this number of coordinates coincides with the number of degrees of freedom of the system in the case of open kinematic chains. This fact makes these coordinates the most popular in the robotics field since robots can usually be modeled with open chains. However, the equations of motion described with relative coordinates use dense matrices whose evaluation can be numerically expensive. This fact in addition to the pre-processing (definition of the tree structure) and post-processing (definition of the absolute positions) tasks, constitute the main drawbacks of these

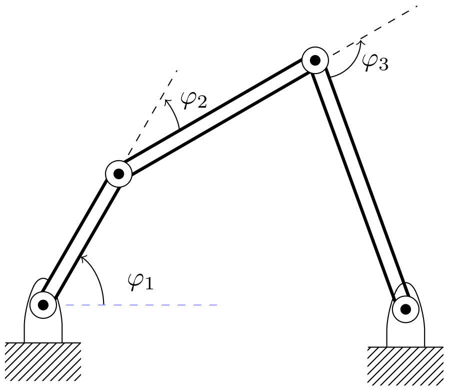


Figure 1.3: 4-bar mechanism with relative coordinates

coordinates (see [52] for more details). Note that for a closed kinematic chain, the number of relative coordinates is larger than the number of degrees of freedom. Using the relative coordinates, the 4-bar mechanism can be described using the rotation angles $q = [\varphi_1, \varphi_2, \varphi_3]$ in Fig. 1.3.

1.1.3 Natural coordinates

The **natural coordinates**, initially introduced by García de Jalón and co-workers in 1988 [51, 52], define the position of a body using the coordinates of basic points, in the frame of reference, that belong to the body. The choice of such points and vectors shall meet certain conditions recalled hereinafter as quoted from in [52, Chapter 2]:

- Each element should have at least two basic points for the motion to be defined.
- There should be a basic point in each revolute joint. This point is shared by the two elements linked at this joint.
- Each prismatic joint links two bodies, and the two basic points at one of these determine the direction of the relative motion. Although one of the basic points of the other body can be located on the segment determined by the two basic points of the first one, this is not absolutely necessary.
- In addition to the basic points that model the body, any other important point of any body can be selected as a basic point, and its coordinates would then automatically become part of the set of unknown variables.

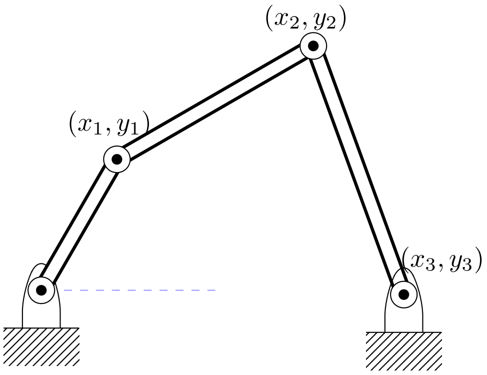


Figure 1.4: 4-bar mechanism with natural coordinates

In the 2D case, the natural coordinates can be obtained by moving the reference points to the joints. This is illustrated on the 4-bar mechanism in Fig.1.4. The basic points are taken at the revolute joints, and the system is therefore described by the set of coordinates $q = [x_1, y_1, x_2, y_2, x_3, y_3]$.

When these points are strategically located (for example in the joints), one can reduce the number of variables and even avoid writing additional constraints that describe the way the coordinates are inter-related. From the numerical viewpoint, natural coordinates offer a simple formulation and implementation. Furthermore, the constraint equations that arise from the rigid body condition or from the kinematic joints are quadratic or linear, which makes it easy for the evaluation of the Jacobian matrix.

1.1.4 Representation of rotations

The encoding of the orientation is a crucial point which deserves some explanations. In the 3D case, the finite rotations can be represented by 3 major means: **rotation matrices**, **systems of** 3 **parameters** and **unit quaternions**.

1.1.4.1 Rotation matrices

In the case of a rotation matrix representation, the columns represent the direction cosines of the moving axes with respect to the fixed frame. 3D rotation matrices represent the rotations about 3 individual axis. We consider that we work in the frame $R(O, x, y, z)$. Let us begin with a rotation of angle θ_1 about the x-axis. This rotation can be written as:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{pmatrix}. \tag{1.2}$$

Similarly, a rotation of angle θ_2 about the y-axis can be written as

$$R_y = \begin{pmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix}. \tag{1.3}$$

Likewise, a rotation of angle θ_3 about the z-axis can be written as:

$$R_z = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.4}$$

In the general case, a rotation can be considered as a sequence of 3 rotations about 3 updated axis. Therefore, a general rotation can be thought as the product of 3 rotation matrices. If we consider a rotation about the x-axis, followed by a rotation about the new y-axis and finally a rotation about the newest z-axis, the global rotation is written:

$$R = R_z R_y R_x = \begin{pmatrix} \cos\theta_2\cos\theta_3 & \sin\theta_1\sin\theta_2\cos\theta_3 - \cos\theta_1\sin\theta_3 & \cos\theta_1\sin\theta_2\cos\theta_3 + \sin\theta_1\sin\theta_3 \\ \cos\theta_2\sin\theta_3 & \sin\theta_1\sin\theta_2\sin\theta_3 + \cos\theta_1\cos\theta_3 & \cos\theta_1\sin\theta_2\sin\theta_3 - \sin\theta_1\cos\theta_3 \\ -\sin\theta_2 & \sin\theta_1\cos\theta_2 & \cos\theta_1\cos\theta_2 \end{pmatrix}. \tag{1.5}$$

The rotation matrix R satisfies the condition $R^T R = I$, where I is the identity matrix. Orthogonality and normalization conditions of the rotation matrix lead to a set of 6 equations that relate the nine elements of the matrix. This representation has the advantage that the inversion of the rotation matrix is easy to perform since the inverse is equal to the transpose. In turns, its major disadvantage is that we cannot extract a set of 3 elements that can unequivocally represent the orientation of the moving frame in relation to the fixed reference frame. In addition during the simulation, the drift of the constraints of orthogonality and normalization needs to be properly corrected.

1.1.4.2 The 3-parameter representation

The 3-parameter representation (Euler angles for example) enables to alleviate the issues that are faced when choosing the matrix representation. The most famous representations are *Euler angles*

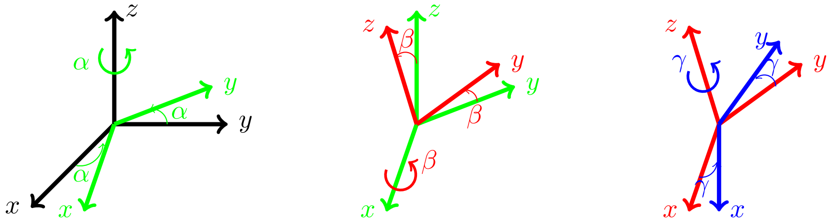


Figure 1.5: Euler angles, $z-x-z$ convention

and *Bryan (or Cardan) angles*. This representation is based on three successive rotations. If we consider the Euler angles for example, there are many conventions for the choice of the consecutive rotations. The $z-x-z$ convention is one of the most commonly used: first a rotation of angle α about the z-axis, then a rotation of angle β about the new x-axis and finally a rotation of angle γ about the newest z-axis, see Fig. 1.5.

The global rotation is represented by the composition of the 3 rotations as explained in Section 1.1.4.1. Euler and Bryan angles do not need any additional algebraic constraints, however, they have the drawback of presence of singularities. Indeed, let's denote $T(q)$ the Olinde-Rodrigues transformation matrix that links the angular velocity ω to $\dot{q} = [\dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T$:

$$\omega = T(q)\dot{q}, \tag{1.6}$$

the matrix $T(q)$ may be singular for some values of (α, β, γ). As a consequence, there may exist different values of the parameters of rotation q leading to the same configuration of the system.

1.1.4.3 Quaternions

The quaternion representation remedies the singularity issue. A quaternion is denoted by

$$q = [w,\ x,\ y,\ z]^T \ = \ w + xi + yj + zk, \tag{1.7}$$

where $(w, x, y, z) \in \mathbb{R}^4$ and i, j and k satisfy

$$\begin{cases} i^2 = j^2 = k^2 = -1 \\ ij = -ji = k \\ jk = -kj = i \\ ki = -ik = j. \end{cases} \tag{1.8}$$

The quaternion can also be denoted by (w, v) with v being the vector containing the imaginary parameters. The norm of the quaternion q is given with

$$|q| = \sqrt{w^2 + x^2 + y^2 + z^2} = \sqrt{w^2 + ||v||^2}. \tag{1.9}$$

In the case of a unit quaternion, that is $w^2 + ||v||^2 = 1$, there exists an angle θ such that:

$$\begin{cases} \cos^2\theta = w^2 \\ \sin^2\theta = ||v||^2. \end{cases} \tag{1.10}$$

Therefore, the unit quaternion can be expressed in terms of the angle θ and the unit vector $u = \frac{v}{\|v\|} = (u_1, u_2, u_3)$:

$$q = \begin{pmatrix} \cos(\theta) \\ \sin(\theta)u_1 \\ \sin(\theta)u_2 \\ \sin(\theta)u_3 \end{pmatrix}, \tag{1.11}$$

that is the rotation of angle θ about the axis defined by the unit vector u. It is worth noting that when choosing the quaternion formalism to encode the orientation of bodies, the normalization condition of the quaternion must be verified during the simulation, and corrected in case that constraint drifts. Despite the fact that they are less intuitive than the 3-parameter formalism, the advantage of quaternions over this latter representation is their mathematical simplicity and lack of singularities which make them very popular for encoding the orientation of rigid body orientation. For more information about rigid bodies orientation, we refer to [132, Chapter 2] and [34, 57, 97].

1.2 Formulation of the dynamics of rigid multibody systems

This section deals with the approaches to formulate the equations of motion of a rigid multibody system. The form of the equations of motion is directly linked to the choice of the coordinates (dependent/independent). An overview of the different formulations has been presented in [52, 17, 88], a summary of the formalisms is presented in this section. The Lagrangian formalism with and without friction will be detailed.

A very important feature characterizing a multibody system is the presence of kinematic joints and contacts. These joints result in equality constraint equations as well as Lagrange multipliers associated to the contact forces at the joints, while the contacts can be modeled as unilateral constraints.

Two major types of formalisms are utilized to compute the dynamics of constrained multibody systems: the first one keeps these multipliers in the equations of motion (Section 1.2.1), with non-minimal coordinates. The second one uses techniques of elimination of the Lagrange multipliers (Section 1.2.2) by using a minimal set of generalized coordinates.

1.2.1 Formulations solving the Lagrange multipliers

In this section, we shall derive the Lagrange equations of motion for the frictionless and for the frictional case.

1.2.1.1 The equations of dynamics using the Lagrangian formulation

Let us consider a rigid multibody system described with a set of generalized coordinates $q \in \mathbb{R}^n$. The Lagrangian of the system is $L = T(q, \dot{q}) - U(q)$, where T is the kinetic energy of the system and U is its potential energy. The kinetic energy can be expressed as $T = \frac{1}{2} v^T M(q) v$, with $v = \dot{q}$, and $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite matrix of inertia.

If in addition, the system is submitted to m holonomic and scleronomic constraints of the form $g(q) = 0$, $g : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, a new Lagrangian $\mathcal{L}$ is introduced as

$$\mathcal{L} = L(q, \dot{q}) - g(q)^T \lambda, \tag{1.12}$$

where $\lambda \in \mathbb{R}^m$ is the vector of Lagrange multipliers. Let's introduce the *action integral* $\mathcal{I}$ defined as

$$\mathcal{I} = \int_{t_1}^{t_2} \mathcal{L}(q, \dot{q}, \lambda) dt, \tag{1.13}$$

where $[t_1, \, t_2]$ is the interval of time during which the system evolves. According to Hamilton's principle, $\mathcal{I}$ has a stationary value, which can be expressed as

$$\delta \mathcal{I} = 0, \tag{1.14}$$

or in other words

$$\frac{\partial \mathcal{L}}{\partial q} - \frac{d}{dt}(\frac{\partial \mathcal{L}}{\partial \dot{q}}) = 0. \tag{1.15}$$

It follows that

$$\frac{\partial L}{\partial q} - \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) - (\frac{\partial g}{\partial q})^T \lambda = 0. \tag{1.16}$$

Finally, the Lagrange equations read

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q, v, t) + G^T(q)\lambda, \end{cases} \tag{1.17}$$

where:

- $F(q,v,t) = \frac{dM}{dt}\dot{q} - \dot{q}^T\frac{\partial M}{\partial q}\dot{q} + F_{ext}$ comprises the external applied loads and the non linear inertial terms,
- $g(q) \in \mathbb{R}^m$ is the vector of constraints imposed to the system.
- $G(q)$ is the Jacobian matrix of the constraints:

  $G(q) = \dfrac{\partial g(q)}{\partial q} = \nabla^T g(q) \in \mathbb{R}^{m \times n}$ where $\nabla g(q) = (\nabla g_1(q), ..., \nabla g_m(q)) \in \mathbb{R}^{n \times m}$ is the gradient matrix

Lagrange equations can also be derived when the system is subject to unilateral constraints of the form $g(q) \geq 0$. Many options have been proposed to modify the Lagrangian $\mathcal{L}$. These options are not addressed in the present document, however, interested readers are referred to [30, Section 3.5].

1.2.1.2 The frictionless case

In general, multibody systems are subjected to both bilateral and unilateral constraints. Both types of constraints can be taken into account in the Lagrangian formalism. Therefore, the dynamics can be described by the system of equations:

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q,v,t) + G^T(q)\lambda \\ g^\beta(q) = 0,\ \beta \in \mathbb{B} \\ 0 \leq g^\alpha(q) \perp \lambda^\alpha \geq 0,\ \alpha \in \mathbb{U}, \end{cases} \tag{1.18}$$

where:

- $\mathbb{B} \subset \mathbb{N}$ denotes the index set of bilateral constraints,
- $\mathbb{U} \subset \mathbb{N}$ denotes the set of unilateral constraints,
- the set of all constraints is denoted by $I_0 = \mathbb{B} \cup \mathbb{U}$.

The complementarity condition in (1.18) illustrates the fact that there is a reaction force only when the bodies are in contact. When the constraint $g^\alpha(q)$ is active, the reaction force has to be positive, that is $\lambda^\alpha > 0 \Rightarrow g^\alpha(q) = 0$, $\alpha \in \{\mathbb{B} \cup \mathbb{U}\}$. Furthermore, if the contact is open, that is to say $g^\alpha(q) > 0$, then the reaction force is null: $\lambda^\alpha = 0$.

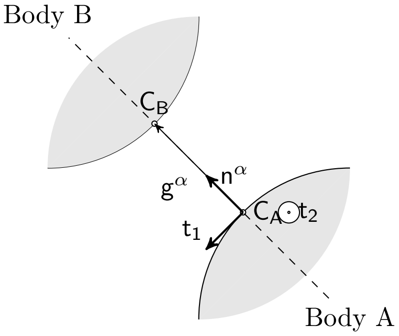


Figure 1.6: Signed distance between two bodies A and B at contact α

Fig. 1.6 illustrates one unilateral constraint g^α defined as the signed distance between the points $\mathsf{C_A}$ and $\mathsf{C_B}$. In this case, $g^\alpha = (C_B - C_A).\mathsf{n}^\alpha$, where n^α is the outward normal vector with respect to body A at C_A.

We suppose that impacts occur in infinitely short periods so that the displacements of the bodies during the collisions can be neglected, and we use the global Newton impact law with a coefficient of restitution e. For the closed contacts index set $I_1 = \{\alpha \in I_0,\, g^\alpha = 0\}$, we compute the impulse p_n and the post-impact velocity $v^+(t_n)$ by solving the Newton impact equations

$$\begin{cases} M(q(t_n))(v^+(t_n) - v^-(t_n)) = p_n \\ U_N^{\alpha,+}(t_n) = G^\alpha(q(t_n))v^+(t_n) \\ U_N^{\alpha,-}(t_n) = G^\alpha(q(t_n))v^-(t_n) \\ p_n = \displaystyle\sum_{\alpha\in I_1} G^{\alpha T}(q(t_n))P_{N,n} \\ 0 \leq U_N^{\alpha,+}(t_n) + eU_N^{\alpha,-}(t_n) \perp P_{N,n}^\alpha \geq 0, \end{cases} \tag{1.19}$$

where

$$\begin{cases} U_N^\alpha(t) = \dfrac{dg^\alpha(q(t))}{dt} = G^\alpha(q(t))\dot{q} \\ v^+(t_n) = \displaystyle\lim_{\tau\to t_n,\, \tau>t_n} v(\tau) \\ v^-(t_n) = \displaystyle\lim_{\tau\to t_n,\, \tau<t_n} v(\tau). \end{cases} \tag{1.20}$$

The complementarity condition in (1.19) describes the Signorini condition written at the velocity level and augmented by the Moreau impact law [98]. In (1.19), U_N is the vector of normal relative velocities and P_N is the vector of local impulses at the contact points. The scalar $e \in [0,\, 1]$ is the global Newton's coefficient of restitution. When the inertia matrix is invertible, problem (1.19) can be reduced to local unknowns $U_N^+(t_n)$ and $P_{N,n}$, computed by solving the following Linear Complementarity Problem

(LCP)

$$\begin{cases} U_N^+(t_n) = G(q(t_n))M^{-1}(q(t_n))G^T(q(t_n))P_{N,n} + U_N^-(t_n) \\ 0 \leq U_N^+(t_n) + eU_N^-(t_n) \perp P_{N,n} \geq 0. \end{cases} \tag{1.21}$$

The LCP matrix $G(q(t_n))M^{-1}(q(t_n))G^T(q(t_n))$ is called the Delassus matrix at $t = t_n$. Equation (1.21) describes the so-called *contact-impact LCP*.

1.2.1.3 Definition of the index of a DAE

The techniques solving the Lagrange multipliers are inextricably related to the resolution of an index-3 DAE. Many authors advise to lower index-3 DAE or to a lower index DAE or to an ODE because of the numerical problems associated to the resolution of these DAEs. García de Jalón and Bayo detailed these numerical issues, the salient conclusions are as follows. First, instability problems may be present for index-3 DAEs, particularly in presence of discontinuities of the dynamics (in the case of impacts for example). Second, the numerical schemes suffer from round-off errors which become larger as the time step size decreases. These issues will be detailed in the section dealing with the numerical schemes for the integration of the equations of motion.

In the literature [29, 42], the differential-index of a DAE is the number of differentiations of this DAE that must be performed in order to transform it into an ODE. For a constrained multibody system, when the contacts are closed for a non trivial period of time, that is $\alpha \in I_2$, $I_2(t) = \{\gamma \in I_0, (t)\, g^\alpha(q(t)) = 0,\, \dot{g}^\alpha(q(t)) = 0\}$, the dynamics of the system is described with the following index-3 DAE

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q, v, t) + G^T(q)\lambda \\ g^\alpha(q) = 0,\, \alpha \in I_2 \\ g^\alpha(q_0) = 0,\, \alpha \in I_2. \end{cases} \tag{1.22}$$

For the sake of readability, we omit the time argument t in the index sets I_0, I_2. It is well known that index-3 differential algebraic equations are difficult to numerically handle [121]. Therefore, the dynamics is usually integrated with an ODE by reducing the original index 3 of the system to 1. It amounts to solving the problem at the acceleration level by differentiating twice the constraints. Index reduction consists in differentiating w.r.t time the constraints as many times as necessary to get a set of equations that may be solved using methods for lower index problems. Hence, if the constraint

$g(\,\cdot\,)$ is differentiated once with respect to time, one obtains the following index-2 DAE

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q,v,t) + G^T(q)\lambda \\ G^{\alpha}(q)v = 0,\ \alpha \in I_2 \\ g^{\alpha}(q_0) = 0,\ \ \alpha \in I_2 \\ G^{\alpha}(q_0)v_0 = 0,\ \alpha \in I_2. \end{cases} \tag{1.23}$$

If $g(.)$ is differentiated twice, one gets the index-1 DAE

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q,v,t) + G^T(q)\lambda \\ G^{\alpha}(q)\dot{v} + \dfrac{dG^{\alpha}(q)}{dt}v = 0,\ \alpha \in I_2 \\ g^{\alpha}(q_0) = 0,\ \alpha \in I_2 \\ G^{\alpha}(q_0)v_0 = 0,\ \alpha \in I_2 \\ G^{\alpha}(q_0)\dot{v}_0 + \dfrac{dG^{\alpha}}{dt}(q_0)v_0 = 0,\ \alpha \in I_2. \end{cases} \tag{1.24}$$

The system (1.24) can be written using matrices as

$$\begin{pmatrix} M(q) & -G^T(q) \\ G(q) & 0 \end{pmatrix} \begin{pmatrix} \dot{v} \\ \lambda \end{pmatrix} = \begin{pmatrix} F(q,v,t) \\ -\dfrac{dG(q)}{dt}v \end{pmatrix}. \tag{1.25}$$

When the Delassus operator $G(q)M^{-1}(q)G^T(q)$ is invertible, the DAE (1.24) can be rewritten as an ODE of the form $\dot{y} = f(y,t)$, or more explicitly

$$\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} v \\ -M^{-1}(q)G^T(q)(G(q)M^{-1}(q)G^T(q))^{-1}(\frac{dG}{dt}v + G(q)M^{-1}(q)F(q,v,t)) + M^{-1}(q)F(q,\dot{q},t) \end{pmatrix}, \tag{1.26}$$

It can be checked that the dynamics in (1.26) renders the manifold $\{(q,\dot{q})\,|\,g(q) = 0\,,\,\dot{g}(q) = 0\}$ invariant. Under the Lipschitz continuity of the right hand side in (1.26), a unique solution (q,v) of (1.26) is guaranteed. If in addition the initial values (q_0,v_0) satisfy the position and the velocity constraints, that is to say $g(q_0) = 0$ and $G(q_0)v_0 = 0$, then the solution of (1.26) satisfies the initial problem in (1.22).

An additional differentiation of the constraints w.r.t. time leads to

$$g^{(3)}(q) = \frac{d^2G}{dt^2}(q)\dot{q} + 2\frac{dG}{dt}(q)\ddot{q} + G(q)q^{(3)} = 0. \tag{1.27}$$

We also differentiate the second equation of (1.22) and we get:

$$q^{(3)} = \frac{d(M^{-1}(q)F(q,\dot{q},t))}{dt} + \frac{d(M^{-1}(q)G^T(q))}{dt}\lambda + M^{-1}(q)G^T(q)\frac{d\lambda}{dt}. \tag{1.28}$$

By injecting the expression of $q^{(3)}$ into (1.27), it follows that:

$$\begin{aligned} G(q)M^{-1}(q)G^T(q)\dot{\lambda} = &-\frac{d^2G}{dt^2}(q)\dot{q} - 2\frac{dG}{dt}(q)\ddot{q} - G(q)\frac{d(M^{-1}(q)F(q,\dot{q},t))}{dt} \\ &- G(q)\frac{d(M^{-1}(q)G^T(q))}{dt}\lambda. \end{aligned} \tag{1.29}$$

Therefore, we can formulate an ODE for the Lagrange multipliers λ as

$$\dot{\lambda} = \tilde{f}(q,\dot{q},\ddot{q},\lambda), \tag{1.30}$$

where

$$\begin{aligned} \tilde{f}(q,\dot{q},\ddot{q},\lambda) = &\big(G(q)M^{-1}(q)G^T(q)\big)^{-1}\big(-\frac{d^2G}{dt^2}(q)\dot{q} - 2\frac{dG}{dt}(q)\ddot{q} \\ &- G(q)\frac{d(M^{-1}(q)F(q,\dot{q},t))}{dt} - G(q)\frac{d(M^{-1}(q)G^T(q))}{dt}\lambda\big). \end{aligned} \tag{1.31}$$

This implies that the differentiation index of the DAE (1.22) is indeed 3, provided that the Delassus's operator $G(q)M^{-1}(q)G^T(q)$ be invertible.

In the case of unilateral constraints, a *relative degree* can be defined as in [4] which can be seen as the counterpart of the differential index for DAEs. We can adopt the same principle as for bilateral constraints and write the complementarity relation at the velocity or the acceleration levels. Indeed, the complementarity relation $0 \leq g(q) \perp \lambda \geq 0$ can be formulated at the velocity level as

$$\begin{aligned} &\text{if } g(q) = 0, \text{then } 0 \leq \dot{g}(q) \perp \lambda \geq 0, \\ &\text{else, } \lambda = 0. \end{aligned} \tag{1.32}$$

or at the acceleration level as

$$\begin{aligned} &\text{if } g(q) = 0,\ \ \dot{g}(q) = 0,\ \ \text{then } \ 0 \leq \ddot{g}(q) \perp \lambda \geq 0. \\ &\text{else, } \lambda = 0. \end{aligned} \tag{1.33}$$

We refer to [61] for a rigorous derivation of the previous relations.

1.2.1.4 The frictional case

Friction is a fundamental mechanical phenomenon that appears at the contact points between two bodies. Friction has extensively been studied and its modeling is still the object of an active research driven by a strong need to build more robust models for engineering applications.

In this section, we shall present the Lagrange equations of motion in the frictional case. One major model is usually used for dry friction, namely the Coulomb model. This model is used in the simulations presented in this section and exposed in Section 1.2.1.4.

When using the Lagrange formalism, the equations of motion are written as

$$\begin{cases} M(q)\ddot{q} = F(q,\dot{q},t) + G_N^T(q)\lambda_N + G_T^T(q)\lambda_T \\ g^\beta(q) = 0,\ \beta \in \mathbb{B} \\ 0 \leq g^\alpha(q) \perp \lambda_N^\alpha \geq 0,\ \alpha \in \mathbb{U}, \end{cases} \tag{1.34}$$

where $G_N(q) = \frac{\partial g_N}{\partial q} \in \mathbb{R}^{m \times n}$ comprises the normal directions of the constraints while $G_T(q) = \frac{\partial \dot{g}_T}{\partial \dot{q}}(q) \in \mathbb{R}^{d\,m \times n}$ contains the tangent directions of the constraints. The value of the scalar d is 1 in the 2D case and is 2 in the 3D case.

Coulomb's model In this work, we only consider the friction model of Coulomb. It is probably the most practical and popular friction model, derived from the experimental work of Coulomb and published in 1785. In the case of a frictional contact, the reaction force λ has a normal component λ_N as well as a tangential part λ_T. Coulomb's friction states that when the tangential relative velocity $\dot{g}_T$ between the contacting bodies is nonzero, then the tangential (frictional) force is proportional to the normal force: $\parallel \lambda_T \parallel = \mu\lambda_N$ and acts in the opposite direction to the relative velocity: this case is called *sliding*. If the relative velocity is zero, then the components of the contact force satisfy the condition: $\parallel \lambda_T \parallel \leq \mu\lambda_N$, we are then in the *sticking* case. The coefficient $\mu \geq 0$ denotes the coefficient of friction and has experimentally been shown to depend on several parameters including the sliding velocity and the materials of the bodies in contact [19]. Both cases of friction are illustrated in Figures 1.7 and 1.8.

In case of planar isotropic friction, the set of possible friction forces is a disk of radius μ. In the 3D case, this set defines the so-called friction cone given by:

$$\mathbb{FC} = \{\lambda_N \mathsf{n} + \lambda_T \,|\, \lambda_N \geq 0 \,\text{and}\, \parallel \lambda_T \parallel \leq \mu|\lambda_N|\}, \tag{1.35}$$

where n is the normal vector at the contact point.

Mathematical formulation of the frictional contact The Coulomb friction model can be summarized as

$$\parallel \lambda_T \parallel \leq \mu|\lambda_N| \,\text{and} \begin{cases} \parallel \lambda_T \parallel < \mu|\lambda_N| \;\Rightarrow\; \dot{g}_T = 0 \\ \parallel \lambda_T \parallel = \mu|\lambda_N| \;\Rightarrow\; \dot{g}_T \neq 0, \text{ and } \exists\, \alpha \geq 0 \,|\, \lambda_T = -\alpha\dot{g}_T. \end{cases} \tag{1.36}$$

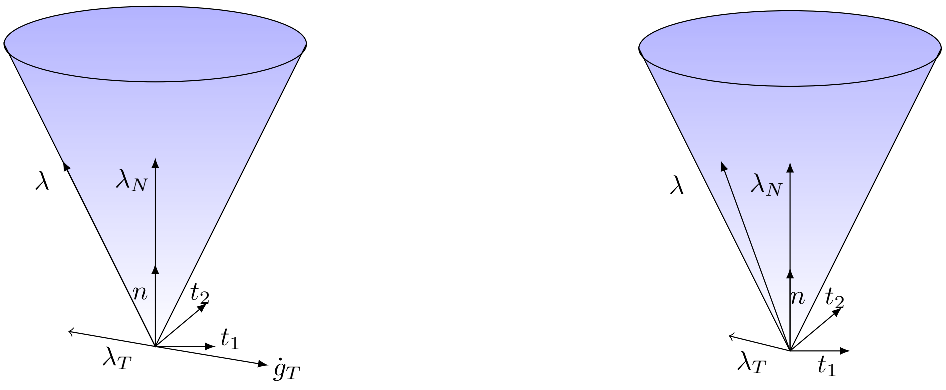


Figure 1.7: Sliding case ($\dot{g}_T \neq 0$)

Figure 1.8: Sticking case ($\dot{g}_T = 0$)

Figure 1.9: Coulomb's friction cases

This law can be formulated as an inclusion written as

$$-\dot{g}_T \in N_{C_{(\mu\lambda_N)}}(\lambda_T),\, C(\mu\lambda_N) \subset \mathbb{R}^2, \tag{1.37}$$

where $C(\mu\lambda_N)$ denotes the section of the friction cone and $N_{C_{(\mu\lambda_N)}(\lambda_T)}$ is the normal cone at λ_T. For an anisotropic friction, this section is elliptical. This work is limited to isotropic friction for which the cone's section is a disk of radius $\mu\lambda_N$ defined with

$$C(\mu\lambda_N) = \{\lambda_T \mid \mu|\lambda_N| - \parallel \lambda_T \parallel \geq 0\}. \tag{1.38}$$

Coulomb's friction model is a set valued force law and it is discontinuous at the transitions between sliding and sticking phases. From the mathematical point of view, in the 3D case, this law involves non-linearities and can be formulated as Nonlinear Complementarity Problem (NCP). However, NCPs solvers are not as developed as LCPs solvers, and available methods can be very cumbersome from the computation time standpoint. Therefore, the friction cone is usually approximated with a polyhedral cone in order to formulate the friction problem as LCP. Klarbring [84] was probably the first to propose a polyhedral approximation to the friction cone to solve the quasi static problem. The authors in [116] used the approximation in the dynamical case and proposed a more compact formulation leading to a faster solution. However, one should be aware that a good approximation (increasing the number of edges of the polyhedral cone) is computationally expensive and will impact the numerical efficiency of the LCP solvers, on the other hand, a poor approximation is to the detriment of accuracy. In addition, Glocker [58] pointed out that the polyhedral approximation does not allow to consider constraints dependency in a multibody system as it is usually the case, which leads to a lack in the physical properties in the model; he also mentioned that the pyramidal cone can affect the collinearity between the friction force λ_T and the relative tangent velocity $\dot{g}_T$, this can be avoided by increasing the number of vertices of the polyhedral, but it will be to the detriment of the computation time. A quantitative evaluation of the efficiency may be found in [114] where several solvers of the frictional contact problem in the 3D case are compared on various problems. Another drawback related to the polyhedral approximation is the problem of cycling due to the degeneracy, this issue is discussed in [4, Chapter 13].

For the LCP formulations of the friction problem, review papers that may be consulted are [4, 49, 58, 62, 111, 112]. Most of the aforementioned references formulate the problem with constraints on the acceleration level. The major problem of using this force-acceleration model is that the problem does not necessarily have a solution, and when this solution exists, it may not be unique for some configurations. This problem is known as the **Painlevé paradox**. Painlevé showed on a simple model that the contact LCP may not have a solution for some configurations and velocities. In [124], Stewart explains that in configurations of inconsistency, the problem does not have a continuous force solution, but proved the existence of an impulsive force solution [123]. The proof of existence of a solution is also valid under restrictive conditions. In the work of Cadoux et al. [6, 5], it is shown that even with an impulsive motion, it may happen that the system has no solution. In [21] for example, upper bounds are defined for the friction coefficients which guarantee that the Painlevé paradox never occurs.

1.2.2 Formulations eliminating the Lagrange multipliers

In this section we shall discuss the formulations that eliminate the Lagrange multipliers from the equations of motion. Probably the most classical technique is to transform the index-3 DAE in (1.22) to an ODE as in (1.26). In the following, other techniques are discussed.

1.2.2.1 Udwadia and Kalaba formulation

Based on the Moore-Penrose generalized inverses, **Udwadia and Kalaba** derived a more general form of the null space formulation. Let's introduce the "free-constraints" accelerations $\ddot{q}_f = M^{-1}(q)F(q,v,t)$. The authors consider a more general form of the constraints, at the acceleration level, in the form: $G(q,\dot{q},t)\ddot{q} = b(q,\dot{q},t)$, with $b \in \mathbb{R}^m$. The inertia matrix $M(q)$ is assumed to be positive definite, as it is usually the case. Two matrices $M^{-1/2}$ and K are defined as: $M(q)^{-1} = M(q)^{-1/2}M(q)^{-1/2}$ and $K(q,\dot{q},t) = M(q)^{-1/2}\big(G(q,v,t)M(q)^{-1/2}\big)^{+}$. The superscript

$+$ denotes the Moore-Penrose generalized inverse of $G(q, \dot{q}, t)M(q)^{-1/2}$. Finally, the acceleration of the system is given by

$$\ddot{q} = \ddot{q}_f + M^{-1/2}(q)K(q, v, t)(b(q, v, t) - G(q, v, t)\ddot{q}_f). \tag{1.39}$$

For more details about this formulation, we refer to [128, 129, 16, 88].

1.2.2.2 Maggi's/Kane's formulation

While the previous methods consider the acceleration constraints, ***Maggi's/Kane's formulation*** enforces the velocity constraints. In 1896 and then in 1901, Maggi [88, 16, 22, 53] presented his formulation of the dynamics for the more general case of systems with nonholonomic constraints, based on the D'Alembert's principle. Later, in the second half of the XX^{th} century, Kane separately derived a formulation of the dynamics which is very close to that of Maggi. The core idea of both methods is to extract a minimal set of independent coordinates, and to formulate the dynamics using only these coordinates. This minimal set of coordinates can for example be established with a graph analysis of the multibody system. When the system has a tree structure, the independent coordinates correspond to the joint coordinates. When the system contains closed loops, the graph representing the system is transformed into a tree by "cutting" some appropriate joints, a minimal set of coordinates is then extracted. For a deep explanation about tree representation of multibody systems with closed loops, we refer to [132, Section. 5.6].

Since both methods are equivalent [22], in the following we only present Maggi's method, derived from the D'Alembert's principle that states that the virtual work of the difference between applied forces and inertial forces must vanish for any virtual displacement. This can be formulated as

$$\delta W = \left(F(q, \dot{q}, t) - M(q)\ddot{q}\right)^T \delta q, \tag{1.40}$$

where δq is the virtual displacement. In 1896 and then in 1901, Maggi presented his formulation of the dynamics for the more general case of systems with m nonholonomic constraints based on the D'Alembert's principle. Many authors [88, 16, 22, 53] have studied this method, in the following we recall the salient steps that lead to it. Let us assume that the nonholonomic constraints take the form

$$G(q, t)\dot{q} + g_t(q, t) = 0. \tag{1.41}$$

$G \in \mathbb{R}^{m \times n}$ is not necessarily the jacobian of some set of constraints, it is assumed to be of full row rank. Maggi defines a set of $n - m$ independent *kinematic variables* e, not integrable in the general case, as

$$e = \tilde{B}(q, t)\dot{q} + \tilde{b}(q, t),\ \tilde{B} \in \mathbb{R}^{(n-m) \times n}. \tag{1.42}$$

The block matrix $(G \quad \tilde{B})^T$ is assumed to be invertible, its inverse is denoted $(\tilde{G} \quad B)$. Therefore, the velocity vector can be expressed as a function of the kinematic variables as

$$\dot{q} = B(q,t)e + b(q,t),\ B \in \mathbb{R}^{n\times(n-m)}, \tag{1.43}$$

with $b(q,t) = -(\tilde{G}c + B\tilde{b})$, with $\tilde{G} \in \mathbb{R}^{n\times m}$. From equations (1.41), (1.42) and (1.43), we can write:

$$\begin{cases} G\tilde{G} = I \in \mathbb{R}^{m\times m} \\ \tilde{B}B = I \in \mathbb{R}^{(n-m)\times(n-m)} \\ \tilde{B}\tilde{G} = 0 \in \mathbb{R}^{(n-m)\times m} \\ GB = 0 \in \mathbb{R}^{m\times(n-m)}, \end{cases} \tag{1.44}$$

where I denotes the identity matrix. From the fourth equation in (1.44), we can see that $B(q)$ spans the null space of $G(q)$. By taking the derivative of equation (1.43), we get:

$$\ddot{q} = B(q,t)\dot{e} + h(q,\dot{q},t), \tag{1.45}$$

where $h(q,\dot{q},t) = \dot{B}(q,t)e + \dot{b}(q,t) = \dot{B}(q,t)\tilde{B}(q,t)\dot{q} + \dot{B}(q,t)\tilde{b}(q,t) + \dot{b}$. Let us consider the Lagrange equations of motion;

$$M(q)\ddot{q} = F(q,\dot{q},t) + G^T(q,t)\lambda. \tag{1.46}$$

By injecting the expression of $\ddot{q}$ in equation (1.45) into equation (1.46), it follows that:

$$M(q)B(q,t)\dot{e} = F(q,\dot{q},t) - M(q)h(q,\dot{q},t) + G^T(q,t)\lambda \tag{1.47}$$

To reduce the problem to the minimal coordinates set, that is the set of the kinematic variables, we multiply by the transpose of the matrix B. One obtains

$$B^T(q,t)M(q)B(q,t)\dot{e} = B^T(q,t)F^\star(q,\dot{q},t) + B^T(q,t)G^T(q,t)\lambda, \tag{1.48}$$

where $F^\star(q,\dot{q},t) = F(q,\dot{q},t) - M(q)h(q,\dot{q},t)$. Since $B^TG^T = 0$, the problem is reduced to

$$B^T(q,t)M(q)B(q,t)\dot{e} = B^T(q,t)F^\star(q,\dot{q},t). \tag{1.49}$$

Equation (1.49) defines the *Maggi's formulation* of the equations of motion for a system with nonholonomic constraints. It is clear that the main challenge of this formulation is the choice of the kinematic variables e and the evaluation of matrix B. The matrix B can be computed by means of Gauss-Jordan elimination of matrix G.

In the literature, we can find several works on the equivalence between Kane's formalism and other well-known formulations of the equations of the dynamics. In [22], Borri, Bottasso and Mantegazza

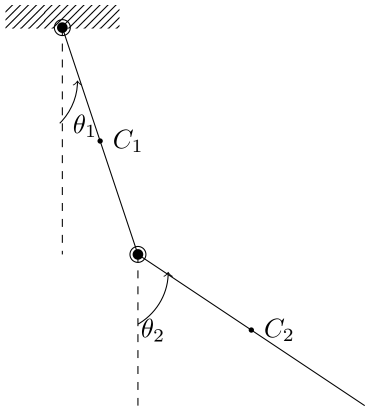


Figure 1.10: Double pendulum

have established the equivalence between Kane's equations and Maggi's equations based on the fact that the essence of both methods is to find a minimal set of variables to describe the dynamics.

Let's illustrate Maggi-Kane's equations on the double pendulum depicted in Fig.1.10. The bars are of masses m_1 and m_2 and of lengths l_1 and l_2 respectively. The vector of generalized coordinates is composed of the cartesian coordinates of the centers of mass C_1 and C_2, therefore $q = [x_{C_1}, y_{C_1}, x_{C_2}, y_{C_2}]^T$.

Intuitively, the vector of kinematic variables is composed of the time derivatives of the angles: $e = [\dot{\theta}_1, \dot{\theta}_2]^T$. The mechanism studied here being simple, we can establish by inspection the matrix B by expressing $\dot{q}$ as a function of e. We have

$$\begin{cases} x_{C_1} = \frac{1}{2} l_1 \sin(\theta_1) \\ y_{C_1} = \frac{1}{2} l_1 \cos(\theta 1) \\ x_{C_2} = l_1 \sin(\theta_1) + \frac{1}{2} l_2 \sin(\theta_2) \\ y_{C_2} = l_1 \cos(\theta_1) + \frac{1}{2} l_2 \cos(\theta_2), \end{cases} \tag{1.50}$$

therefore

$$\begin{cases} \dot{x}_{C_1} = \frac{1}{2} l_1 \dot{\theta}_1 \cos(\theta_1) \\ \dot{y}_{C_1} = -\frac{1}{2} l_1 \dot{\theta}_1 \sin(\theta_1) \\ \dot{x}_{C_2} = l_1 \dot{\theta}_1 \cos(\theta_1) + \frac{1}{2} l_2 \dot{\theta}_2 \cos(\theta_2) \\ \dot{y}_{C_2} = -l_1 \dot{\theta}_1 \sin(\theta_1) - \frac{1}{2} l_2 \dot{\theta}_2 \sin(\theta_2). \end{cases} \tag{1.51}$$

It follows that

$$B = \begin{pmatrix} \frac{1}{2}l_1\cos(\theta_1) & 0 \\ -\frac{1}{2}l_1\sin(\theta_1) & 0 \\ l_1\cos(\theta_1) & \frac{1}{2}l_2\cos(\theta_2) \\ -l_1\sin(\theta_1) & -\frac{1}{2}l_2\sin(\theta_2) \end{pmatrix}. \tag{1.52}$$

The mass matrix being

$$M = \begin{pmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 \\ 0 & 0 & m_2 & 0 \\ 0 & 0 & 0 & m_2 \end{pmatrix}, \tag{1.53}$$

we have

$$B^T M B = \begin{pmatrix} (\frac{m_1}{4} + m_2)l_1^2 & m_2\frac{l_1 l_2}{2}\cos(\theta_2 - \theta_1) \\ m_2\frac{l_1 l_2}{2}\cos(\theta_2 - \theta_1) & m_2\frac{l_2^2}{4} \end{pmatrix}. \tag{1.54}$$

On the other hand

$$h = \dot{B}e = \begin{pmatrix} -\frac{l_1}{2}\dot{\theta}_1^2\sin(\theta_1) \\ -\frac{l_1}{2}\dot{\theta}_1^2\cos(\theta_1) \\ -l_1\dot{\theta}_1^2\sin(\theta_1) - \frac{l_2}{2}\dot{\theta}_2^2\sin(\theta_2) \\ -l_1\dot{\theta}_1^2\cos(\theta_1) - \frac{l_2}{2}\dot{\theta}_2^2\cos(\theta_2) \end{pmatrix}, \tag{1.55}$$

The vector of external forces is

$$F = \begin{pmatrix} 0 \\ -m_1\mathsf{g} \\ 0 \\ -m_2\mathsf{g} \end{pmatrix}, \tag{1.56}$$

where g denotes the gravity. Finally, after some calculations that we do not show here, we obtain Maggi's formulation of the dynamics of this double pendulum:

$$\begin{pmatrix} (\frac{m_1}{4} + m_2)l_1^2 & m_2\frac{l_1 l_2}{2}\cos(\theta_2 - \theta_1) \\ m_2\frac{l_1 l_2}{2}\cos(\theta_2 - \theta_1) & m_2\frac{l_2^2}{4} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = \begin{pmatrix} (\frac{m_1}{2} + m_2)l_1\mathsf{g}\sin(\theta_1) - m_2\frac{l_1 l_2}{2}\dot{\theta}_2^2\sin(\theta_2 - \theta_1) \\ m_2\frac{l_2}{2}\mathsf{g}\sin(\theta_2) + m_2\frac{l_1 l_2}{2}\dot{\theta}_1^2\sin(\theta_2 - \theta_1) \end{pmatrix}. \tag{1.57}$$

1.2.2.3 The null space formulation

The null space formulation uses the null space matrix $R(q) \in \mathbb{R}^{(n-m)\times m}$ of the Jacobian matrix $G(q) \in \mathbb{R}^{m\times n}$, that is $R^T(q)G^T(q) = 0$. The number of rows of $R(q)$ is equal to the number of degrees of freedom of the system. The idea behind this formulation is to write the equations of motion in a way that enables to distinguish the kinematic equations from the dynamics equations, as in equation

(1.58). This is performed by projecting the equations of the dynamics onto the tangent plane of the constraints manifold. When the inertia matrix $M(q)$ is invertible and the Jacobian matrix $G(q)$ has full rank, the matrix $\begin{pmatrix} R^T(q)M(q) \\ G(q) \end{pmatrix}$ is square and invertible, and the initial problem formulated in the index-1 DAE (1.24) is transformed into an ordinary differential equation (ODE), as follows

$$\begin{pmatrix} R^T(q)M(q) \\ G(q) \end{pmatrix} \ddot{q} = \begin{pmatrix} R^T(q)F(q,\dot{q},t) \\ \frac{dG}{dt}\dot{q} \end{pmatrix}, \tag{1.58}$$

which has a unique solution [54]. The computation steps that lead to this equation are detailed in [16, 52].

It is clear that the evaluation of the null space $R(q)$ dominates the computational cost of this method. Several techniques can be used to evaluate the matrix $R(q)$. To cite but a few: the Singular Value decomposition, the QR Decomposition and the Gaussian triangularization. It is worth noting that for most applications, the evaluation of the constraint forces is required. Since they are eliminated by this technique, an additional post-processing task will be necessary, which constitutes an additional numerical effort.

1.2.2.4 The coordinate partitioning method

An alternative solution to the previous techniques is the ***coordinate partitioning method***. Initially proposed by Wehage and Haug [130, 131], its basic idea is to consider a partition of the generalized coordinates vector q in two sets of dependent and independent coordinates in the form: $q = [q_d, q_{\mathsf{ind}}]^T$, where $q_{\mathsf{ind}} \in \mathbb{R}^{\mathsf{ndof}}$ represents the set of independent coordinates while $q_d \in \mathbb{R}^{n-\mathsf{ndof}}$ is the set of dependent coordinates, with ndof being the number of degrees of freedom of the system. This partitioning is performed by applying a Gauss-Jordan reduction of the Jacobian matrix $G(q)$, with column pivoting or by using SVD or LU factorization. LU factorization leads to a set of independent coordinates as part of the generalized coordinates, while the SVD leads to a set expressed as a linear combination of the generalized coordinates. The partition requires the matrix $G(q)$ to be of full row rank. The set of independent coordinates q_{ind} locally parameterizes the constraints manifold, and the partitioning requires the Jacobian of the constraints w.r.t q_d to be non singular, that is $\mathsf{det}G_d(q) = \mathsf{det}(\frac{\partial g}{\partial q_d}(q)) \neq 0$. This property added to the implicit function theorem enables to express q_d as a function of q_{ind} using a function u, as many times differentiable as g, as: $q_d = u(q_{\mathsf{ind}})$.

While in the null space method, the entire set of coordinates q is computed during the simulation, in the coordinate partitioning formalism, only the independent velocities are numerically integrated.

The dependent velocities are deduced by solving the linear system

$$\begin{pmatrix} G_d(q) & G_{\mathsf{ind}}(q) \end{pmatrix} \begin{pmatrix} \dot{q}_d \\ \dot{q}_{\mathsf{ind}} \end{pmatrix} = -g_t, \tag{1.59}$$

The equations of motion are then partitioned as

$$\begin{cases} M^{ii}\ddot{q}_{\mathsf{ind}} + M^{id}\ddot{q}_d + G_{\mathsf{ind}}^T\lambda = F_{\mathsf{ind}} \\ M^{di}\ddot{q}_{\mathsf{ind}} + M^{dd}\ddot{q}_d + G_d^T\lambda = F_d \\ G_d\ddot{q}_d + G_{\mathsf{ind}}\ddot{q}_{\mathsf{ind}} = \dfrac{dG}{dt}\dot{q}, \end{cases} \tag{1.60}$$

where M^{ii}, M^{id}, M^{di}, M^{dd}, F_{ind}, F_d are "appropriate" submatrices and subvectors of M and F respectively [20, 102, 47]. After eliminating $\ddot{q}_d$ and λ, one obtains

$$\hat{M}\ddot{q}_{\mathsf{ind}} = \hat{F}, \tag{1.61}$$

where

$$\begin{cases} \hat{M} = M^{ii} - M^{id}G_d^{-1}G_{\mathsf{ind}} - G_{\mathsf{ind}}^T\big(G_d^{-1}\big)^T\big(M^{di} - M^{dd}G_d^{-1}G_{\mathsf{ind}}\big) \\ \hat{F} = F_{\mathsf{ind}} - M^{id}G_d^{-1}\dfrac{dG}{dt}\dot{q} - G_{\mathsf{ind}}^T\big(G_d^{-1}\big)^T\big(F_d - M^{dd}G_d^{-1}\dfrac{dG}{dt}\dot{q}\big) \end{cases} \tag{1.62}$$

For more details about the computations leading to equation (1.61), we refer to [93] for example. From a numerical point of view, some crucial points should be mentioned. First, the choice of q_{ind} is not global, that is, this set should be updated during the simulation. This update increases the computational effort and propagates integration errors. In addition, a bad choice of this set may result in algorithms with ill-conditioned systems and/or very numerically demanding. Several algorithms have been proposed that optimize the partitioning technique and propose solutions to choose the best set of independent coordinates, we refer to [102, 20] for example.

As explained in [52], in general, no set of independent coordinates is able to describe neither the whole motion of the system nor all the possible positions. Therefore the set of independent coordinates must be updated during the time integration of the dynamics equations. In [52], the authors propose several techniques to perform this update, which is the main challenge of this approach.

McClamroch and Wang [94] introduced a different method of partitioning based on a transformation of the coordinates. Assuming that the Jacobian matrix $G(q)$ is of full row rank, and using the implicit function theorem, the authors define a function $\Omega : \mathbb{R}^{n-m} \longrightarrow \mathbb{R}^m$, such that : $g(\Omega(q_2), q_2) = 0$, $\forall q_2 \in \mathbb{R}^{n-m}$. The vector of generalized coordinates q is partitioned as $q = [q_1,\, q_2]^T$ where $q_1 \in \mathbb{R}^m$ and $q_2 \in \mathbb{R}^{n-m}$, and a transformation $X : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is introduced as follows

$$x = X(q) = \begin{pmatrix} q_1 - \Omega(q_2) \\ q_2 \end{pmatrix}, \tag{1.63}$$

X is differentiable and its differentiable inverse is

$$q = Q(x) = \begin{pmatrix} x_1 + \Omega(x_2) \\ x_2 \end{pmatrix}, \tag{1.64}$$

with $x = [x_1, x_2]^T$, $x_1 \in \mathbb{R}^m$, $x_2 \in \mathbb{R}^{n-m}$. The jacobian T of Q is introduced

$$T(x) = \frac{\partial Q}{\partial x} = \begin{pmatrix} I_m & \frac{\partial \Omega}{\partial x_2} \\ 0 & I_{n-m} \end{pmatrix}. \tag{1.65}$$

Eq (1.18) with only bilateral constraints becomes

$$T^T(x)M(Q(x))T(x)\ddot{x} + T^T(x)\big(F(Q(x), T(x)\dot{x}) + M(Q(x))\dot{T}(x)\dot{x}\big) = T^T(x)F_c, \tag{1.66}$$

with $F_c = G^T(q)\lambda$. This equation is then reduced to

$$\begin{cases} E_1\tilde{M}(x_2)E_2^T\ddot{x}_2 + E_1\tilde{F}(x_2, \dot{x}_2) = E_1T^T(x_2)F_c \\ E_2\tilde{M}(x_2)E_2^T\ddot{x}_2 + E_2\tilde{F}(x_2, \dot{x}_2) = 0, \end{cases} \tag{1.67}$$

where

$$\begin{cases} \tilde{M}(x) = T^T(x)M(Q(x))T(x) \\ \tilde{F}(x, \dot{x}) = T^T(x)\big(F(Q(x), T(x)\dot{x}) + M(Q(x))\dot{T}(x)\dot{x}\big) \\ I_n = [E_1^T,\, E_2^T],\ E_1 \in \mathbb{R}^{m\times n},\, E_2 \in \mathbb{R}^{(n-m)\times n} \\ E_2T^T(x_2)F_c = 0. \end{cases} \tag{1.68}$$

The vector $\tilde{F}(x_2, \dot{x}_2)$ denotes $F(x, \dot{x})$ evaluated at $x = [0, x_2]$ and $\dot{x} = [0, \dot{x}_2]$. The ODE presented in the second equation of system (1.67) describes the motion in the constraint manifold. A big discrepancy with respect to other methods is that the dynamics in (1.66) or (1.67) is a Lagrange dynamics because the transformation is based on the generalized coordinates transformation (1.64) and generalized diffeomorphic coordinates transformations are known to preserve the Lagrangian structure.

1.3 Conclusion

In this chapter, we discussed several coordinates systems and several formulations of the equations of motion of multibody systems. Relative coordinates are probably the one that are the most frequently used, they enable to write the equations of motion with the least number of bilateral constraints. As regards the formulation of the dynamics, some formulations aim at eliminating the Lagrange multipliers from the equations of the motion by means including :

- using some particular set of coordinates as in Maggi-Kane's formulaion,
- using the null space matrix of the jacobian matrix as in the null space formulation,
- partitionning the coordinates into independent and dependent coordinates, and writing the dynamics as an ODE with the independent coordinates,
- using the Moore-Penrose inverse as in the Udwadia and Kalaba formulation.

The Lagrange equations solve the Lagrange multipliers, they have been exposed for both the frictionless and the frictional case. The obtained equations may be formulated as an index-3, index-2 or index-1 DAE together with an impact law. For instance, the Newton's impact law may be used. Let us mention that concerning the work presented in this document, the dynamics of the academic tests is defined with Lagrange equations, while the Maggi-Kane formulation is used for the industrial benchmarks from the Ansys software.

Chapter 2

Numerical integration of the equations of motion

The simulation of the dynamics of multibody systems requires a great care. Indeed, the bodies are linked with kinematic joints, modeled with bilateral constraints that must be enforced with selected integration scheme. In addition, bodies may be subjected to impacts leading to jumps in the state. Classical DAE schemes are not able to handle such a nonsmooth dynamics. In this chapter, we discuss the integration methods that are commonly used for the computation of the dynamics of constrained multibody systems.

2.1 The global integration process

There are two major techniques for the simulation of the dynamics of multibody systems subjected to non-smooth events such as impacts and transitions from sticking to sliding: ***event-detecting time–stepping schemes (a.k.a. event–driven schemes)*** and ***event-capturing time–stepping schemes (also shortly called time–stepping schemes)***. In this section, we shall discuss both families, their advantages and their drawbacks.

2.1.1 Event-driven schemes

The event-driven schemes are based on the separation between events or nonsmooth dynamics at discrete times, and the smooth dynamics [4, 1, 49]. The detection of the occurrence of an event (an impact for example) is a major step. Indeed, the time of occurrence of the events must be defined very accurately not to break down the order of the consistency of the method as discussed in Section 2.1.1.2. Between two successive events, the dynamics is described with a smooth formulation (DAE, ODE) and integrated with any classical numerical scheme [66, 67]. When an event occurs, the dynamics is updated after computing the new initial conditions at the **switching time** (time of occurrence of an event) with some suitable algorithm. These initial conditions are used to advance the time integration up to the next event. In this section, we introduce the definition of the index sets that describe the status of the contact, and then we present the general algorithm of the event-driven scheme.

2.1.1.1 The abstract algorithm of an event-driven scheme

In the event-driven strategy, three index sets are generally introduced in order to characterize the state of the contacts:

- ◦ the index set I_0 of all possible constraints to which the system is submitted: $I_0 = \mathbb{B} \cup \mathbb{U}$,
- ◦ the index set I_1 of contacts active in position: $I_1 = \{\alpha \in I_0 \mid g^\alpha(q) = 0\}$,
- ◦ the index set I_2 of contacts that are active in position and velocity: $I_2 = \{\alpha \in I_1 \mid \dot{g}^\alpha(q) = 0\}$.

In the frictionless case, events are defined by a change in I_1 or in I_2. When an event occurs, it is handled and the dynamics is initialized from its time of occurrence. The event-driven scheme used in this work is illustrated in Fig.2.1.

Some steps in the algorithm of Fig.2.1 deserve further consideration:

- *Compute the index sets*: some numerical thresholds τ_0 and τ_1 are defined that are required to evaluate equalities and inequalities of floating point numbers when considering the index sets I_1 and I_2. Therefore, we write: $I_1 = \{\alpha \in I_0 \mid g^\alpha(q_m) \leq \tau_0\}$ and $I_2 = \{\alpha \in I_1 \mid \dot{g}^\alpha(q_m) \leq \tau_1\}$, where q_m denotes the position evaluated at time t_m.

- *Project on active position and velocity constraints*: for bilateral constraints $g^\alpha(q)$, $\alpha \in \mathbb{B}$, if during the simulation we have $|g^\alpha(q)| > \tau_0$ or $|\dot{g}^\alpha(q)| > \tau_1$, then we project the constraints on the position and velocity manifolds (see Section 2.1.1.3). For unilateral constraints, we project if there is some numerical interpenetration, that is to say $g^\alpha(q) \leq 0$, $\alpha \in \mathbb{U}$.

- *Handle the event*: there are many types of events, including:

  1. nonsmooth change in the loading conditions.
  2. impacts (activation of constraints), in this case we solve the impact LCP (1.21) at the instant $t^\star$ of the event.

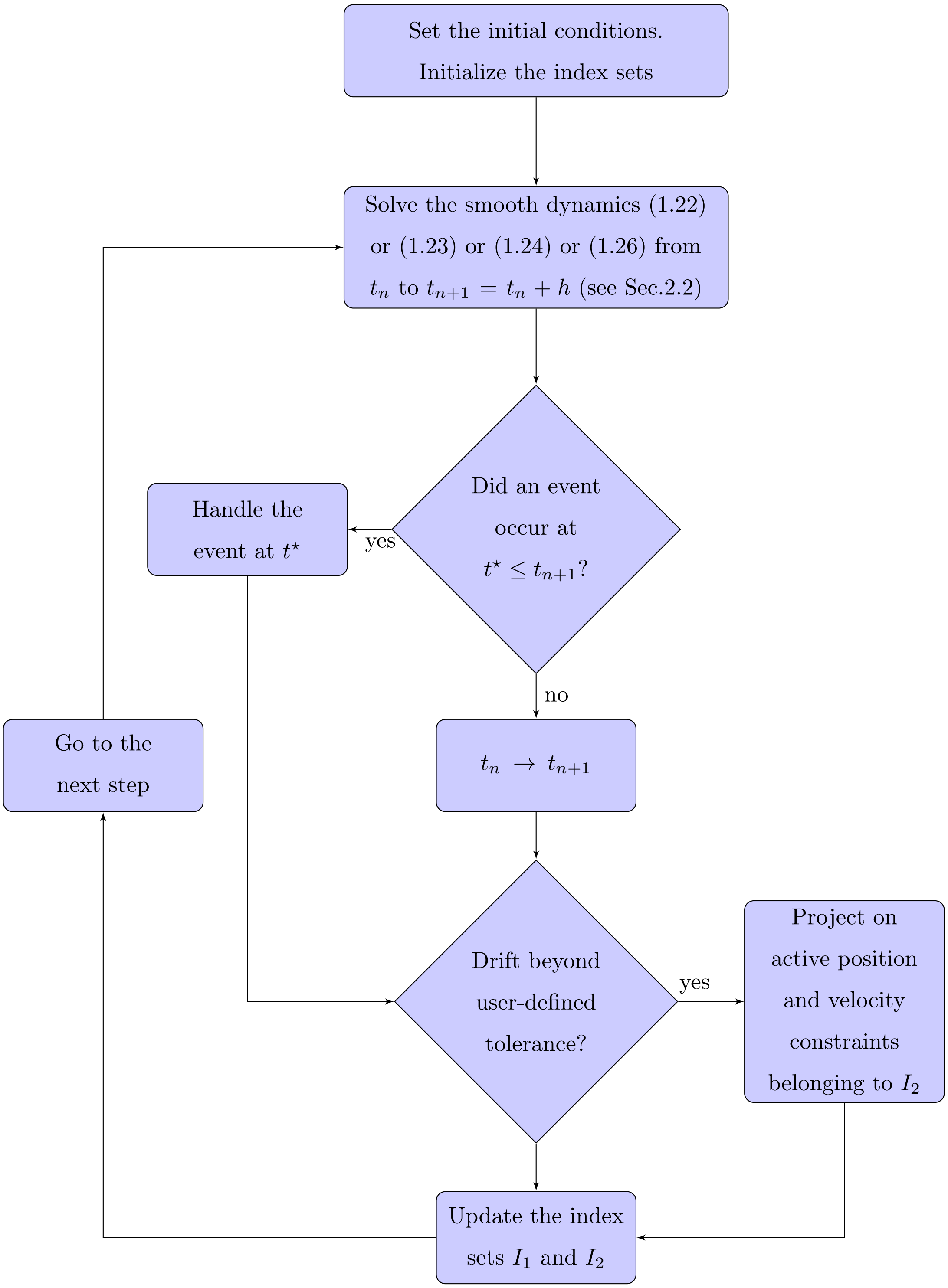

Set the initial conditions.
Initialize the index sets
Solve the smooth dynamics (1.22) or (1.23) or (1.24) or (1.26) from t_n to $t_{n+1} = t_n + h$ (see Sec.2.2)
Did an event occur at $t^\star \leq t_{n+1}$?
yes
Handle the event at $t^\star$
no
Go to the next step
$t_n \rightarrow t_{n+1}$
Drift beyond user-defined tolerance?
yes
Project on active position and velocity constraints belonging to I_2
Update the index sets I_1 and I_2

Figure 2.1: The event-driven scheme. Synopsis of a one-step integration.

3. change in the normal vector that leads to a jump in the constraints and contact forces. This point is discussed in chapter 4.
4. change in the sliding status in case of friction (sticking, sliding).

When friction is taken into account, then additional index sets are written which define the status of the frictional contacts. These additional sets are:

- $I_{St} = \{i \in I_2 \mid \parallel \dot{g_T}_i(q) \parallel = \parallel (G_T(q)\dot{q})_i \parallel = 0\}$: the set of sticking contacts.
- $I_{Sl} = \{i \in I_2 \mid \parallel \dot{g_T}_i(q) \parallel = \parallel (G_T(q)\dot{q})_i \parallel \neq 0\}$: the set of sliding contacts.

A frictional contact α can go through the following modes:

- Sliding: $\parallel \dot{g}_T^\alpha \parallel \geq 0$, and $\parallel \lambda_T^\alpha \parallel = \mu^\alpha |\lambda_N^\alpha|$.
- Sliding to sticking: $\parallel \dot{g}_T^\alpha \parallel$ moves from a positive value to 0, and $\parallel \lambda_T^\alpha \parallel \leq \mu^\alpha |\lambda_N^\alpha|$.
- Sticking: $\parallel \dot{g}_T^\alpha \parallel = 0$, and $\parallel \lambda_T^\alpha \parallel < \mu^\alpha |\lambda_N^\alpha|$.
- Sticking to sliding: $\parallel \dot{g}_T^\alpha \parallel$ moves from 0 to a positive value, and $\parallel \lambda_T^\alpha \parallel = \mu^\alpha |\lambda_N^\alpha|$.

From a numerical standpoint, the transitions sliding-sticking or sticking-sliding have to be detected by the interpolation of the tangential velocities $\dot{g}_T$. Then we solve the equation $\dot{g}_T(t) = 0$ with some numerical method like the Newton's method, or the dichotomy. When the sticking occurs, the tangential velocity vanishes and the contact problem is formulated on the acceleration level as:

- Sticking: $\parallel \lambda_T^\alpha \parallel < \mu^\alpha |\lambda_N^\alpha|$, and $\parallel \ddot{g}_T^\alpha \parallel = 0$.
- Sliding: $\parallel \lambda_T^\alpha \parallel = \mu^\alpha |\lambda_N^\alpha|$, and $\parallel \ddot{g}_T^\alpha \parallel \neq 0$.

In the 2D case, if $\parallel \dot{g}_T^\alpha \parallel \neq 0$, the sliding contact can be formulated as

$$\begin{pmatrix} M(q) & -(G_N^{\alpha,T}(q) - \mu^\alpha \,\mathsf{sign}(\dot{g}_T) G_T^{\alpha,T}(q)) \\ G_N^\alpha(q) & 0 \end{pmatrix} \begin{pmatrix} \dot{v} \\ \lambda_N \end{pmatrix} = \begin{pmatrix} F(q,\dot{q},t) \\ \frac{dG_N^\alpha(q)}{dt}\dot{q} \end{pmatrix}. \tag{2.1}$$

The authors in [21] proved that if

- $M(q)$ is positive definite,
- $G_N^\alpha(q)$ have full column rank,
- $\mu^\alpha < \dfrac{\sigma_{min}(W^{NN}(q))}{\sigma_{max}(W^{TT}(q))}$, where: $W^{NN}(q) = G_N^\alpha M^{-1} G_N^{\alpha,T}$, $W^{TT}(q) = G_N^\alpha M^{-1} G_T^{\alpha,T}$, σ_{min} and σ_{max} denote the smallest and the biggest singular values of W^{NN} and W^{TT} respectively,

then $\ddot{q}$ and λ_N^α can be computed uniquely. Other results on the existence and uniqueness of a solution to the friction problem may be found in [7, 14, 15, 21, 41, 68, 69, 75, 76, 77, 85, 106, 107, 115]. Most of these works derive upper bounds on the friction coefficients which guarantee the existence of a solution.

When g_T^α vanishes, the sticking may be followed by a sliding phase. Glocker and Pfeiffer [62] have formulated sufficient conditions for a possible sticking-slipping transitions in the 2D case as

$$\begin{cases} \tilde{\lambda}_T^\alpha = \mu^\alpha \lambda_N^\alpha - |\lambda_T^\alpha| \\ \tilde{\lambda}_T^\alpha \geq 0,\ \ddot{g}_T^\alpha \leq 0,\ \tilde{\lambda}_T^\alpha \ddot{g}_T^\alpha = 0. \end{cases} \tag{2.2}$$

Due the non-linearity brought by the term $|\lambda_T^\alpha|$ in addition to the fact that the sliding in two possible directions, the resolution of the problem (2.2) requires the introduction of new variables. Then the problem is formulated as an LCP whose resolution enables to compute the acceleration and the contact force at sticking.

2.1.1.2 Global order of the event-driven scheme

Let us consider that the event-driven method incorporates a DAE solver of order p. Since in the event-driven strategy we stop at every single event to handle it and manage the jumps in the state, the influence of the non-smoothness on the order p is questionable. Janin and Lamarque [74] provided a theoretical and an numerical answer to this matter on a single-degree-of-freedom system whose motion is described with:

$$\ddot{x}(t) + 2a\dot{x}(t) + \omega^2 x(t) = f(t), \tag{2.3}$$

together with the Newton's impact law

$$x(t) = x_{max} \Longrightarrow \dot{x}(t^+) = -e\dot{x}(t^-). \tag{2.4}$$

The analysis shows that the order is affected by the accuracy of the detection of the impacts. Specifically, three means of detecting the impact times are addressed:

1. the first one, referred to as (IM1) consists of a linear interpolation using the approximation of the solution at the beginning and at the end of the time step,
2. the second one (IM2) uses a second order polynomial involving the approximations at the beginning and at the end of the time step and the derivative at the beginning of the step,
3. the third one (IM3) is the dichotomy method, for which the precision must be set to h^p where h is the time step size.

It is concluded that:

- if the numerical DAE solver is at least of order 2, and is used together with the methods (IM1) or (IM2), with f assumed to be differentiable and of a bounded derivative in $\mathbb{R}^+$, then the event-driven scheme is consistent and of order 2.

- if the numerical DAE solver is at least of order 3, and is used together with the method (IM2), with f assumed to be differentiable and of a bounded second derivative in $\mathbb{R}^+$, then the event-driven scheme is consistent and of order 3.

- if the numerical DAE solver is at least of order 4, and is used together with the method (IM3), with f assumed to be differentiable and of a bounded third derivative in $\mathbb{R}^+$, then the event-driven scheme is consistent and of order 4.

It also appears that the order is affected by the phenomenon of accumulation of impacts. In this case, the impacts are detected with an accuracy of h^p, however the transition to the sticking state is not very clear.

An experimental analysis is provided which consists in studying three numerical schemes, namely: the Newmark scheme ($\mathsf{p} = 2$), and two Runge-Kutta methods: RK24 ($\mathsf{p} = 3$) and DOPRI5 ($\mathsf{p} = 4$). The general findings in the case of multiple impacts are reported as follows:

- when using the Newmark scheme: the accuracy of the event-driven scheme is not improved by improving the localization method. When the time step size is small enough, then a second order accuracy for the velocity is reached.

- when using the RK24 method: the event-driven scheme is of order 3 with the (IM2) method and of order 4 with the (IM3) method.

- when using the DOPRI5 method: the event-driven is of order 4 with the (IM3) method set to the precision h^4.

2.1.1.3 Violation of the constraints

In continuous time, considering the constraints on the position, velocity or acceleration levels are the same provided that the initial conditions satisfy the constraints on the three aforementioned levels. When using a time discretization, this is no more valid. Therefore, when reducing the original index-3 system (1.22) to the index-2 DAE (1.23), we loose the information about the position constraint $g(q_i) = 0$. And when reducing it to the index-1 DAE (1.24), we lose information about both position and velocity constraints. To quote Simeon in [121], these two quantities are "*invariants of the system*". It is well know that invariants are not enforced under discretization and their violation is even increased due to truncation errors. Therefore, these invariants require a specific discretization in order to be conserved during time integration. The phenomenon of violation of the constraints is called the *drift-off phenomenon* and illustrates the fact that the numerical approximations do not belong to the manifold of the constraints. Let us make a global estimation of the propagation of this drift, w.r.t time, for the index-2 and index-1 formulations. We assume that at $t = 0$, the initial conditions of the problem verify

$$g(t_0) = g_0, \;\; \dot{g}(t_0) = \dot{g}_0, \;\; \ddot{g}(t_0) = \ddot{g}_0, \tag{2.5}$$

then by a first order approximation, that we obtain with a low order scheme, we get

$$\dot{g}(t) = (t - t_0)\ddot{g}_0 + \dot{g}_0 + \mathcal{O}(t - t_0)^2, \tag{2.6}$$

which means that the drift propagates linearly for the velocity constraints. With a second integration, we get

$$g(t) = \frac{1}{2}(t - t_0)^2\ddot{g}_0 + (t - t_0)\dot{g}_0 + g_0 + \mathcal{O}(t - t_0)^3, \tag{2.7}$$

which illustrates a quadratic drift of the acceleration constraints w.r.t time. For a detailed analysis of this phenomenon, we refer to [121, section 7.1.2]. To remedy this issue, there are two major solutions: stabilization of the constraints and projection on the constraints manifold.

Stabilization of the constraints In [18], Baumgarte proposed one of the first solutions to the problem of drift of the constraints. The idea is to consider a new constraint $\tilde{g}$ that is a linear combination of the original constraint and its time derivatives, as

$$\tilde{g} = \ddot{g} + 2\alpha\dot{g} + \beta^2 g = 0, \tag{2.8}$$

where α and β are two scalars. The constraint $\tilde{g}$ is now to replace the acceleration constraint $\ddot{g}$ in the system of equations (1.24), which becomes

$$\begin{pmatrix} M(q) & -G^T(q) \\ G(q) & 0 \end{pmatrix} \begin{pmatrix} \ddot{q} \\ \lambda \end{pmatrix} = \begin{pmatrix} F \\ -\ddot{g} - 2\alpha\dot{g} - \beta^2 g \end{pmatrix}, \tag{2.9}$$

where the scalars α and β are usually chosen to be positive in order to guarantee the stability of the solution of (2.9). This modification of the original constraint can be interpreted as a spring-damper model to bring back the constraints to their manifold. Indeed, the term β^2 can be seen as the spring coefficient, while 2α can be interpreted as a damping coefficient. This method has been widely used, and some of its drawbacks have been reported. Namely, there is no systematic method to choose the parameters α and β, which seem to be problem-dependent; in addition, giving them "large" values may lead to stiff systems.

Projection on the constraints manifold Let us denote Q_n and V_n the position and the velocity obtained by a DAE/ODE scheme at time t_n. In order to bring the positions to the invariant manifold, the minimization quadratic problem (2.10) has to be solved:

$$\begin{aligned} q_n &= \underset{q_n}{argmin}\ \frac{1}{2}(q_n - Q_n)^T A\,(q_n - Q_n) \\ &\text{subject to } \ g(q_n) = 0, \end{aligned} \tag{2.10}$$

where A is a symmetric and positive definite matrix. Likewise, another quadratic problem (2.11) is solved to bring the velocity V_n to the invariant manifold:

$$\begin{aligned} v_n &= \underset{v_n}{argmin}\ \frac{1}{2}(v_n - V_n)^T A\,(v_n - V_n) \\ &\text{subject to } \ G(q_n)v_n = 0. \end{aligned} \tag{2.11}$$

Usually, the inertia matrix M is chosen to solve the aforementioned problems (2.10) and (2.11), that is to say $A = M$. This choice may be explained by the fact that the inertia matrix already appears in the equations of motion, which enables to save the computation time that would be spent in computing a new matrix. From a numerical point of view, since the corrected positions and velocities are closed to the values provided by the integrators, using the inertia matrix prevents from updating the iteration matrix used in the Newton-Raphson method for solving (2.10) and (2.11). Another advantage of using the inertia matrix is the consistency

One method for correcting the position and velocity constraints is the *Lubich stabilization by projection* procedure [67], performed in two steps:

1. Projection on position constraint: The projected position q_n to the solution manifold is the solution of the system
$$\begin{cases} M(Q_n)(q_n - Q_n) + G^T(Q_n)\Lambda = 0 \\ g(q_n) = 0 \end{cases} \tag{2.12}$$
This system is solved with a nonlinear equations solver (Newton method for example).

2. Projection on velocity constraint: The projected velocity v_n is obtained by the resolution of the system
$$\begin{cases} M(Q_n)(v_n - V_n) + G^T(q_n)\Lambda = 0 \\ G(q_n)v_n = 0. \end{cases} \tag{2.13}$$

The scaling with the inertia matrix $M(Q_n)$ enables one to perform a resolution which is consistent with the metrics of the problem.

2.1.1.4 Discussion about event-driven schemes

Event-driven schemes have sensitivity to the thresholds, which are for example used for the evaluation of the index sets or for the occurrence of an event. They also have difficulties handling finite accumulations of impacts, called the *Zeno phenomenon.* Indeed, when too many events occur in a very short time interval, the simulation becomes slow because every single event has to be handled within the event-driven strategy. On the other hand, the separation of the smooth dynamics from the non-smooth dynamics enables to use high order schemes to compute the smooth part and also to efficiently adapt the time step size to the required precision and thus save the numerical effort. For more details and applications, we refer for example to [49, 113, 90, 89].

2.1.2 Time-stepping schemes

In time-stepping schemes [82, 83, 99, 100, 80, 9, 126, 60, 81] , the formulation of the dynamics enables to simultaneously handle the smooth dynamics and the non-smooth events. Let us consider a system with perfect unilateral constraints, the smooth dynamics can be formulated as a *differential inclusion* [4, 80, 99]

$$\begin{cases} \dot{q} = v \\ M(q)\dfrac{dv}{dt} = F(q,v,t) + r \\ -r \in N_C(q), \end{cases} \tag{2.14}$$

where r is the vector of generalized forces/reactions associated to the unilateral constraints, and

$$\begin{cases} C = \{q \in \mathbb{R}^n |\ g_j(q) \geq 0,\ j \in \{1\dots m\}\} \text{ is the admissible set,} \\ N_C(q) = \{\Lambda \in \mathbb{R}^n |\ \Lambda = -\displaystyle\sum_j \lambda_j \nabla g_j(q), \lambda_j \geq 0, \lambda_j g_j(q) = 0\}, \text{ is the normal cone to } C. \end{cases} \tag{2.15}$$

The basic idea behind these schemes is to consider the dynamics equations as a *measure differential inclusion.* On the velocity level, we get

$$\begin{cases} \dot{q} = v \\ M(q)dv = F(q,v,t)dt + di \\ -di \in N_{T_C(q)}(v^+), \end{cases} \tag{2.16}$$

where dv is the velocity measure, di is the reaction measure and dt is the Lebesgue measure. $T_{C(t)}$ denotes the tangent cone to C. It is worth noting that equation (2.16) is equivalent to an index reduction of system (2.14). The inclusion in (2.16) is known as Moreau's sweeping process of second order [98]. Let us consider a simple case to illustrate this fact. If $C = \mathbb{R}^+$, then

$$0 \leq \lambda \perp g \geq 0 \Leftrightarrow -\lambda \in N_{\mathbb{R}^+}(g) \supseteq N_{T_{\mathbb{R}^+}(g)}(\dot{g}^+), \tag{2.17}$$

therefore, we have

$$\begin{cases} g > 0 \Rightarrow \dot{g}^+ \in T_{\mathbb{R}^+}(g) = \mathbb{R} \text{ and } N_{\mathbb{R}^+}(g) = \{0\}, \text{ thus } \lambda = 0 \\ g = 0 \Rightarrow \dot{g}^+ \in T_{\mathbb{R}^+}(g) = \mathbb{R}^+ \text{ and } N_{\mathbb{R}^+}(g) = \mathbb{R}^- \text{ , thus } 0 \leq \dot{g}^+ \perp \lambda \geq 0. \end{cases} \tag{2.18}$$

Time-stepping schemes consider the integrals of the forces and allow for impulsive forces. The value of the impulse $\int_{t_n}^{t_{n+1}} di$ over a time step $[t_n,\, t_{n+1}]$ is considered as the primary unknown. J.J. Moreau and M. Jean [98, 82, 83] have performed a pioneering work in the field of simulation of nonsmooth dynamics, and the scheme they proposed, the *Moreau-Jean* method, is one of the most popular time-stepping schemes, it has proved its robustness while being simple. Since then, various schemes have been developed. In the following section, the Moreau-Jean scheme is discussed and some of the most popular time-stepping schemes are presented.

2.1.2.1 The Moreau-Jean time-stepping scheme

In the Moreau-Jean method, the unilateral contact is modeled with a Signorini condition at the velocity level, that is a complementarity between the contact impulse and the relative velocity, which ensures the impenetrability. In order to handle several contacts, with possibly changing status, an implicit algorithm is chosen to integrate the dynamics. Let us consider for simplicity that $F_{int}(q, v) = -C\dot{q} - Kq$ and that $M(q) = M$. The discrete form of the equations of motion reads

$$\begin{cases} q(t_{i+1}) = q(t_i) + \displaystyle\int_{]t_i,t_{i+1}]} v^+ \,\mathrm{d}t \\ M(v^+(t_{i+1}) - v^+(t_i)) = \displaystyle\int_{t_i}^{t_{i+1}} (-Cv(t) - Kq(t) + F_{ext}) \,\mathrm{d}t + \int_{]t_i,t_{i+1}]} \mathrm{d}i. \end{cases} \tag{2.19}$$

The smooth terms may be discretized using a $\theta-$method as

$$\begin{cases} q_{i+1} = q_i + h\,[\theta v_{i+1} + (1-\theta) v_i] \\ \displaystyle\int_{t_i}^{t_{i+1}} (Cv^+ + Kq) \,\mathrm{d}t = h\,[\theta (Cv_{i+1} + Kq_{i+1}) + (1-\theta)(Cv_i + Kq_i)] \\ \displaystyle\int_{t_i}^{t_{i+1}} F_{ext} \,\mathrm{d}t = h\,\left[\theta (F_{ext})_{i+1} + (1-\theta)(F_{ext})_i\right] \end{cases} \tag{2.20}$$

One then gets

$$\begin{cases} q_{i+1} = q_i + h\left[\theta v_{i+1} + (1-\theta)v_i\right] \\ v_{i+1} = v_{\mathsf{free}} + \hat{M}^{-1} p_{i+1} \\ \hat{M} = M + h\theta C + h^2\theta^2 K \\ v_{\mathsf{free}} = v_i + \hat{M}^{-1}\left[-hCv_i - hKq_i - h^2\theta K v_i + h\left[\theta(F_{ext})_{i+1} + (1-\theta)(F_{ext})_i\right]\right], \end{cases} \tag{2.21}$$

with $p_{i+1} \approx \int_{]t_i,t_{i+1}]} di$ is the approximation of the impulse over the time step. We can notice that the acceleration of the system is never explicitly computed because it becomes infinite for impulsive forces. The kinematic relations are discretized as follows for a contact α

$$\begin{cases} \dot{g}^\alpha_{n+1} = G^\alpha(q_n)v_{n+1} \\ P^\alpha_{n+1} \approx \displaystyle\int_{]t_n,t_{n+1}]} \mathrm{d}\lambda^\alpha \\ p^\alpha_{n+1} = G^{\alpha T} P^\alpha_{n+1} \\ g^\alpha_{n+1} = g^\alpha_n + h\left[\theta \dot{g}^\alpha_{n+1} + (1-\theta)\dot{g}^\alpha_n\right] \\ \tilde{q}_{n+1} = q_n + \dfrac{h}{2} v_n \\ \text{if } g^\alpha(\tilde{q}_{n+1}) \leq 0\,, \text{ then } 0 \leq \dot{g}^\alpha_{n+1} \perp P^\alpha_{n+1} \geq 0. \end{cases} \tag{2.22}$$

Finally, the Newton's law is implicitly formulated as

$$\begin{cases} \text{if}\, g^\alpha(\tilde{q}_{n+1}) \leq 0, \text{ then}\, 0 \leq U^\alpha_{n+1} + eU^\alpha_n \perp P^\alpha_{n+1} \geq 0 \\ \text{if}\, g^\alpha(\tilde{q}_{n+1}) > 0, \text{ then}\, P^\alpha_{n+1} = 0 \\ U^\alpha_n = \dot{g}^\alpha_n \\ U^\alpha_{n+1} = \dot{g}^\alpha_{n+1}. \end{cases} \tag{2.23}$$

In the case of a frictional impact, the nonsmooth problem formulated in (2.23) becomes:

$$\begin{cases} \hat{M}(v_{n+1} - v_{\mathsf{free}}) = p_{n+1} = \displaystyle\sum_\alpha p^\alpha_{n+1} \\ U^\alpha_{n+1} = G^\alpha(q_n)v_{n+1} = \hat{W}\displaystyle\sum_\alpha P^\alpha_{n+1} + G^\alpha(q_n)v_{\mathsf{free}} \\ p^\alpha_{n+1} = G^{\alpha,T}(q_n)P^\alpha_{n+1} \\ \text{if } g^\alpha(\tilde{q}_{n+1}) \leq 0\, \text{then :} \\ \qquad \hat{U}^\alpha_{n+1} = \left(U^\alpha_{N,n+1} + e^\alpha U^\alpha_{N,n} + \mu^\alpha \left\| U^\alpha_{T,n+1} \right\|_\infty, U^\alpha_{T,n+1}\right)^T, \forall\alpha \in I_1 \\ \qquad \mathbb{FC}^\star \ni \hat{U}^\alpha_{n+1} \perp P^\alpha \in \mathbb{FC} \\ \text{if } g^\alpha(\tilde{q}_{n+1}) > 0 \text{ then } P^\alpha_{n+1} = 0. \end{cases} \tag{2.24}$$

The frictional impact law formulated in the last four equations of (2.24) are derived from bipotential function of De Saxcé which enables to write a complementarity problem involving the modified velocity $\hat{U}^{\alpha}_{n+1}$ and the contact force P^{α}_{n+1} as discussed in [4, Chapters 3 and 10]. The subscripts N and T refer to the normal and tangent directions, respectively. In the case of a single contact, $\hat{W} = \begin{pmatrix} W^{NN} & W^{NT} \\ W^{TN} & W^{TT} \end{pmatrix}$, where $W^{NN} = G_N \hat{M}^{-1} G_N^T$, $W^{NT} = G_N \hat{M}^{-1} G_T^T$, $W^{TN} = G_T \hat{M}^{-1} G_N^T$ and $W^{TT} = G_T \hat{M}^{-1} G_T^T$. In the case of more than one contact, the extra diagonal blocks of the LCP matrix $\hat{W}$ must take into account the interactions between the active constraints. Let us write the matrix in the case of two active constraints α and β, for example:

$$\hat{W} = \begin{pmatrix} W^{NN}_{\alpha\alpha} & W^{NT}_{\alpha\alpha} & W^{NN}_{\alpha\beta} & W^{NT}_{\alpha\beta} \\ W^{TN}_{\alpha\alpha} & W^{TT}_{\alpha\alpha} & W^{TN}_{\alpha\beta} & W^{TT}_{\alpha\beta} \\ W^{NN}_{\beta\alpha} & W^{NT}_{\beta\alpha} & W^{NN}_{\beta\beta} & W^{NT}_{\beta\beta} \\ W^{TN}_{\beta\alpha} & W^{TT}_{\beta\alpha} & W^{TN}_{\beta\beta} & W^{TT}_{\beta\beta} \end{pmatrix}. \tag{2.25}$$

An implementation of the scheme is proposed in [4, Chapter 10] and a sketch of the algorithm is presented in appendix A.3.

In the Moreau-Jean's method, the constraints are discretized at the velocity level, therefore a violation of the position constraints may be observed during the simulation. This drift can be corrected by some means of projection on the constraints manifold as proposed in [3].

2.1.2.2 Other velocity-based time-stepping schemes

The Moreau-Jean time-stepping method has inspired various extensions which are briefly summarized hereafter.

In [125, 122], Stewart and Trinkle use the Moreau-Jean's scheme and consider the polyhedral approximation of the friction cone as proposed by Klarbring [84]. This approximation reads

$$\hat{FC}(q) = \{\lambda_N \mathbf{n} + D\beta \,|\, \lambda_N \geq 0,\, \beta \geq 0,\, e^T\beta \leq \mu\lambda_N\}, \tag{2.26}$$

where $e \in \mathbb{R}^k$, k being the number of edges of the polyhedral, the matrix $D \in \mathbb{R}^{3\times k}$ contains the directions $d_i \in \mathbb{R}^2$, and $\beta \in \mathbb{R}^k$ contains the weights corresponding to each direction. It is assumed that if d_i belongs to D, then there exists d_j in D such that $d_i = -d_j$. The polyhedral cone is illustrated in Fig.2.2. The discretized equations of motion read

$$\begin{cases} M_{n+1}(v_{n+1} - v_n) = h\tilde{n}\lambda_N + hD\beta + hF_{n+1} \\ q_{n+1} = q_n + hv_{n+1}, \end{cases} \tag{2.27}$$

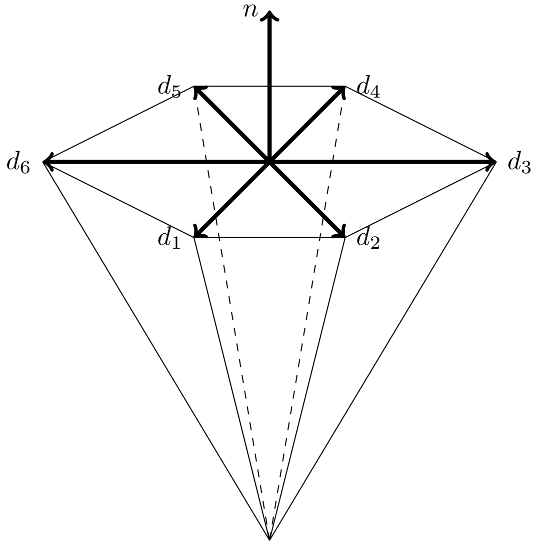


Figure 2.2: Polyhedral approximation of the friction cone

with $\tilde{n} = \nabla_q g(q)$. With the assumption of inelastic collisions and shocks, these equations are augmented by the complementarity relations describing the frictional contact:

$$\begin{cases} 0 \leq \tilde{n} q_{n+1} - \alpha_0 \perp \lambda_N \geq 0 \\ 0 \leq \alpha_1 e + D^T v_{n+1} \perp \beta \geq 0 \\ 0 \leq \mu \lambda_N - e^T \beta \perp \alpha_1 \geq 0, \end{cases} \tag{2.28}$$

where α_0 is a user defined tolerance, and α_1 is a positive scalar that satisfies $\alpha_1 \geq \max_i\{d_i^T v_{n+1}\}$. The first complementarity relation of (2.28), written on the position level, involves the normal multiplier. The other two complementarity relations describe the Coulomb friction model. Indeed, if $\mu\lambda_N - e^T\beta > 0$, then $\alpha_1 = 0$ and therefore $D^T v_{n+1} \geq 0$. If $D_i^T v_{n+1} > 0$, then there is an index j such that $D_j^T v_{n+1} = -D_i^T v_{n+1} < 0$, which contradicts the fact that $D^T v_{n+1} \geq 0$. Therefore $D_i^T v_{n+1} = 0$, which means that the tangential velocity is zero. In case there is some relative tangential motion, then there is at least one index i such that $D_i^T v_{n+1} > 0$, which implies $\alpha_1 > 0$. It follows that $\mu\lambda_N - e^T\beta = 0$, which means that the contact force lays on the boundary of the polyhedral cone.

Inspired by the work in [125, 122], Anitescu and Potra [10] proposed a time-stepping scheme in which the equations of motion are discretized with a backward Euler method, both unilateral and bilateral constraints are taken into account, on the velocity level, and the friction cone is approximated as in (2.28). The equations of motion read

$$M(v_{n+1} - v_n) = hF(q_{n+1}, v_{n+1}, t_{n+1}) + B_b^T(q_{n+1})\lambda_b + \tilde{n}\lambda_N + D\beta, \tag{2.29}$$

together with the complementarity relations

$$\begin{cases} B_b v_{n+1} = 0 \\ 0 \leq \tilde{n} v_{n+1} \perp \lambda_N \geq 0 \\ 0 \leq \alpha_1 e + D^T v_{n+1} \perp \beta \geq 0 \\ 0 \leq \mu \lambda_N - e\beta \perp \alpha_1 \geq 0, \end{cases} \tag{2.30}$$

where B_b represents the jacobian of bilateral constraints, λ_b is the contact forces vector associated with them.

This scheme is also able to handle partially elastic collisions using Poisson's model, as developed in a generalized framework by Pfeiffer and Glocker [49]. Complementarity relations corresponding to contact laws in the normal and tangent directions are formulated as an LCP which is solved with Lemke's algorithm that guarantees a solution within a finite number of iterations, provided that the constraints are linear. This work has then been extended to deal with stiff systems [11], by discretizing the dynamics with an implicit Euler method. In [8], the authors propose a time-stepping scheme in which bilateral and unilateral constraints are enforced. The constraints are linearized and implicit Euler method is used to discretize the dynamics.

Dzonou and Monteiro Marques [40] propose a numerical approximation to (2.14) where they consider a purely inelastic constraint. In this paper, the authors present the first proof of convergence and existence of a solution, in the frictionless case, under the assumption of a varying inertia matrix. They extended their work in [39] to elastic impact with a restitution coefficient. Paoli [108] proposed a time-stepping scheme where the formulation uses the proximal method. The author presents a proof of convergence of the proposed algorithm.

Funk and Pfeiffer [50] consider the measure differential equation (2.16) and discretize forces and velocities with a θ-method. Considering the case of planar friction, the authors formulate the contact problem as a linear system by decomposing the tangent velocity into positive and negative parts, and by writing the contact force as a linear combination of the two edges of the friction cone.

Forg, Pfeiffer and Ulbrich [112] discretize the dynamics implicitly and formulate the contact problem in the normal and tangent directions using the proximal point function. The gap function and the relative velocity are linearized with a Taylor expansion of first order and an iterative method is used to solve this augmented Lagrangian approach. This approach is then compared, on a benchmark, to LCP formulation solved with Lemke's algorithm, and to a projected Gauss-Seidel algorithm.

Actually, some researchers have explored ways in which they can increase the order of time-stepping schemes. Let us cite for example the work in [127] where the dynamics during smooth periods is computed using extrapolation methods, and with the classical Moreau-Jean scheme during impulsive methods. Extrapolation methods enable for a step size adjustment, and a suitably chosen minimum time step is used for the impulsive periods, in order not to break down the global order of the scheme.

Another solution for augmenting the order of classical time-stepping schemes is addressed in [2] where the author proposes a solution for augmenting the global order of an integration method incorporating the Moreau-Jean scheme. The proposed algorithm uses classical DAEs integrators during smooth periods. Events are roughly identified and the dynamics during nonsmooth periods is integrated using the Moreau-Jean time-stepping method. Conditions on the time step size of the Moreau-Jean method are identified which guarantee that the order of the DAE solver will not break down when switching to the Moreau-Jean scheme.

In [117, 118, 119] another option is proposed which consists of using the classical time-stepping scheme with the constraints on the velocity level, together with time-discontinuous Galerkin methods. The authors [119] propose to embed a classical time-stepping scheme in time discontinuous Galerkin methods, therefore allowing for a high order numerical approximation of the solution during the smooth periods.

Authors in [35] propose to split the contribution of smooth variables from that of the impulsive variables. The smooth variables can be computed using any classical high order DAE integrators. For instance, the implicit generalized-α method is chosen to take advantage of its ability of dealing with stiff dynamics. The impulsive variables are evaluated with the classical Moreau-Jean time-stepping scheme, which ensures the consistency of the global integration method. Both unilateral and bilateral constraints are discretized on the velocity level, which leads to an index-2 scheme. However, considering the velocity constraints leads to the violation of the position constraints, which can be a critical problem in some applications. As a solution to this issue, the authors in [31] propose to exactly solve both position and velocity constraints, based on the well-known approach of Gear, Gupta and Leimkuhler (GGL method). Contrary to [35], the bilateral constraints are taken into account as smooth variables which enables to handle them with high order DAE methods. The unilateral constraints are formulated as complementarity relations at both position and velocity levels and are implicitly solved.

2.1.2.3 A position-level time-stepping scheme: Schatzman-Paoli algorithm

Contrary to the aforementioned time-stepping schemes, Schatzman and Paoli propose a scheme that takes into account the constraints on the position level [109, 110]. In practice, this method is of order one with respect to the positions, and of order zero with respect to velocities. The authors discretize the equations of motion using finite differences. Their method reads

$$\begin{cases} M(q_{n+1})(q_{n+1} - 2q_n + q_{n-1}) = h^2 F(q_{n+1}, v_{n+1}, t_{n+1}) + p_{n+1} \\ v_{n+1} = \dfrac{q_{n+1} - q_{n-1}}{2h} \\ p_{n+1} = G^T(\dfrac{q_{n+1} + eq_{n-1}}{1+e})P_{n+1} \\ - p_{n+1} \in N_K(\dfrac{q_{n+1} + eq_{n-1}}{1+e}), \end{cases} \tag{2.31}$$

where $N_K(.)$ is the normal cone to the admissible set $K = \{q \in \mathbb{R}^n \,|\, g(q) \geq 0\}$. We note that (2.31) incorporates a Newton-like impact law, since the coefficient of restitution e is applied to the positions, while it is applied to the velocities in the original Newton law. Equation (2.31) can be written as

$$0 \leq g(\frac{q_{n+1} + eq_{n-1}}{1+e}) \perp G^T(\frac{q_{n+1} + eq_{n-1}}{1+e})P_{n+1} \geq 0. \tag{2.32}$$

Schatzman-Paoli's method projects the generalized coordinates on the admissible set. Therefore, the position constraints are enforced during the numerical integration of the motion. However, the computed velocity does not directly satisfy the Newton impact law, which gives no physical meaning to the impulse p_{n+1}. It goes without saying that p_{n+1} is an important quantity for most mechanical applications. Another issue related to this method is that in the case of a collision, the velocity is reversed two steps after the interval containing the time of the collision.

2.1.2.4 Discussion about time-stepping schemes

The advantage of time-stepping schemes over event-driven schemes is that no accurate event-detection is required and only one or no index sets are needed. This makes the former algorithms more robust for handling problems with several contact points as well as the problem of accumulation of impact (Zeno phenomenon). In addition, time-stepping algorithms have been proved to converge (under the assumption of independent constraints), which is not the case of event-driven methods. However, time-stepping schemes are of low order which may lead to the use of small time step sizes to meet the defined precision. This may lead to increase the numerical effort.

2.2 Numerical methods for ODEs and DAEs

When choosing the event-driven method as an integration strategy, the smooth dynamics is integrated using some numerical scheme that is suited to the formulation used: index-3 DAE, index-2 DAE or an index-1 DAE equivalent to an ODE when the Delassus operator is invertible. In this section, we present definitions of some concepts that are often used in this section, and then we address the numerical methods that are widely used in the field of computational mechanics.

2.2.1 A few definitions

Let us consider the initial value problem

$$\begin{cases} \dot{y} = f(y,t) \\ y(t_0) = y_0. \end{cases} \tag{2.33}$$

solved using a general one-step method of the form:

$$y_{n+1} = y_n + h\varphi(y_n, t_n, h), \tag{2.34}$$

where φ is an increment that depends on f, y_n, t_n and h. For numerical methods, the concept of *error* can be broken down into:

- Rounding error, due to finite precision of floating-point arithmetic.
- Truncation error, also called discretization error, due to the approximation method. Two types of truncation errors are defined:
  - local error, which is the error made in one step and can be written at time step n, starting from $y_{n-1} = y(t_{n-1})$, as
    $$l_n = y_n - y(t_n), \tag{2.35}$$
    where $y(.)$ denotes the exact solution.
  - global error, which is the difference between the exact solution and the approximation given by the method and reads
    $$e_n = y_n - y(t_n), \text{ starting from } y_0 = f(t_0). \tag{2.36}$$

The method is said to be *consistent to order p* if

$$l_n = \mathcal{O}(h^p), \tag{2.37}$$

and is said to be consistent if

$$\lim_{h \to 0} l_n = 0. \tag{2.38}$$

The scheme is said to be *convergent to order p* if

$$e_n = \mathcal{O}(h^p), \tag{2.39}$$

and is said to be convergent if

$$\lim_{h\to 0, t\to T} e_n = 0, \tag{2.40}$$

with $[0, T]$ being the interval of time during which the calculation has been performed. Another important concept has to be introduced: stability. In [55], Gear gives the following definition: "A one-step method is stable if for each differential equation satisfying a Lipschitz condition there exist positive constants h_0 and K such that the difference between two different numerical solutions y_n and $\tilde{y}_n$ each satisfying (2.34) is such that: $\parallel y_n - \tilde{y}_n \parallel \leq K \parallel y_0 - \tilde{y}_0 \parallel, \ \ \forall\, 0 \leq h \leq h_0$".

Finally, we should recall that according to Lax theorem, a consistent and stable method is convergent.

2.2.2 Methods for index-1 DAEs / ODEs

When the Delassus's operator $G(q)M^{-1}(q)G^T(q)$ is invertible, the index-1 DAE (1.24) is equivalent to the ODE (1.26) which is of the form

$$\dot{y} = f(y, t). \tag{2.41}$$

The literature in the field of numerical methods for ODEs is richer than that for DAEs because it is older. Numerical methods for ODEs can be split into to categories: ***one-step*** methods, and ***multistep*** methods. In the sequel, we will present some of the numerical schemes that are widely used in computational mechanics.

2.2.2.1 Runge-Kutta methods

Runge-Kutta methods [66, 67, 42, 121] are probably the most famous one-step schemes that are used for the numerical integration of ODEs. The discretization of (2.41) is given with

$$\begin{cases} Y_i = y_n + h \displaystyle\sum_{j=1}^{i} a_{ij} f(Y_j, t_n + c_j h), \quad i = 1 \ldots s, \\ y_{n+1} = y_n + h \displaystyle\sum_{j=1}^{s} b_j f(Y_j, t_n + c_j h), \end{cases} \tag{2.42}$$

where y_n denotes the approximation of the solution at the beginning of the time step, h is the size of the current time step, s is the number of stages, while a_{ij}, c_j and b_j are the coefficients of the method. Butcher [33] gave a more compact form to this formulation, using a tableau as

$$\begin{array}{c|c} c & A \\ --- & --- \\ & b^T \end{array}$$

Depending on the structure of the coefficients matrix A, RK methods can be split into explicit and implicit schemes.

- If $a_{ij} = 0$ for $i \leq j$, then we have *Explicit RK methods*, abbreviated as *ERK*,
- If $a_{ij} = 0$ for $i < j$ and at least one $a_{ii} \neq 0$, we have *Diagonal Implicit RK methods*, abbreviated as *DIRK*.
- In the other cases, we have fully *Implicit RK methods*, abbreviated as *IRK*.

Usually, the evaluation of the integration error is done by comparing the used method with another scheme, this can be costly from a numerical effort point of view. To remedy this, *Embedded RK* methods have been developed which consist of pairs of RK methods using the same coefficients c and A and differ in the coefficients b which are computed in order to get two estimations of the solution: one of order p and the other of order $p + 1$. The integration error is given as the difference between these two estimations. The ERK methods have bounded domains of stability, which becomes an issue especially for stiff ODEs. Probably, the most popular ERK schemes are RK4, Runge-Kutta-Fehlberg and Dormand-Prince scheme. These schemes are briefly presented in Appendix A.1.

IRK schemes remedy the issues of stability domain and handling stiff ODEs. The most popular IRK schemes are based on *collocation methods*, which consist in finding a polynomial $u \in \mathbb{P}^s$ ($\mathbb{P}^s$ being a set of polynomials of order s) satisfying

$$\begin{cases} u(t_n) = y_n \\ \dot{u}(t_n + c_i h) = f(u(t_n + c_i h), t_n + c_i h), \quad i = 1 \ldots s. \end{cases} \tag{2.43}$$

The coefficients c_i are called in this case the *collocation points*. Collocation methods are shown to be equivalent to RK methods, described with (2.42), with

$$\begin{cases} \text{for } \ j = 1 \ldots s : \\ \quad a_{ij} = \displaystyle\int_0^{c_i} l_j(\theta) d\theta \\ \quad b_j = \displaystyle\int_0^{1} l_j(\theta) d\theta \\ \quad l_j(\theta) = \displaystyle\prod_{k \neq j} \frac{\theta - c_k}{c_j - c_k} \\ u_{n+1} = u_n + \displaystyle\sum_{i=1}^{s} b_i f(u(t_n + c_i h), t_n + c_i h). \end{cases} \tag{2.44}$$

IRK schemes that are the most widely used are: $2s$-order Gauss methods for which the coefficients c_i are the roots of the Legendre polynomials $\frac{d^s}{dt^s}(t^s(t-1)^s)$, and $(2s-1)$-order Radau IIA methods, where the coefficients c_i are the zeros of the polynomial $\frac{d^{s-1}}{dt^{s-1}}(t^{s-1}(t-1)^s)$. For an application of IRK schemes in the context of multibody dynamics, we refer to [70, 71] or to [102] where the authors transform the initial index-3 DAE describing the dynamics of a multibody system into a State Space ODE (SSODE) by using the technique of coordinate partitioning. This SSODE is obtained after performing a partitioning on the coordinates to extract a set of minimal coordinates whose dynamics is formulated as an ODE. This SSODE is solved using the SDIRK (Singly Diagonally Implicit Runge Kutta) algorithm. Comparisons with the explicit DDEABM (Direct Differential Equations Adams-Bashforth-Moulton) solver on stiff mechanical systems show that the SDIRK method outperforms DDEABM; indeed, this latter can be 50 times more time-consuming than the SDIRK scheme. However, such a result is not surprising since the DDEABM is an explicit solver, and therefore it is not the best choice to deal with stiff dynamics because it uses very tight time steps for stability concerns.

2.2.2.2 Symplectic schemes

As said in Section 2.1.1.3, invariants of a given system require a specific treatment to be preserved. **Symplectic algorithms** [66, 67] can be used to this aim, particularly to preserve the Hamiltonian after discretization. These numerical methods have been widely investigated for long-time simulation of Hamiltonian systems, described with the index-3 DAE:

$$\begin{cases} \dot{q} = \dfrac{\partial H}{\partial p}(p,q) \\ \dot{p} = -\dfrac{\partial H}{\partial q}(p,q) + G^T(q)\lambda \\ g(q) = 0, \end{cases} \tag{2.45}$$

and whose symplectic structure is lost under discretization. In (2.45), the vector p is the momentum, H denotes the Hamiltonian function and is given by $H(p,q) = \dfrac{1}{2}p^T M^{-1}(q)p + U(q)$, U is the potential energy of the system. The most popular symplectic schemes are probably: *Störmer-Verlet*, *SHAKE* and *RATTLE* algorithms. The Störmer-Verlet scheme is intended for unconstrained Hamiltonian systems, whose discretization is given by

$$\begin{cases} q_{n+1} = q_n + \dfrac{h}{2}\big(\nabla_p H(p_{n+1/2}, q_n) + \nabla_p H(p_{n+1/2}, q_{n+1})\big) \\ p_{n+1/2} = p_n - \dfrac{h}{2}\nabla_q H(p_{n+1/2}, q_n) \\ p_{n+1} = p_{n+1/2} - \dfrac{h}{2}\nabla_q H(p_{n+1/2}, q_{n+1}). \end{cases} \tag{2.46}$$

The SHAKE algorithm is an adaptation of the Störmer-Verlet scheme for constrained Hamiltonian systems, whose discretization becomes

$$\begin{cases} q_{n+1} = q_n + hM^{-1}(q_{n+1/2})p_{n+1/2} \\ p_{n+1/2} = p_{n-1/2} - \dfrac{h}{2}\nabla_q U(q_n) + \dfrac{h}{2}G^T(q_n)\lambda_n \\ g(q_{n+1}) = 0. \end{cases} \tag{2.47}$$

To get around the difficulty of evaluating p_{n+1} that requires the evaluation of q_{n+1} in the Störmer-Verlet scheme, p_{n+1} is evaluated in the RATTLE scheme as

$$\begin{cases} p_{n+1} = p_{n+\frac{1}{2}} - \dfrac{h}{2}\nabla_q U(q_{n+1}) + \dfrac{h}{2}G^T(q_{n+1})\mu_n \\ G(q_{n+1})M^{-1}p_{n+1} = 0, \end{cases} \tag{2.48}$$

where the Lagrange multiplier μ_n arises from the hidden constraint $G(q)\dfrac{\partial H}{\partial p}(p,q) = 0$ obtained with a time differentiation of the constraint appearing in (2.45).

2.2.2.3 Families of Newmark and Generalized-α schemes

In the field of linear structural dynamics, where the equations of motion take the classical form

$$M\ddot{q} + C\dot{q} + Kq = F(t), \tag{2.49}$$

one of the most widely used schemes is probably the family of **Newmark schemes** [104, 86]. In (2.49), C denotes the damping matrix and K denotes the stiffness matrix. The discretization of the state variables in (2.49) is defined by

$$\begin{cases} M\ddot{q}_{n+1} + C\dot{q}_{n+1} + Kq_{n+1} = F(q_{n+1}, v_{n+1}, t_{n+1}) \\ q_{n+1} = q_n + h\dot{q}_n + \dfrac{h^2}{2}((1-2\beta)\ddot{q}_n + 2\beta\ddot{q}_{n+1}) \\ \dot{q}_{n+1} = \dot{q}_n + h((1-\gamma)\ddot{q}_n + \gamma\ddot{q}_{n+1}), \end{cases} \tag{2.50}$$

where β and γ are some coefficients that determine the stability and the order of the method. Indeed, if $\gamma \geq \frac{1}{2}$ and $\beta \geq \frac{\gamma+\frac{1}{2}^2}{4}$, then we have a stable method. It is worth mentioning that for $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$, we obtain the *trapezoidal method.* When $\beta = \frac{1}{2}$, then we have a 2^{nd} order accuracy. It is well known that when a multibody system contains some flexible bodies, classical numerical schemes are unable to properly deal with the stiffness arising from the presence of high frequencies. **HHT methods** [73, 34] alleviate this issue. These are a modification of the Newmark schemes that introduce some numerical dissipation in order to damp out the spurious frequencies without dropping the order to one. These schemes replace the equations of the dynamics as

$$M\ddot{q}_{n+1} + (1+\alpha)C\dot{q}_{n+1} - \alpha C\dot{q}_n + (1+\alpha)Kq_{n+1} - \alpha Kq_n = F_{n+1}, \tag{2.51}$$

where the state variables are discretized as in (2.50). For the solution to be stable, the coefficients α, β and γ must satisfy: $0.3 \leq \alpha \leq 0$, $\gamma = \frac{1}{2} - \alpha$ and $\beta = \frac{(1-\alpha)^2}{4}$. It is worth noting that for $\alpha = 0$, we retrieve the Newmark scheme. In [36], Chung and Hulbert proposed the **generalized-α methods**, a family of schemes that generalizes all the α methods. An acceleration-like variable a_n is introduced which is defined by the recurrence relation

$$(1-\alpha_m)a_{n+1} + \alpha_m a_n = (1-\alpha_f)\ddot{q}_{n+1} + \alpha_f \ddot{q}_n. \tag{2.52}$$

The discretization of the equations of motion becomes

$$\begin{cases} q_{n+1} = q_n + h\dot{q}_n + h^2(\frac{1}{2} - \beta)a_n + h^2\beta a_{n+1} \\ \dot{q}_{n+1} = \dot{q}_n + h(1-\gamma)a_n + h\gamma a_{n+1}, \end{cases} \tag{2.53}$$

where the constants α_f, α_m, β and γ are suitably chosen so that the scheme is stable for the linear dynamics. The algorithm is unconditionally stable [32] if the coefficients are chosen such that for $\rho_\infty < 1$,

$$\begin{cases} \alpha_m = \dfrac{2\rho_\infty - 1}{\rho_\infty + 1} < \dfrac{1}{2} \\ \alpha_f = \dfrac{\rho_\infty}{\rho_\infty + 1} < \dfrac{1}{2} \\ \gamma = \dfrac{1}{2} - \alpha_m + \alpha_f > \dfrac{1}{2} \\ \beta = \dfrac{1}{4}(\gamma + \dfrac{1}{2})^2. \end{cases} \tag{2.54}$$

The numerical damping is made easier in this scheme thanks to the coefficient ρ_∞, called *spectral radius parameter* . Indeed, $\rho_\infty = 0$ corresponds to asymptotic annihilation of the high frequencies, while $\rho_\infty = 1$ corresponds to no numerical damping.

At the beginning of the simulation, this variable is initialized as $a_0 = \ddot{q}_0$. The generalized-α scheme incorporates a multistep-like method since at each time step, the evaluation of the acceleration requires the knowledge of the acceleration-like variable computed in the previous step, as written in (2.52).

The generalized-α methods have been modified to be used in the context of constrained multibody systems whose dynamics is formulated as an index-2 or index-3 DAE, as discussed in Sections 2.2.3 and 2.2.4.

2.2.2.4 Multistep schemes

The previous methods require only the approximation of the solution at the beginning of the time step to compute the solution at the end of the step. Multistep methods require an initialization of the solutions, meaning that one or more estimations at the beginning of the simulation have to be computed using a one-step method. In this section, we limit ourselves to a breve presentation of three of the most widely used multistep methods: BDF, Adams-Bashforth and Adams-Moulton schemes [67, 121, 42].

The **BDF** methods are a class of implicit multistep schemes and are the most widely used to cope with stiff ODEs of the form (2.41). The idea behind these methods is to find a polynomial P that fits the solution y_{n+1} at the end of the time step and whose derivative interpolates the function f at k previous solutions, that is

$$\begin{cases} P(t_{n+1-i}) = y_{n+1-i}, \;\; i = 0 \dots k \\ \dot{P}(t_{n+1}) = f(y_{n+1}, t_{n+1}). \end{cases} \tag{2.55}$$

Using the Lagrange polynomials, one obtains the formula: $\sum_{i=0}^{k} \alpha_i y_{n-i} = h\beta_0 f(t_n, y_n)$ with $\alpha_0 = 1$. The backward differentiation formulas for orders from 1 to 6 are presented in Table A.1 of Annex A.2. For an order greater than 6, the stability domain is so small that the method is useless.

Adams methods are derived by transforming the ODE (2.41) into its integral form: $y(t) = y_0 + \int_{t_0}^{t_e} f(\theta, y(\theta)) d\theta$. Then the method is defined with $y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} P(\theta) d\theta$, where P is the polynomial that interpolates f at the k previously calculated solutions. **Adams-Bashforth** methods are explicit multistep methods that assume the general form $y_{n+1} = y_n + h \sum_{j=0}^{k-1} a_j f(y_j, t_j)$. The Adams-Bashforth formulas for orders from 1 to 5 are presented in Table A.2 of Annex A.2.

Adams-Moulton methods are implicit multistep methods. Their formulas have the general form: $y_{n+1} = y_n + h \sum_{j=0}^{k} a_j f(y_j, t_j)$, where k is the order of the method. The coefficients are derived in the same manner as for the Adams-Bashforth methods. Table A.3 of Annex A.2 shows the Adams-Moulton formulas for orders from 1 to 5.

When reducing the initial index 3 of the DAE by differentiating twice the constraints, the position level and velocity level constraints are not enforced anymore. Therefore, a great research work has been performed in order to extend the numerical schemes from ODEs to DAEs of higher index.

2.2.3 Methods for index-2 DAEs

To numerically compute the dynamics of a constrained multibody system formulated as a semi-explicit system of index 2 as in (1.23) , we can use **half-explicit** or **implicit** methods.

2.2.3.1 Half-explicit methods

Half-explicit schemes were initially introduced by V.Brasey and E.Hairer and we can cite their well known scheme in its 4^{th} order and 5^{th} order forms: HEM4 and HEM5 [25, 28, 66]. The discretization of the equations of motion in the HEM5 scheme is

$$\begin{cases} M(Q_i)\dot{V}_i = F(Q_i, V_i, t_n + c_i h) + G^T(Q_i)\Lambda_i \\ \dot{Q}_i = V_i \\ G(Q_i)V_i = 0,\ i = 1\ldots 8, \end{cases} \tag{2.56}$$

where the stages are defined by

$$\begin{cases} Q_i = q_n + h\sum_{j<i} a_{ij}\dot{Q}_j, \\ V_i = v_n + h\sum_{j<i} a_{ij}\dot{V}_j,\ i = 1\ldots 8. \end{cases} \tag{2.57}$$

The computation of coefficients c_i and a_{ij} is explained in [26]. At each stage, the estimations of position Q_i and velocity V_i are explicitly computed thanks to (2.57), while the acceleration $\dot{V}_i$ and the Lagrange multiplier Λ_i are obtained by solving the implicit system

$$\begin{pmatrix} M(Q_i) & -G^T(Q_i) \\ G(Q_{i+1}) & 0 \end{pmatrix} \begin{pmatrix} \dot{V}_i \\ \Lambda_i \end{pmatrix} = \begin{pmatrix} F(Q_i, V_i, t_n + c_i h) \\ r_i \end{pmatrix},\ i = 1\ldots 8, \tag{2.58}$$

where

$$r_i = -\frac{G(Q_{i+1})}{ha_{i+1,i}}(v_n + h\sum_{j=1}^{i-1} a_{i+1,j}\dot{V}_j). \tag{2.59}$$

Note that the matrix in (2.58) is not necessarily symmetric since we evaluate the first line of (2.56) at $t_n + c_i h$ and the third line of (2.56) at $t_n + c_{i+1}h$ to form the linear system. At the end of the time step, the numerical solution is given by

$$q_{n+1} = Q_9 = q_n + h\sum_{i=1}^{8} b_i\dot{Q}_i \tag{2.60a}$$

$$v_{n+1} = V_9 = v_n + h\sum_{i=1}^{8} b_i\dot{V}_i, \tag{2.60b}$$

with $b_i = a_{9i}$. In order to get the acceleration and the Lagrange multiplier at the end of the time step, an additional linear system has to be solved

$$\begin{pmatrix} M(q_{n+1}) & -G^T(q_{n+1}) \\ G(q_{n+1}) & 0 \end{pmatrix} \begin{pmatrix} \dot{v}_{n+1} \\ \lambda_{n+1} \end{pmatrix} = \begin{pmatrix} F(q_{n+1}, v_{n+1}, t_{n+1}) \\ r_{n+1} \end{pmatrix} \tag{2.61}$$

where $r_{n+1} = -G_{qq}(v_{n+1}, v_{n+1}) = -(\frac{dG(q)}{dt}v)\Big|_{(q_{n+1},v_{n+1},t_{n+1})}$. This method is of order 5 for the positions and the velocities, and of order 3 for the Lagrange multipliers.

In the case of the HEM5 solver, the estimations of velocities are built in such a way that there is no drift of the constraints at the velocity level in the internal stages. Indeed, from Equation (2.58), one can deduce that

$$G(Q_{i+1})\dot{V}_i = \frac{G(Q_{i+1})}{ha_{i+1,i}}(v_n + h\sum_{j<i} a_{ij}\dot{V}_j)\,, \forall i = 1\dots 7. \tag{2.62}$$

Then, we have

$$G(Q_{i+1})v_n + ha_{i+1,i}G(Q_{i+1})\dot{V}_i + hG(Q_{i+1})\sum_{j<i} a_{ij}\dot{V}_j \;=\; 0, \forall i = 1\dots 7. \tag{2.63}$$

On the other hand, from Equation (2.57), by multiplying the second equation by $G(Q_{i+1})$, one can deduce that

$$\begin{aligned} G(Q_{i+1})V_{i+1} &= G(Q_{i+1})v_n + hG(Q_{i+1})\sum_{j<i} a_{ij}\dot{V}_j \\ &= G(Q_{i+1})v_n + ha_{i+1,i}G(Q_{i+1})\dot{V}_i + hG(Q_{i+1})\sum_{j<i} a_{ij}\dot{V}_j \\ &= 0. \end{aligned} \tag{2.64}$$

Finally, from Equation (2.63), we have

$$G(Q_i)V_i = 0\,, \forall i = 1\dots 8. \tag{2.65}$$

For the last stage, on the one hand, we have

$$G(Q_9)\dot{V}_8 = \frac{G(Q_9)}{hb_8}(v_n + h\sum_{j<8} b_i\dot{V}_j), \tag{2.66}$$

and on the other hand

$$V_9 = v_n + h\sum_{j\leq 8} b_i\dot{V}_j. \tag{2.67}$$

Multiplying the last equation by $G(Q_9)$, we obtain

$$G(Q_9)\,V_9 = G(Q_9)\,v_n + h\,G(Q_9)\sum_{j\leq 8} b_i\dot{V}_j = G(Q_9)\,v_n + h\,b_8G(Q_9)\dot{V}_9 + h\,G(Q_9)\sum_{j\leq 7} b_i\dot{V}_j \tag{2.68}$$

Finally, we can write $G(Q_9)V_9 = 0$.

For more details about half-explicit methods, the reader is referred to [13, 66]. It is worth noting that half-explicit methods ensure the enforcement of the velocity level constraints since they are directly solved, however their is still a drift of the position constraints that must be treated properly.

2.2.3.2 Partitioned Runge-Kutta methods

Half-explicit schemes suffer from an order reduction when it comes to the calculation of the Lagrange multipliers [12]. However the problem can be avoided by improving the approximation of the Lagrange multipliers, for this we refer to [12] in which the author suggests to introduce an additional stage to get a better approximation of λ_{n+1} and substitute the first stage by an explicit Runge-Kutta stage. These methods have been given the name of *Type B Half-Explicit schemes*, most popular examples include the PHEM56 scheme of Murua [101]. With this class of methods, the discretization reads

$$\begin{cases} \dot{Q}_i = V_i \\ M(Q_i,\tau_i)\dot{V}_i = F(Q_i,V_i,\tau_i) + G^T(\bar{Q}_i,\tau_i)\Lambda_i \\ G(\bar{Q}_i,\bar{\tau}_i)\bar{V}_i = 0, \end{cases} \tag{2.69}$$

where

$$\begin{cases} Q_i = q_n + h\sum_{j<i} a_{ij}V_j \\ V_i = v_n + h\sum_{j<i} a_{ij}\dot{V}_j \\ \bar{Q}_i = q_n + h\sum_{j\leq i} \bar{a}_{ij}V_j \\ \bar{V}_i = v_n + h\sum_{j\leq i} \bar{a}_{ij}\dot{V}_j \\ \tau_i = t_n + c_i h \\ \bar{\tau}_i = t_n + \bar{c}_i h. \end{cases} \tag{2.70}$$

The coefficients a_{ij} are the components of a strictly lower triangular matrix A and $\bar{a}_{ij}$ are the coefficients of a lower triangular matrix $\bar{A}$. The PHEM56 scheme is a 6-stage partitioned half-explicit method of order 5. At each stage, the accelerations and the Lagrange multipliers are evaluated with:

$$\begin{pmatrix} M(Q_i,\tau_i) & -G^T(Q_i,\tau_i) \\ G(\bar{Q}_i,\bar{\tau}_i) & 0 \end{pmatrix} \begin{pmatrix} \dot{V}_i \\ \Lambda_i \end{pmatrix} = \begin{pmatrix} F(Q_i,V_i,\tau_i) \\ r_i \end{pmatrix} \tag{2.71}$$

with $r_i = -\dfrac{G(\bar{Q}_i,\bar{\tau}_i)}{h\bar{a}_{i,i}}(v_n + h\sum_{j=1}^{i-1}\bar{a}_{i,j}\dot{V}_j)$. Once again, we are dealing with a non-symmetric matrix in (2.71). The acceleration a_{n+1} and the multipliers λ_{n+1} at the end of the time step can be evaluated with one of the following two solutions:

- Solution 1: Use (2.71) with: $\bar{Q}_i = Q_i = q_{n+1}$, $\bar{V}_i = V_i = v_{n+1}$ and replace r_i with $G_{q_{n+1}q_{n+1}}(v_{n+1}, v_{n+1})$.

- Solution 2: adding a 7^{th} stage
  Compute $\bar{Q}_7 = q_n + h\sum_{j\leq i}\bar{a}_{7j}V_j$
  Solve

$$\begin{pmatrix} M(Q_6,\tau_6) & -G^T(Q_6,\tau_6) \\ G(\bar{Q}_7,\bar{\tau}_7) & 0 \end{pmatrix} \begin{pmatrix} a_{n+1} \\ \lambda_{n+1} \end{pmatrix} = \begin{pmatrix} F(Q_6,V_6,\tau_6) \\ r_7 \end{pmatrix}. \tag{2.72}$$

The solution at the end of the time step is given with $q_{n+1} = \bar{Q}_6$ and $v_{n+1} = \bar{V}_6$. The PHEM56 schemes solves the constraints at the velocity level. Therefore, at each time step, we verify $G(\bar{Q}_i)\bar{V}_i = 0$. This scheme is of order 5 for the positions and velocities, and of order 3 for the Lagrange multipliers [12].

2.2.3.3 Generalized-α schemes for index-2 DAEs

The most popular technique to overcome the issue of violation of the position constraints in the index–2 formulation is certainly that due to Gear/Gupta/Leimkuhler [56] who added the position constraints to the equations of (1.23) and gave rise to the *GGL method*, where the equations of motion read

$$\begin{cases} \dot{q} = v + G^T(q)\mu \\ M(q)\dot{v} = F(q,v,t) + G^T(q)\lambda \\ G(q)v = 0 \\ g(q) = 0. \end{cases} \tag{2.73}$$

As seen in the first equation of (2.73), this method requires the introduction of additional Lagrange multipliers μ that vanish analytically $\mu(t) = 0$. In [56] the authors solve (2.73) using the BDF schemes. The GGL method has been extended to the case of multibody systems with unilateral constraints, as in the work of Acary [3] where the GGL method is used to enforce the position and the velocity constraints in the Moreau-Jean time-stepping scheme, and the work of Brüls et al. [31] that has already been presented in section 2.1.2.

In [91], C. Lunk and B. Simeon present a customization of the generalized–α to the index-2 DAE case based on the idea of Gear/Gupta/Leimkuhler.Therefore, using (2.53) and introducing the holonomic constraints, the discretization of the equations of motion reads

$$\begin{cases} M(q_{n+1})\dfrac{q_{n+1}-q_n}{h} = M(q_{n+1})(v_n + h(\dfrac{1}{2}-\beta)a_n + h\beta a_{n+1}) + h\big((\dfrac{1}{2}-\bar{\beta})G^T(q_n) + \bar{\beta}G^T(q_{n+1})\big)\lambda_{n+1} \\ M(q_{n+1})\dfrac{v_{n+1}-v_n}{h} = M(q_{n+1})((1-\gamma)a_n + \gamma a_{n+1}) + \dfrac{1}{2}(G^T(q_n)\lambda_{n+1} + G^T(q_{n+1})\mu_{n+1}) \\ (1-\alpha_m)M(q_{n+1})a_{n+1} = \alpha_f F(q_n, v_n, t_n) + (1-\alpha_f)F(q_{n+1}, v_{n+1}, t_{n+1}) - \alpha_m M(q_n)a_n \\ G(q_{n+1})v_{n+1} = 0 \\ g(q_{n+1}) = 0, \end{cases} \tag{2.74}$$

where a denotes the acceleration-like variable. We can mention that in addition to the classical coefficients (β, α_m, α_f and γ) of the α–method, there is a new parameter $\bar{\beta}$ that is introduced in the first equation of (2.74). The second order convergence for position and velocity is proved in [91], first order accuracy is ensured for the Lagrange multipliers. Separately, Jay and Negrut [78] also applied the generalized-α method to the GGL formulation, leading to a discretization that is almost the same as the one in (2.74).

The generalized-α scheme can be adapted to the resolution of the dynamics described by the index-2 DAE (1.23). The discretization of the system is then

$$\begin{cases} q_{n+1} = q_n + h\dot{q}_n + (\dfrac{1}{2}-\beta)h^2 a_n + 2\beta h a_{n+1} \\ \dot{q}_{n+1} = \dot{q}_n + (1-\gamma)h a_n + \gamma h a_{n+1} \\ G(q_{n+1})v_{n+1} = 0, \end{cases} \tag{2.75}$$

In this case, the correction step is performed using Newton iterations to solve the linear system

$$\begin{pmatrix} \beta' M(q_{n+1}, t_{n+1}) + \gamma' C_t(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) + K_t(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) & -G^T(q_{n+1}) \\ G(q_{n+1}) & 0 \end{pmatrix} \begin{pmatrix} \Delta\dot{q} \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} R_q \\ G(q_{n+1})v_{n+1}. \end{pmatrix} \tag{2.76}$$

where $\beta' = \frac{1-\alpha_m}{h\gamma(1-\alpha_f)}$ and $\gamma' = \frac{h\beta}{\gamma}$, $R_q = M(q_{n+1})\ddot{q}_{n+1} - F(q_{n+1}, v_{n+1}, t_{n+1}) - G^T(q_{n+1})\lambda_{n+1}$.

While the scheme in (2.74) takes into account only the holonomic constraints, a more interesting and general extension of the generalized–α method is proposed in [79] that also takes into account nonholonomic constraints as well as non-constant mass matrices.

Starting from the *underlying ODE*

$$\ddot{q} = M^{-1}(q)(F(q, v, t) + G^T(q)\lambda), \tag{2.77}$$

the authors in [133] wanted to bring a solution to the problem of the numerical oscillations that appear when applying the α-schemes family to index-2 or index-3 DAEs. By differentiating twice the constraints, projecting on the position and velocity constraints manifold and eliminating a_{n+1} from the equations, the discretization of (2.77) reads:

$$\begin{cases} M_{n+1}(q_{n+1} - \hat{q}_n) - \hat{\beta}h^2 F_{n+1} + G_{n+1}^T \nu_{n+1} = 0 \\ M_{n+1}(v_{n+1} - \hat{v}_n) - \hat{\gamma}h F_{n+1} + G_{n+1}^T \mu_{n+1} = 0 \\ G_{n+1}v_{n+1} = 0 \\ g(q_{n+1}) = 0, \end{cases} \tag{2.78}$$

where

$$\begin{cases} \hat{q}_n = q_n + hv_n + h^2\big((\frac{1}{2} - \beta)a_n - \beta\alpha M_n^{-1}(F_n - G^T\lambda_n)\big) \\ \hat{v}_n = v_n + h\big((1-\gamma)\ddot{q}_n - \gamma\alpha M_n^{-1}(F_n - G^T\lambda_n)\big) \\ \hat{\beta} = \beta(1+\alpha) \\ \hat{\gamma} = \gamma(1+\alpha). \end{cases} \tag{2.79}$$

The Lagrange multiplier ν results from the projection on the position constraints:

$$\begin{cases} M(q)(q - \tilde{q}) + G^T(q)\nu = 0 \\ g(q) = 0, \end{cases} \tag{2.80}$$

while μ results from the projection on the velocity constraints:

$$\begin{cases} M(q)(v - \tilde{v}) + G^T(q)\mu = 0 \\ G(q)v = 0, \end{cases} \tag{2.81}$$

where $\tilde{q}$ and $\tilde{v}$ are respectively the projections of the positions and velocities on the manifolds. We note that this scheme is nothing but the Generalized-α scheme applied to the so-called underlying ODE. While in the classical generalized-α for an index-2 DAE, the velocity constraints are solved exactly, this scheme needs additional methods (projection on the constraints manifold as in (2.81)), which represents additional computational time.

The method can also be applied for the index-1 formulation using the *coordinate split formulation*. For instance if an annihilation matrix $P(q)$ is used such that $P(q)G^T(q) = 0$, the method reads

$$\begin{cases} P_{n+1}\big(M_{n+1}(q_{n+1} - \hat{q}_n) - \hat{\beta}h^2 F_{n+1}\big) = 0 \\ P_{n+1}\big(M_{n+1}(v_{n+1} - \hat{v}_n) - \hat{\gamma}h F_{n+1}\big) = 0 \\ G_{n+1}v_{n+1} = 0 \\ g(q_{n+1}) = 0. \end{cases} \tag{2.82}$$

Numerical tests have been performed that show that the proposed algorithms are stable and able to damp out the spurious oscillations.

2.2.4 Methods for index-3 DAEs

Index-3 DAEs are difficult to solve. Indeed, dedicated schemes face difficulties to deal with the instabilities that arise from the constraints and it is usually advisable to reduce the index to compute the dynamics. However, many authors have proposed extensions of some classical schemes initially designed for ODEs. To cite but one example, the joint work of M. Arnold and O. Brüls.

In [32], M. Arnold and O. Brüls present an extension of the generalized–α method for index–3 DAEs, in which the discretization of the state variables (q and $\dot{q}$) is the same as in (2.53), with the discretization of the position constraints as

$$g(q_{n+1}) = 0. \tag{2.83}$$

This scheme is based on a prediction step and a correction step where some Newton iterations are performed in order to reduce the dynamical and the constraint residuals defined by

$$\begin{cases} R_q = M(q_{n+1}, t_{n+1})\ddot{q}_{n+1} - F(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) - G^T(q_{n+1})\lambda_{n+1} \\ R_\lambda = g(q_{n+1}). \end{cases} \tag{2.84}$$

The Newton iterations amount to solving the following linear system

$$\begin{pmatrix} \beta^{'} M(q_{n+1}, t_{n+1}) + \gamma^{'} C_t(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) + K_t(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) & -G^T(q_{n+1}) \\ G(q_{n+1}) & 0 \end{pmatrix} \begin{pmatrix} \Delta\ddot{q} \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} R_q \\ R_\lambda \end{pmatrix}, \tag{2.85}$$

where $\beta^{'} = \frac{1-\alpha_m}{h^2\beta(1-\alpha_f)}$, $\gamma^{'} = \frac{\gamma}{h\beta}$, $K_t = \frac{\partial(M\ddot{q}-F+G^\top\lambda)}{\partial q}$ is the stiffness matrix, and $C_t = -\frac{\partial F}{\partial \dot{q}}$ is the damping matrix. The second order convergence of position, velocity and acceleration is ensured provided that the coefficients satisfy (2.54).

Negrut et al. have also worked on the adaptation of the HHT scheme for constrained multibody systems in [103]. The authors propose a discretization of the dynamics that reads

$$\frac{1}{1+\alpha}(M\ddot{q})_{n+1} + (G^T\Lambda - F)_{n+1} - \frac{\alpha}{1+\alpha}(G^T\Lambda - F)_n = 0. \tag{2.86}$$

Methods to estimate the integration error and to control the time step size are also proposed. The authors compared their scheme to the GStiff solver on mechanical systems with large number of degrees of freedom. GStiff is a solver of stiff mechanical systems, it can manage index-3, index-2 and index-1 DAE forms of the equations of motion, which are integrated using BDF methods. Results of the comparisons show that the GStiff solver is 3 to 5 times more time-consuming than the HHT scheme. This result may be explained by the fact that the BDF method faces stability problems and accuracy issues when the step size changes frequently. Therefore, the GStiff solver needs tight time steps to handle the accuracy demanded by the user.

2.2.5 Stability of the DAEs numerical solvers

The complexity of solving the initial index-3 DAE form of the equations of motion comes from the propagation of the errors in the dynamical variables: positions, velocities, accelerations and contact forces. These errors can make the integration of the dynamics very difficult and must therefore be treated properly. In this section, we will analyze the propagation of errors and recall the scaling that has been proposed to obtain well-conditioned systems. In [23], the authors analyze the propagation of the errors in the dynamical variables computed from the integration of an index-3 formulation using the BDF family, which takes the form $\sum_{i=0}^{k} \alpha_i y_{n-i} = h\beta_0 f(t_n, y_n)$. However, results may be extended to other numerical schemes. The authors address the problem with the assumption that the mass matrix is the identity, as

$$\begin{cases} \dot{q} = v \\ \dot{v} = F(q, \dot{q}, t) + G^T(q)\lambda \\ g(q) = 0. \end{cases} \tag{2.87}$$

When using implicit schemes, some Newton iterations have to be performed, which take the form of a linear system

$$Az = b. \tag{2.88}$$

In our case, the iteration matrix is

$$A = hJ_n = \begin{pmatrix} \alpha_0 I & -hI & 0 \\ hX & \alpha_0 I + hY & -hG \\ hG^T & 0 & 0 \end{pmatrix}, \tag{2.89}$$

where

$$\begin{cases} X = -\dfrac{\partial F}{\partial q}(q, \dot{q}, t) - G^T(q)\lambda \\ Y = -\dfrac{\partial F}{\partial v}(q, \dot{q}, t). \end{cases} \tag{2.90}$$

The inverse of the iteration matrix is given by

$$A^{-1} = (hJ_n)^{-1} = \frac{1}{\alpha_0} \begin{pmatrix} I - T & \gamma(I-T)R^{-1} & \gamma^{-1}R^{-1}GS \\ -\gamma^{-1}T & (I-T)R^{-1} & \gamma^{-2}R^{-1}GS \\ -\gamma^{-2}SG^T & -\gamma^{-1}SG^T R^{-1} & \gamma^{-3}S \end{pmatrix}, \tag{2.91}$$

where

$$\begin{cases} R = I + \gamma Y + \gamma^2 X \\ S = \left(G^T R^{-1} G\right)^{-1} \\ T = R^{-1} G S G^T \\ \gamma = \dfrac{h}{\alpha_0}. \end{cases} \tag{2.92}$$

The authors use a result of Petzold and Lötstedt stating that the accuracy of the ith component of the solution of (2.88) can be evaluated as

$$|\Delta z_i| \leq r\epsilon \sum_j |(A^{-1})_{ij}| \, \|A\|_\infty \|z + \Delta z\|_\infty, \tag{2.93}$$

where r is some unknown coefficient that is of order of the size of A [72], ϵ is the machine accuracy and

$$\begin{cases} (A + \Delta A)(z + \Delta z) = b \\ \|\Delta A\|_\infty \leq r\epsilon \, \|A\|_\infty. \end{cases} \tag{2.94}$$

Therefore, using (2.91) and (2.93), the round-off errors are:

$$\begin{cases} \Delta q = \mathcal{O}(h^{-1}) \\ \Delta v = \mathcal{O}(h^{-2}) \\ \Delta\lambda = \mathcal{O}(h^{-3}) \\ \quad C = \|A\|_\infty \|A^{-1}\|_\infty = \mathcal{O}(h^{-3}). \end{cases} \tag{2.95}$$

We can see then that when the time step size becomes tight the Lagrange multipliers and the conditioning number C deteriorates quickly. To remedy this, the authors propose not only a left and a right preconditioning, but also to scale the λ which is the most affected by the round-off errors. The dynamics reads:

$$\begin{cases} \dot{q} = v \\ \dot{v} = F + sG^T\tilde{\lambda} \\ g = 0, \end{cases} \tag{2.96}$$

where $\tilde{\lambda} = \frac{\lambda}{s}$ and $s = \mathcal{O}(h^{-2})$ is the scaling factor. The Newton iterations are modified as $\tilde{A}\tilde{z} = \tilde{b}$, where

$$\begin{cases} \tilde{A} = D_L A D_R \\ \tilde{z} = {D_R}^{-1} z \\ \tilde{b} = D_L b. \end{cases} \tag{2.97}$$

D_L and D_R are the left and the right preconditioning matrices, respectively. They are defined by

$$D_L = \begin{pmatrix} I & 0 & 0 \\ 0 & hI & 0 \\ 0 & 0 & h^{-1}I \end{pmatrix}, D_R = \begin{pmatrix} I & 0 & 0 \\ 0 & h^{-1}I & 0 \\ 0 & 0 & h^{-2}I \end{pmatrix}. \tag{2.98}$$

When using the left-right-preconditioning, the round-off errors are:

$$\begin{cases} \Delta q = \mathcal{O}(h^0) \\ \Delta v = \mathcal{O}(h^0) \\ \Delta \tilde{\lambda} = \mathcal{O}(h^0) \\ \quad C = \|A\|_\infty \|A^{-1}\|_\infty = \mathcal{O}(h^0). \end{cases} \tag{2.99}$$

The authors have used this preconditioning on several mechanical systems, and proved that their solution makes the variables less sensitive to round-off errors, better than using only left or right preconditioning.

In [24], the authors have performed an asymptotic analysis on the index-3 DAE formulation of the dynamics, integrated this time using the Newmark family. After eliminating the acceleration-like variable a_{n+1}, the index-3 DAE (1.22) can be discretized using the Newmark scheme as follows:

$$\begin{cases} q_{n+1} - q_n = h\Big(\dfrac{\beta}{\gamma} v_{n+1} + (1 - \dfrac{\beta}{\gamma} v_n)\Big) - \dfrac{h^2}{2}\Big(1 - \dfrac{2\beta}{\gamma}\Big) a_n \\ \dfrac{1}{\gamma h} M_{n+1}(v_{n+1} - v_n) = F_{n+1} + G_{n+1}^T \lambda_{n+1} - \Big(1 - \dfrac{1}{\gamma}\Big) M_n a_n \\ g(q_{n+1}) = 0. \end{cases} \tag{2.100}$$

This system is linearized and solved using Newton iterations in which a linear system of the form (2.88) is solved with

$$A = \begin{pmatrix} X & \frac{1}{\gamma h} U & -G \\ I & -\frac{\beta h}{\gamma} I & 0 \\ G^T & 0 & 0 \end{pmatrix}, \tag{2.101}$$

where

$$\begin{cases} X = -\dfrac{\partial F}{\partial q}(q_{n+1}, v_{n+1}, t_{n+1}) - G^T(q_{n+1})\lambda \\ Y = -\dfrac{\partial F}{\partial v}(q_{n+1}, v_{n+1}, t_{n+1}) \\ U = M(q_{n+1}) + h\gamma Y. \end{cases} \tag{2.102}$$

b is defined as

$$b = \begin{pmatrix} -\frac{1}{h\gamma}M_{n+1}(v_{n+1}-v_n) + (F_{n+1}+G_{n+1}^T\lambda_{n+1}) - \frac{1}{\gamma}M_n a_n\Big) \\ -(q_{n+1}-q_n) + h\Big(\frac{\beta}{\gamma}v_{n+1} + (1-\frac{\beta}{\gamma}v_n)\Big) - \frac{h^2}{2}\Big(1-\frac{2\beta}{\gamma}\Big)a_n \\ -g(q_{n+1}) \end{pmatrix}. \tag{2.103}$$

The inverse of this iteration matrix is given with

$$A^{-1} = \begin{pmatrix} h^2\beta W & WU & T^{-1}GR^{-1} \\ h\gamma W & \frac{\gamma}{h\beta}(I-WU) & -\frac{\gamma}{h\beta}T^{-1}GR^{-1} \\ -R^{-1}G^TT^{-1} & \frac{1}{h^2\beta}R^{-1}G^TT^{-1}U & \frac{1}{h^2\beta}R^{-1} \end{pmatrix}, \tag{2.104}$$

where

$$\begin{cases} T = M + h\gamma Y + h^2\beta X \\ R = G^TT^{-1}G \\ S = GR^{-1}G^T \\ W = T^{-1}(I - ST^{-1}). \end{cases} \tag{2.105}$$

Therefore, we can write

$$\lim_{h\to 0} A = \begin{pmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^{-1}) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^1) & 0 \\ \mathcal{O}(h^0) & 0 & 0 \end{pmatrix}, \tag{2.106}$$

and

$$\lim_{h\to 0} A^{-1} = \begin{pmatrix} \mathcal{O}(h^2) & \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^1) & \mathcal{O}(h^{-1}) & \mathcal{O}(h^{-1}) \\ \mathcal{O}(h^0) & \mathcal{O}(h^{-2}) & \mathcal{O}(h^{-2}) \end{pmatrix}. \tag{2.107}$$

Likewise, we have

$$\lim_{h\to 0} \Delta b = \begin{pmatrix} \mathcal{O}(h^0) \\ \mathcal{O}(h^0) \\ \mathcal{O}(h^0) \end{pmatrix}. \tag{2.108}$$

Expanding (2.88) in Taylor series about $\epsilon = 0$ and after some computations that are not presented here, the authors obtain

$$\begin{aligned} |\lim_{h\to 0} z_i(h,\epsilon)| &\leq \sum_j |\lim_{h\to 0} A_{ij}^{-1}(h,\epsilon)|\,|\lim_{h\to 0} b_j(h,\epsilon)| \\ &\leq \|\lim_{h\to 0} A^{-1}(h,\epsilon)\|_{\infty} \|\lim_{h\to 0} b(h,\epsilon)\|_{\infty}. \end{aligned} \tag{2.109}$$

The result expressed in (2.109) provides a quantification of the effect of a perturbation of b on the solution. Finally, using (2.107), (2.108) and (2.109), the authors deduce the round-off errors:

$$\begin{cases} \Delta q = \mathcal{O}(h^0) \\ \Delta v = \mathcal{O}(h^{-1}) \\ \Delta\tilde{\lambda} = \mathcal{O}(h^{-2}) \\ \quad C = \|A\|_\infty \|A^{-1}\|_\infty = \mathcal{O}(h^{-3}). \end{cases} \tag{2.110}$$

They propose a left-right-conditioning in which the matrices are

$$D_L = \begin{pmatrix} \beta h^2 I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}, D_R = \begin{pmatrix} I & 0 & 0 \\ 0 & \frac{\gamma}{\beta h} I & 0 \\ 0 & 0 & \frac{1}{\beta h^2} I \end{pmatrix}, \tag{2.111}$$

where γ and β are the coefficients of the Newmark schemes. This preconditioning lead to reduce the effects of the round-off errors, which become

$$\begin{cases} \Delta q = \mathcal{O}(h^0) \\ \Delta v = \mathcal{O}(h^0) \\ \Delta\tilde{\lambda} = \mathcal{O}(h^0) \\ \quad C = \|A\|_\infty \|A^{-1}\|_\infty = \mathcal{O}(h^0). \end{cases} \tag{2.112}$$

In this section, we could form a quantitative idea on the amplifications of the errors and perturbations, w.r.t to the time step size. Pre-conditioning the involved matrices proves to be a good solution to this issue, despite the additional computational effort related to this method.

2.2.6 A few words on the KKT systems

We note that the matrix that appears in equation (1.25), called the augmented matrix, is met very often. When the matrix M is positive definite, the problem formulated in (1.25) can be written as a quadratic optimization problem, as described in [105]. Indeed, let us consider the quadratic problem defined as

$$\begin{aligned} &\underset{v}{\text{minimize}} \quad p(v) = \frac{1}{2} v^T M(q) v + v^T d \\ &\text{subject to} \quad G(q) v = b. \end{aligned} \tag{2.113}$$

Let us assume that $G(q)$ is full row rank. The first-order necessary condition for equation (2.113) to have a solution $\hat{v}$ is the existence of a variable $\hat{\lambda}$ satisfying

$$\begin{pmatrix} M(q) & -G^T(q) \\ G(q) & 0 \end{pmatrix} \begin{pmatrix} \hat{v} \\ \hat{\lambda} \end{pmatrix} = \begin{pmatrix} -d \\ b \end{pmatrix}. \tag{2.114}$$

In optimization, the augmented matrix is called the *Karush-Kuhn-Tucker (KKT)* matrix. Let $R(q)$ denote the null space matrix of the constraints matrix $G(q)$, that is $G(q)R(q) = 0$. If $R^T MR$ is positive definite, then the KKT matrix is nonsingular and the problem (2.114) has a unique solution $(\hat{v}, \hat{\lambda})$ [105, lemma 16.1].

Another result on the existence of a solution of the KKT system (1.24) is given in [21]:

- if $G(q)$ has full column rank and $M(q)$ is positive semi-definite, then the KKT matrix in (1.24) is non-singular if and only if $\mathsf{ker}(M(q)) \cap \mathsf{ker}(G(q)) = 0$. In this case, there is a unique solution $(\ddot{q}, \lambda)$ for (1.24).

- if $G(q)$ has an arbitrary rank but satisfies the compatibility of the constraints, that is to say $\frac{dG}{dt}(q)\dot{q} \in \mathsf{Im}(G(q))$, then a solution $(\ddot{q}, \lambda)$ exists. Moreover, $(\ddot{q}, G^T(q)\lambda)$ are unique if and only if $\mathsf{ker}(M(q)) \cap \mathsf{ker}(G(q)) = 0$ is satisfied.

It may happen that the KKT matrix be assymetric as in (2.58) or (2.71). In [21] such systems are proved to have a unique solution if and only if the rank of the KKT matrix is $n + m$. Let us recall that in (2.58) and (2.71) $M(q) \in \mathbb{R}^{n\times n}$ and $G(q) \in \mathbb{R}^{m\times n}$.

2.2.7 Time step selection

A general method to compute the optimal step size is the *halved step sizes* method, which is described in the scheme of Fig.2.3. We can define two approximations of the solution at the end of two successive time steps, each one of size h: y^1_{n+2} obtained with a direct integration starting from time t_n to time t_{n+2}, and a more precise one y^0_{n+2} obtained with an integration from t_n to t_{n+1} and then to t_{n+2}. In order to explain this method, let us consider that the numerical approximation is obtained with a Runge Kutta method of order p. Let y_{n+1} be the approximation of $y(t_n + h)$. The truncation error of y_{n+1} is:

$$\begin{tikzcd}
y_n \arrow[r, "h"] \arrow[rrd, "2h"] & y_{n+1} \arrow[r, "h"] & y^0_{n+2} \\
& & y^1_{n+2}
\end{tikzcd}$$

Figure 2.3: halved steps method

$$l_1 = y(t_n + h) - y_{n+1} = Ch^{p+1} + \mathcal{O}(h^{p+2}), \tag{2.115}$$

where C depends on the coefficients of the method and also on the derivatives of f of order $p+1$. The error of y^0_{n+2} contains the transported error from t_n to t_{n+1}, added to the error from t_{n+1} to t_{n+2}, therefore:

$$l_2^0 = y(t_n + 2h) - y_{n+2}^0 = 2Ch^{p+1} + \mathcal{O}(h^{p+2}). \tag{2.116}$$

Similarly, the error of y_{n+2}^1 is

$$l_2^1 = y(t_n + 2h) - y_{n+2}^1 = C(2h)^{p+1} + \mathcal{O}(h^{p+2}). \tag{2.117}$$

From equations (2.116) and (2.117), we can eliminate the constant C and deduce a better **extrapolated** approximation $\tilde{y}_2$ for $y(t_n + 2h)$ as

$$\begin{cases} y(t_n + 2h) = \tilde{y}_2 + \mathcal{O}(h^{p+2}) \\ \tilde{y}_2 = y_{n+2}^0 + \dfrac{y_{n+2}^1 - y_{n+2}^0}{2^p - 1}. \end{cases} \tag{2.118}$$

Finally, the optimal step size is defined as

$$h_{opt} = h\left(\frac{tol}{\parallel \tilde{y}_2 - y_{n+2}^0 \parallel}\right)^{\frac{1}{p}}, \tag{2.119}$$

or

$$h_{opt} = s\,h\left(\frac{tol}{\parallel y_{n+2}^1 - y_{n+2}^0 \parallel}\right)^{\frac{1}{p}}, \tag{2.120}$$

where *tol* is the user defined precision, s is a safety coefficient that increases the probability to have an acceptable error for the next step, h is the size of the previous step and p is the order of consistency of the method. But this method is expensive from the computational point of view.

2.2.7.1 Time step control for the generalized-α scheme

To evaluate the integration error of the generalized-α scheme, we follow the methodology proposed by Géradin and Cardona in [57] and Negrut et al. in [103] for the HHT schemes family. The exact value of the positions vector can be approximated by a limited Taylor series development around $t_{n+1} = t_n + h$:

$$q_e(t_n + h) = q_n + h\dot{q}_n + \frac{h^2}{2}\ddot{q}_n + \frac{h^3}{6}\frac{d\ddot{q}_n}{dt} + \mathcal{O}(h^4). \tag{2.121}$$

The integration error is computed as

$$l_n = q_{n+1} - q_e(t_n + h). \tag{2.122}$$

By substituting the expression of q_{n+1} from (2.53) into (2.122), we get:

$$l_n = h^2(\frac{1}{2} - \beta)a_n + h^2\beta a_{n+1} - \frac{h^2}{2}\ddot{q}_n - \frac{h^3}{6}\frac{d\ddot{q}_n}{dt} + \mathcal{O}(h^4). \tag{2.123}$$

The third derivative $\frac{d\ddot{q}_n}{dt}$ of the position can be approximated by

$$\frac{d\ddot{q}_n}{dt} = \frac{\ddot{q}_{n+1} - \ddot{q}_n}{h} + \mathcal{O}(h). \tag{2.124}$$

By substituting this expression of $\dddot{q}_n$ into (2.123) and using the relations in (2.53), we obtain

$$l_n = q_{n+1} - q_n - \frac{1}{h}\dot{q}_n - \frac{h^2}{3}\ddot{q}_n - \frac{h^2}{6}\ddot{q}_{n+1} + \mathcal{O}(h^4). \tag{2.125}$$

We compute the optimal time step size using (2.120) with $p = 2$.

2.2.7.2 Time step selection for the HEM5 solver

The *halved step sizes* method is very expensive since the calculation of the truncation error needs many evaluations of the derivatives. To avoid this, we will rather use a method that is suited to Runge-Kutta schemes: *Embedded Runge-Kutta formulas.* These methods were first proposed by Merson (1957), further methods have been proposed by Fehlberg (1964). They provide 2 approximations of the solution using the same number of estimations of the right-hand side member. These approximations are used to compute the truncation error and then the optimal step size for a much lower price than with the *halved step sizes* method.

Concerning the HEM5 solver, V.Brasey and E.Hairer [26] define an error based on the 7^{th} and 8^{th} estimations

$$\mathsf{l}_1 = \|q_{n+1} - Q_8\|_s = \mathcal{O}(h^4) \tag{2.126}$$

$$\mathsf{l}_2 = \left\|q_{n+1} - q_n - h(\frac{5}{2}V_7 - \frac{3}{2}V_8)\right\|_s = \mathcal{O}(h^3). \tag{2.127}$$

Finally:

$$\mathsf{l_n} = \frac{\mathsf{l}_1{}^2}{\mathsf{l}_1 + c\,\mathsf{l}_2} = \mathcal{O}(h^5), \tag{2.128}$$

where c is a scalar that ensures a good combination between l_1 and l_2 such that l_n is of order 5. This scalar is set by default to 0.01 in [26]. The optimal step size is computed with Eq. (2.120) with $p = 5$. We experimentally studied the influence of the coefficient c by evaluating the order of the error on the slider-crank mechanism discussed in Section 3.1. For different values of c, and for different values of the time step size, we compute the error in (2.128). In Fig.2.4, we draw the integration error with respect to the time step size, and we observe an order 5 of the error, for all the values of c. However, we also see that this error is reduced when we increase the value of c from 0.001 to 10. We can say that the coefficient c is problem-dependent and its calibration is therefore hard to perform in a unique manner. A a consequence, we cannot clearly know if the computed error $\mathsf{l_n}$ is over-estimated or under-estimated unless we know an analytical solution of the problem which enables us to evaluate the integration error.

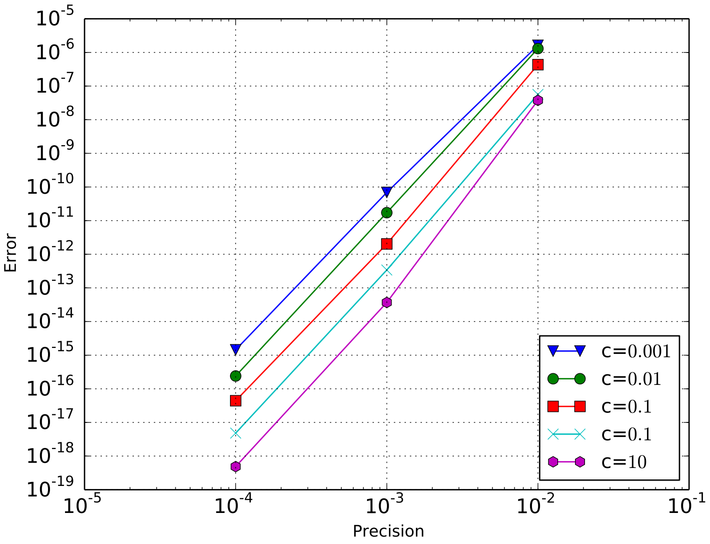


Figure 2.4: Order of the integration error of HEM5

2.2.7.3 Time step selection for the PHEM56 scheme

The integration error is estimated with:

$$\mathsf{e} = ||q_{n+1} - \tilde{Q}_7|[, \tag{2.129}$$

where $\tilde{Q}_7 = q_n + h\sum_{j\leq i}\bar{b}_j V_j$ and $V_7 = \bar{V}_6 = v_{n+1}$. Finally, the optimal time step is given with Eq. (2.120) where $p = 5$.

2.2.7.4 Time step selection for the 4th order RK-Fehlberg method

The computational error is the difference between the 5^{th} and 4^{th} order estimations of the solution, as

$$\mathsf{e} = \|{y_{n+1}}^{(5)} - {y_{n+1}}^{(4)}\| \tag{2.130}$$

Finally, the expression of the optimal step size is given by Eq.(2.120) with $p = 4$.

2.2.8 Dense output

We often need to evaluate a variable at some points which do not correspond to those of the discretization times. For instance, in the event-driven strategy, we need to detect the time of occurrence of a new event, an impact or a detachment for example. For this, we use either an interpolation of the required variables, or even better, a *dense output method*, which is a continuous function which will provide the value of a given variable at any given time at a given accuracy. In the case of a contact, we need to evaluate the gap function $g(q)$ at each point of the step to detect the time t^*at which it vanishes in a given time interval $[t_n,\ t_{n+1}]$, as illustrated in Fig. 2.5.

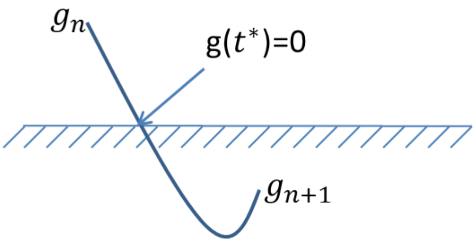


Figure 2.5: Example of a contact active during a step

2.2.8.1 Hermite interpolation

This quintic Hermite interpolation consists in finding a curve $\psi(s)$ that fits

$$\begin{cases} \psi(0) = q_0 \\ \dot{\psi}(0) = v_0 \\ \ddot{\psi}(0) = a_0 \\ \psi(1) = q_1 \\ \dot{\psi}(1) = v_1 \\ \ddot{\psi}(1) = a_1, \end{cases} \tag{2.131}$$

considering that 0 and 1 are the extremities of the interval of interpolation. Using the basis function representation:

$$\psi(s) = H_0(s)q_0 + H_1(s)v_0 + H_2(s)a_0 + H_3(s)a_1 + H_4(s)v_1 + H_5(s)q_1 \tag{2.132}$$

one needs to find polynomials that satisfy the conditions announced above. These polynomials are

$$\begin{cases} H_0(s) = 1 - 10s^3 + 15s^4 - 6s^5 \\ H_1(s) = s - 6s^3 + 8s^4 - 3s^5 \\ H_2(s) = 0.5s^2 - 1.5s^3 + 1.5s^4 - 0.5s^5 \\ H_3(s) = 0.5s^3 - s^4 + 0.5s^5 \\ H_4(s) = -4s^3 + 7s^4 - 3s^5 \\ H_5(s) = 10s^3 - 15s^4 + 6s^5. \end{cases} \tag{2.133}$$

The roots of $\psi(s)$ can be computed using a bisection method for example. The first event, whose time of occurrence corresponds to the smallest real value, is handled first. The parameter s corresponds in our case to

$$s = \frac{t - t_n}{h}, \tag{2.134}$$

where t is time at which we would like to evaluate the solution, t_n is the beginning of the time step and h is the time step size.

2.2.8.2 Dense output for the HEM5 scheme

The dense output proposed by V.Brasey and E.Hairer [26, 27] is of order 4 and satisfies the velocity constraints. Basically, the approach reads

$$\begin{cases} u(t) = y_n + h\displaystyle\sum_{i=1}^{s^\star} \hat{b}_i(\theta) F(Y_i),\ t \in (t_n, t_{n+1}),\ \theta = \frac{t - t_n}{h},\ Y_i = [q_i, v_i]^T \\ g(u(t)) = 0 \\ Y_i = y_n + h\displaystyle\sum_{j=1}^{i-1} a_{ij} F(Y_j) \\ g(Y_i) = 0. \end{cases} \tag{2.135}$$

In order to have a 4th order dense output, it is proved in [27] that we should have :

$$\begin{cases} \hat{b}_1(\theta) = 0,\ \forall\theta \in [0,\ 1] \\ \hat{b}_2(\theta) = 0,\ \forall\theta \in [0,\ 1] \\ \hat{b}_3(\theta) = 0,\ \forall\theta \in [0,\ 1] \\ \hat{b}_4(\theta) = 0,\ \forall\theta \in [0,\ 1]. \end{cases} \tag{2.136}$$

The HEM5 scheme comprises 8 internal stages in addition to the last stage that gives the numerical solution. This last stage is considered as the 9th stage of the dense output, which therefore contains 10 stages, thus $s^\star = 10$. The method is as follows:

- Let us denote: $Q_9 = q_{n+1}$ and $V_9 = v_{n+1}$.
- Construct the 10th stage for the dense output, that is:
  1. Form $Q_{10} = q_n + h(a_{10,6}V_6 + a_{10,7}V_7 + a_{10,9}V_9)$
  2. Compute $\dot{V}_9$ and Λ_9 with

$$\begin{pmatrix} M(Q_9) & -G^T(Q_9) \\ G(Q_{10}) & 0 \end{pmatrix} \begin{pmatrix} \dot{V}_9 \\ \Lambda_9 \end{pmatrix} = \begin{pmatrix} F(Q_9, V_9) \\ r_9 \end{pmatrix}, \tag{2.137}$$

where $r_9 = -\dfrac{G(Q_{10})}{ha_{10,9}}(v_n + h(a10,6\dot{V}_6 + a10,7\dot{V}_7 + a_{10,9}\dot{v}_{n+1}))$.

3. Form $V_{10} = v_n + h(a_{10,6}\dot{V}_6 + a_{10,7}\dot{V}_7 + a_{10,9}\dot{V}_9)$

- Construct vectors p_i and w_i that satisfy:

$$\begin{cases} p_i = D_{i4}V_4 + D_{i5}V_5 + D_{i6}V_6 + D_{i7}V_7 + D_{i8}V_8 + D_{i9}V_{10}, \ i = 1,2,3,4 \\ w_i = D_{i4}\dot{V}_4 + D_{i5}\dot{V}_5 + D_{i6}\dot{V}_6 + D_{i7}\dot{V}_7 + D_{i8}\dot{V}_8 + D_{i9}\dot{V}_9, \ i = 1,2,3,4, \end{cases} \tag{2.138}$$

where D_{ij} are some coefficients.

- The dense outputs $q_d(\theta)$ and $v_d(\theta)$ for the position and velocity are given with:

$$\begin{cases} q_d(\theta) = q_n + h\theta(p_1 + \theta(p_2 + \theta(p_3 + \theta p_4))) \\ v_d(\theta) = v_n + h\theta(w_1 + \theta(w_2 + \theta(w_3 + \theta w_4))). \end{cases} \tag{2.139}$$

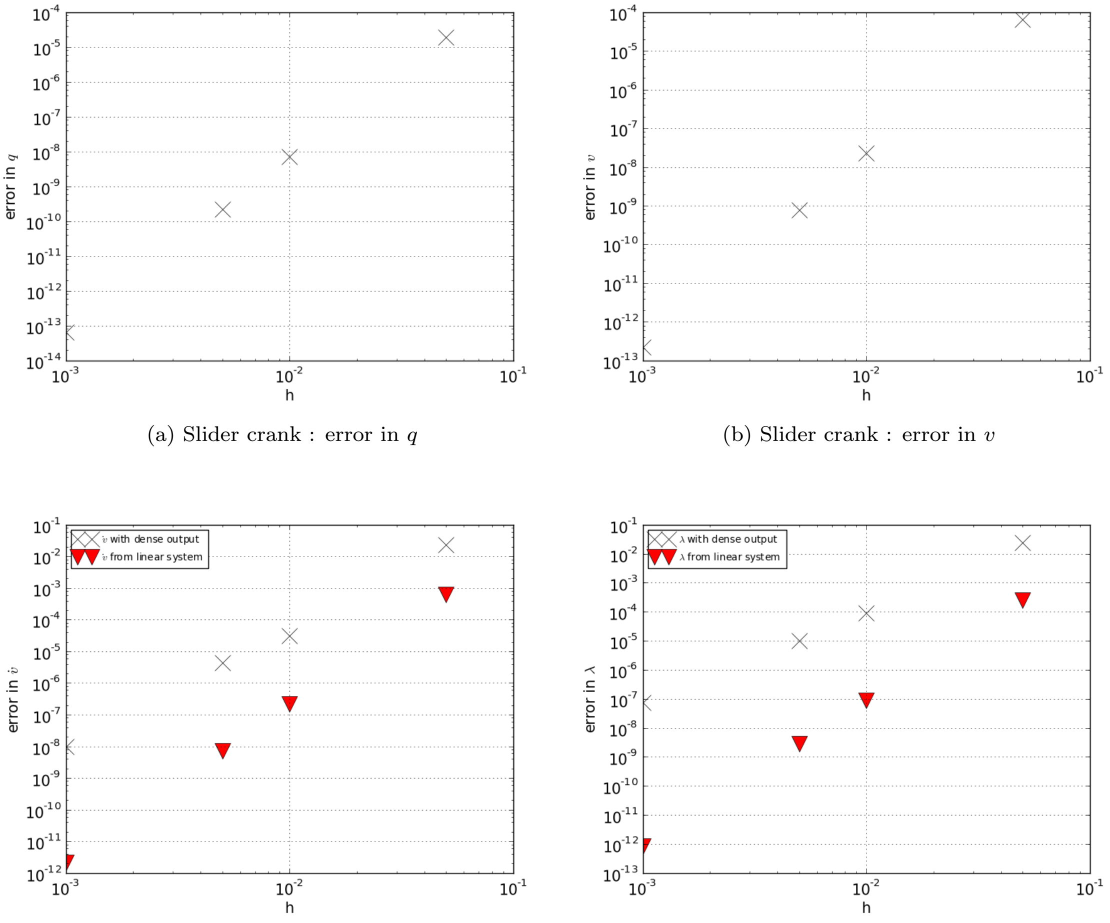
This fitting enables to construct a polynomial approximation P_g

$$P_g(t) = g(q_d(t)), \tag{2.140}$$

of the gap function, and thus makes it easier to look for a root of P_g to detect the occurrence of a contact. Moreover, if we need to detect some detachment, then we need to evaluate the Lagrange multipliers at any time during the simulation. Indeed, a change in the sign of the multipliers during a time step would indicate that a detachment has happened. In this section, we propose a dense output for the accelerations by taking the time derivative of the velocities, and we propose one for the Lagrange multipliers using linear combinations of the 4^{th} to 6^{th} estimations of these variables. These dense outputs read:

$$\begin{cases} \ddot{q}_d(\theta) = w_1 + 2\theta w_2 + 3\theta^2 w_3 + 4\theta^3 w_4 \\ x_i = D_{i4}\Lambda_4 + D_{i5}\Lambda_5 + D_{i6}\Lambda_6 + D_{i7}\Lambda_7 + D_{i8}\Lambda_8 + D_{i9}\Lambda_9 \ \ i = 1,2,3,4 \\ \lambda(\theta) = x_1 + 2\theta x_2 + 3\theta^2 x_3 + 4\theta^3 x_4. \end{cases} \tag{2.141}$$

Let us consider a slider-crank described in Section 3.1.2. We perform a simulation with a time step of 10^{-6} in which we compute accelerations and Lagrange multipliers using Eq.(2.61). The output of this simulation will be considered as a reference solution. We also perform other simulations using larger time steps: 0.05, 0.01, 0.005 and 0.001. For each of these simulations, we compute positions and velocities using the dense output formulas. These values will be compared to the reference values, which will enable us to observe the order of the dense output. On the other hand, we compute the accelerations and Lagrange multipliers by two means:



(a) Slider crank : error in q

(b) Slider crank : error in v

(c) Slider crank : error in $\dot{v}$

(d) Slider crank : error in λ

Figure 2.6: The slider crank: errors made when using the dense output

1. Using equation (1.25) as

$$\begin{pmatrix} M(q(t)) & -G^T(q(t)) \\ G(q(t)) & 0 \end{pmatrix} \begin{pmatrix} \ddot{q}(t) \\ \lambda(t) \end{pmatrix} = \begin{pmatrix} F(q(t), v(t), t) \\ R(t) \end{pmatrix}, \tag{2.142}$$

   with $R(t) = -G_{q(t)q(t)}(v(t), v(t))$. $q(t)$ and $v(t)$ being the positions and velocities given by the dense output (2.139).

2. Using the dense output of λ (2.141).

Results of the comparisons are presented in Fig.2.6. For the positions and velocities, we observe an order close to 5. When accelerations and Lagrange multipliers are computed from the linear system

(1.25), the error is of order 4 but the construction of the block matrix that appears in (1.25) and the evaluation of the second member of the acceleration-level constraints are time consuming. It is worth noting that the obtained orders meet the expected theoretical order for the dense output of the positions, velocities and accelerations, which is 4 [27]. There is no clear theoretical result for the order to expect for the dense output of the Lagrange multipliers, but we know that this order cannot exceed the order of the approximation given by the discretization of the HEM5 method, which is 3. When the accelerations are computed with (2.141), the error is of order 3. Thus, we lose one order of magnitude on the precision with respect to the first method. However we do not have any additional numerical effort but forming linear combinations of variables that have already been calculated in the stages. Let us recall that the theoretical order of the Lagrange multipliers computed with the HEM5 scheme is 3 and can break-down as discussed in section 2.2.3.2. The dense output of the Lagrange multipliers may be used to find the time when it becomes null and therefore detect a detachment.

2.3 Conclusion

In this chapter, we discussed two major integration families that are used for the integration of the equations of motion of nonsmooth multibody systems, namely: event-driven schemes and time-stepping schemes. Event-driven methods stop at every single event, handle the event and continue the integration starting from the time of occurrence of the event. These methods allow for using high order DAE solvers during the smooth period and require accurate methods for detecting the events in order not to break down the order of the DAE scheme. Event-driven schemes are efficient for systems with few number of contacts and become inconsistent and time-consuming for large number of contacts. Time-stepping scheme provide a unified framework for solving both the smooth and the nonsmooth dynamics. They have a low local order in general, but they have proved their efficiency for large number of contacts. Furthermore, time-stepping schemes have convergence proofs as well as energy conservation properties [63].

Moreover, a summary of several numerical methods for index-3, index2 and index-1 DAEs was presented. The literature is abundant with numerical integrators for ODEs and DAEs, however there is a few results ([102, 103, 12] for example) proving the efficiency of these methods, by for example testing them on some industrial benchmarks. Neverthless, the bibliography presented in section 2.2 enabled us to select several numerical methods that from a theoretical point of view seem to be suitable for the integration of the dynamics of multibody systems with unilateral and bilateral constraints. The chosen methods are then tested on several academic and industrial benchmarks and the results are addressed in Chapter 3.

Chapter 3

Comparison of several numerical schemes

The literature is abundant with numerical integration methods for ODEs and DAEs. The selection of the most suitable scheme must be carried out with many considerations in mind. Apart from obtaining a numerical approximation of the solution with a high accuracy and with the least numerical cost, these considerations include:

- Handling bilateral constraints in case of closed kinematic chains. These constraints must be enforced during the simulation, otherwise the obtained results will drift from the expected kinematics of the mechanism. When the discretized dynamics is solved with some numerical scheme that does not enforce the constraints, additional methods have to be performed to correct these constraints. However, this requires additional numerical effort. Therefore, the chosen method should enforce as best as possible the constraints.

- Coping with stiff systems. When an inappropriate scheme is used to integrate stiff dynamics, very tight time step sizes are required to perform the numerical integration, which slows down the simulation. Through several examples, we will demonstrate that explicit and half-explicit methods may fail in computing dynamics involving high frequencies.

- The knowledge of the contact forces is important in most practical applications. Therefore, the chosen scheme has to provide good approximations of the Lagrange multipliers. In addition, the checking of the status of the contacts involves the evaluation of the Lagrange multipliers at the impact times. Therefore, the schemes have to enable dense outputs for an accurate evaluation of these variables.

- Taking into account the change of the status of the contacts in the framework of event-driven schemes. Multistep methods are not considered in this chapter because of their lack of robustness in handling impacts. Indeed, the solution has to be restarted after each impact. This requires a step of initialization with another DAE integrator, involving many evaluations depending on the order of the chosen multistep method.

This chapter is concerned with the analysis of several numerical schemes dedicated to index-3, index-2 and index-1 DAEs in the context of an event–driven strategy: the generalized-α scheme for index-3 DAEs and index-2 DAEs , HEM5 and PHEM56 for index-2 DAEs, and RK4 and Runge-Kutta-Fehlberg for index-1 DAEs. The aforementioned schemes have been chosen based on the state of the art presented in Chapter 2. We compare these schemes in terms of efficiency, violation of the constraints and the way they handle stiff dynamics. Points of comparison include the drift of the constraints which is an important feature when we have to update the index sets of the active unilateral constraints, numerical efficiency and capability of handling stiffness. Comparisons are performed on several academic examples and also on numerous industrial benchmarks that contain several non-smooth additional effects, central in our study, such that the use of CAD description, with all the geometric imperfections related to the design, also introducing discontinuities in the contact surface description.

The work exposed in this chapter has been presented in a paper which has been submitted to the journal *Multibody System Dynamics*, at the time this document is being written.

3.1 Comparisons on academic examples

In this section, we compare the four schemes that have been described in the previous sections, namely, the HEM5, the PHEM56, the RKF and the generalized-α schemes. Several numerical tests with different values of the tolerance on the integration error will be performed on four mechanisms, using the same time step control strategy. The aim of these tests is to compare the selected solvers in terms of computational effort and drift of the constraints. Table 3.1 presents the parameters of time-step control strategy in (2.120). We set the tolerance on the drift of the constraints to a large value (10^{-2}) in order to prevent from a too severe requirement for certain schemes when applying the time-step control strategy.

Table 3.1: Parameters for time step control

Integration tolerance (tol)	Minimum time step	Tolerance for nonlinear equations solver for the implicit integrators	Maximum drift of g and $\dot{g}$	safety factor (s)
$[10^{-10}, 10^{-2}]$(*)	10^{-6}s	10^{-10}	10^{-2}	0.9

(*) We vary the value of tol to compare the computational effort and the drift of the constraints

In order to be able to evaluate the behavior of the constraints when using the aforementioned schemes, we make the choice of not correcting the constraints during the simulations of the academic examples. Therefore, no stabilization or projection methods were used. In addition, the time step size is only adapted to the truncation error, and not to the violation of the constraints. For the simulation of the academic examples, we used a Python implementation of HEM5, PHEM56, RKF and the generalized α-scheme (in its index-2 and index-3 versions). A few words about code implementation are discussed in Annex B.

3.1.1 Four-bar linkage

In this section we consider the four-bar linkage system, with three bars linked with revolute joints and the fourth one fixed to the ground, as described in Figure 3.1. This system is driven by a constant torque of value $\tau = 6\,N.m$ applied at the first joint, and is described by the vector of generalized coordinates $q = [\varphi_1,\, \varphi_2,\, \varphi_3]^T$. During the simulation of the motion of the bars, the lengths of the bars must be kept constant. For this aim, we write two constraints

$$\begin{cases} g_1(q) = l_1\cos(\varphi_1) + l_2\cos(\varphi_2) - l_3\cos(\varphi_3) - d = 0 \\ g_2(q) = l_1\sin(\varphi_1) + l_2\sin(\varphi_2) - l_3\sin(\varphi_3) = 0 \end{cases} \tag{3.1}$$

The equations of motion of the system can be written in the form of (1.18) with:

$$M(q) = \begin{pmatrix} I_1 + m_1(l_1/2)^2 + m_2{l_1}^2 & 0.5m_2l_1l_2\cos(\varphi_1 - \varphi_2) & 0 \\ 0.5m_2l_1l_2\cos(\varphi_1 - \varphi_2) & I_2 + m_1(l_2/2)^2 & 0 \\ 0 & 0 & I_3 + m_3(l_3/2)^2 \end{pmatrix}, \tag{3.2}$$

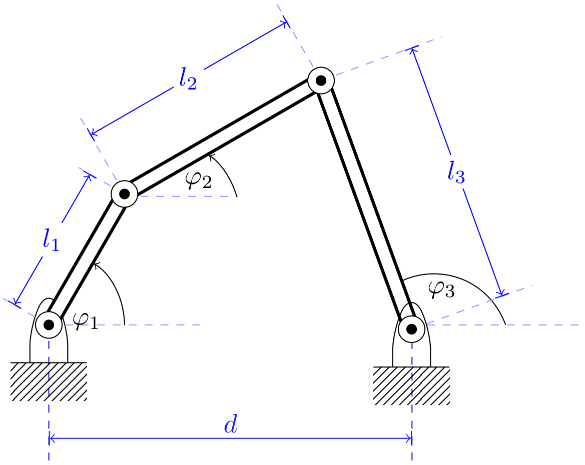


Figure 3.1: Four-bar linkage

with I_1 and I_2 being the moments of inertia of the rods,

$$F(q,\dot{q},t) = \begin{pmatrix} -(0.5m_1l_1 + m_2l_1)\mathbf{g}\cos(\varphi_1) + 0.5m_2l_1l_2\sin(\varphi_2-\varphi_1){\dot{\varphi}_2}^2 + \tau \\ -0.5m_2\mathbf{g}\ l_2\cos(\varphi_2) - 0.5m_2l_1l_2\sin(\varphi_2-\varphi_1){\dot{\varphi}_1}^2 \\ -0.5m_3\mathbf{g}\ l_3\cos(\varphi_3) \end{pmatrix}, \tag{3.3}$$

$$G(q) = \begin{pmatrix} -l_1\sin(\varphi_1) & -l_2\sin(\varphi_2) & l_3\sin(\varphi_3) \\ l_1\cos(\varphi_1) & l_2\cos(\varphi_2) & -l_3\cos(\varphi_3) \end{pmatrix}. \tag{3.4}$$

The values of the parameters are given in Table 3.2.

The initial conditions of position and velocity are compatible with the constraints at both position and velocity levels. We performed simulations with different values of the tolerance (tol) on the truncation error l_n, with the PHEM56 method, HEM5 solver, the Runge-Kutta-Fehlberg scheme and the generalized α-schemes (index-2 and index-3 forms). The results are summarized in Table 3.3.

Table 3.2: The four-bar-linkage characteristics

Initial conditions	$q_0 = [\frac{\pi}{3}, 0.24256387, \frac{2\pi}{3}]^T$ rad $v_0 = [0, 0, 0]^T$ m/s
Gravity **g**	9.81 m/s^2
Inertial properties	$m_1 = m_2 = m_3 = 1$ kg $I_1 = \frac{m_1 l_1^2}{12}$, $I_2 = \frac{m_2 l_2^2}{12}$, $I_3 = \frac{m_3 l_3^2}{12}$
Geometrical properties	$l_1 = 300$ mm $l_2 = 360.55$ mm $l_3 = 400$ mm $d = 700$ mm
Simulation characteristics	$t_{end} = 2$ s

Table 3.3: Four-bar linkage: average time step size, simulation time t_s, number of accepted steps, number of rejected steps and drift of the constraints for different tolerances.

method	tolerance	average h	t_s	accepted	rejected	$\Vert g_1 \Vert_\infty$	$\Vert g_2 \Vert_\infty$	$\Vert \dot{g}_1 \Vert_\infty$	$\Vert \dot{g}_2 \Vert_\infty$
HEM5	10^{-2}	0.088	0.44	50	17	$8.60\ 10^{-5}$	$7.92\ 10^{-5}$	$6.34\ 10^{-16}$	$7.55\ 10^{-16}$
	10^{-4}	0.038	0.88	86	27	$1.78\ 10^{-7}$	$3.47\ 10^{-7}$	$6.07\ 10^{-16}$	$6.65\ 10^{-16}$
	10^{-6}	0.017	2.13	215	56	$6.74\ 10^{-10}$	$3.06\ 10^{-9}$	$5.04\ 10^{-16}$	$9.24\ 10^{-16}$
	10^{-8}	0.006	5.04	557	105	$2.51\ 10^{-12}$	$1.53\ 10^{-9}$	$5.34\ 10^{-16}$	$1.07\ 10^{-15}$
	10^{-10}	0.002	12.47	1188	445	$2.33\ 10^{-15}$	$1.52\ 10^{-9}$	$7.13\ 10^{-16}$	$9.83\ 10^{-16}$
PHEM56	10^{-2}	0.090	0.21	45	7	$6.80\ 10^{-4}$	$6.97\ 10^{-4}$	$5.99\ 10^{-16}$	$5.37\ 10^{-16}$
	10^{-4}	0.054	0.34	64	20	$2.27\ 10^{-5}$	$1.21\ 10^{-5}$	$2.97\ 10^{-16}$	$4.23\ 10^{-16}$
	10^{-6}	0.023	0.77	128	49	$3.36\ 10^{-7}$	$7.08\ 10^{-8}$	$7.35\ 10^{-16}$	$6.34\ 10^{-16}$
	10^{-8}	0.011	1.77	291	83	$4.38\ 10^{-9}$	$1.86\ 10^{-9}$	$6.24\ 10^{-16}$	$7.17\ 10^{-15}$
	10^{-10}	0.004	4.31	699	269	$4.36\ 10^{-11}$	$1.52\ 10^{-9}$	$8.53\ 10^{-16}$	$8.65\ 10^{-16}$
RKF	10^{-2}	$9.97\ 10^{-3}$	2.54	109	10	$3.29\ 10^{-5}$	$1.76\ 10^{-5}$	$3.36\ 10^{-5}$	$1.86\ 10^{-5}$
	10^{-4}	0.016	2.05	99	31	$2.64\ 10^{-4}$	$1.07\ 10^{-4}$	$2.76\ 10^{-4}$	$1.04\ 10^{-4}$
	10^{-6}	0.008	4.60	214	72	$3.85\ 10^{-6}$	$2.74\ 10^{-7}$	$3.78\ 10^{-6}$	$3.66\ 10^{-7}$
	10^{-8}	0.003	12.49	504	106	$1.29\ 10^{-7}$	$2.37\ 10^{-9}$	$1.29\ 10^{-7}$	$1.12\ 10^{-8}$
	10^{-10}	0.001	29.30	1241	447	$3.04\ 10^{-9}$	$1.55\ 10^{-9}$	$2.95\ 10^{-9}$	$1.8\ 10^{-9}$
α-scheme, position level	10^{-2}	0.003	6.97	1146	451	$3.33\ 10^{-16}$	$1.66\ 10^{-16}$	$9.91\ 10^{-4}$	$9.97\ 10^{-4}$
	10^{-4}	0.002	7.06	1356	591	$3.33\ 10^{-16}$	$1.66\ 10^{-16}$	$9.46\ 10^{-4}$	$7.35\ 10^{-4}$
	10^{-6}	$4.64\ 10^{-4}$	24.87	6455	2253	$3.33\ 10^{-16}$	$2.22\ 10^{-16}$	$1.88\ 10^{-4}$	$1.08\ 10^{-4}$
	10^{-8}	$8.63\ 10^{-5}$	119.99	34607	10366	$3.33\ 10^{-16}$	$2.22\ 10^{-16}$	$2.68\ 10^{-5}$	$2.09\ 10^{-5}$
	10^{-9}	$1.81\ 10^{-5}$	565.44	164528	47231	$3.33\ 10^{-16}$	$2.22\ 10^{-16}$	$1.06\ 10^{-6}$	$3.04\ 10^{-4}$
α-scheme, velocity level	10^{-2}	0.042	2.43	66	26	$9.21\ 10^{-3}$	$4.93\ 10^{-3}$	$3.05\ 10^{-12}$	$3.86\ 10^{-12}$
	10^{-4}	0.013	4.74	235	140	$4.75\ 10^{-4}$	$2.44\ 10^{-4}$	$8.67\ 10^{-13}$	$6.32\ 10^{-13}$
	10^{-6}	0.003	4.38	1067	448	$2.18\ 10^{-5}$	$1.03\ 10^{-5}$	$2.08\ 10^{-13}$	$1.56\ 10^{-13}$
	10^{-8}	$6.07\ 10^{-4}$	15.77	4937	1480	$1.00\ 10^{-6}$	$4.65\ 10^{-7}$	$2.22\ 10^{-14}$	$1.96\ 10^{-14}$
	10^{-10}	$1.31\ 10^{-4}$	66.36	22897	5771	$4.65\ 10^{-8}$	$2.28\ 10^{-8}$	$1.49\ 10^{-14}$	$8.44\ 10^{-15}$

It is worth noting that a step is rejected if the integration error is greater than the prescribed tolerance or if the drift exceeds the user defined tolerance. As expected, for HEM5 and PHEM56, the drift of the constraints at the velocity level (columns 9 and 10 of Table 3.3) is at the same order of magnitude (10^{-16}) for all the tolerances. This value corresponds to the machine accuracy.

As shown in column 3 of Table 3.3, the PHEM56 scheme uses larger time steps than those used by HEM5. With less stages, PHEM56 is 2 to 4 times more efficient than the HEM5 scheme. The velocity constraints are maintained at the machine accuracy, and the position constraints are maintained at reasonable levels for both schemes. However, for the same value of the tolerance, HEM5 has better performance dealing with the drift as shown in columns 7 and 8. This is due to the fact that PHEM56 uses larger time step sizes than the HEM5, and we observe that for equivalent time step sizes, HEM5 and PHEM56 reduce the position constraints to the same level. The drift of the constraints with the Runge-Kutta-Fehlberg method reaches acceptable levels for tolerances on the truncation error higher than 10^{-6}, but this drift is much higher for larger tolerances, unlike HEM5 and PHEM56.

With the α-scheme at the position level, the drift of g_1 and g_2 is at the order of magnitude of the machine accuracy. It is worth noting that when considering the velocity constraints instead of the position constraints, the computational effort, measured by the computation time, is divided by 3 or 4. Nevertheless, the violation of the velocity constraints is never reduced to the machine accuracy for the index-2 form of the generalized-α scheme, unlike the HEM5 and PHEM56. This is due to the fact that the generalized-α scheme has a lower order (2) than the order of the other aforementioned methods which is 5.

For this mechanism, we can say that overall the HEM5 and the PHEM56 solvers satisfy the user defined tolerances on the integration error and on the drift of the constraints with a lower cost than in the case of the Runge-Kutta-Fehlberg method or the generalized α-scheme. The difference with the other solvers is even more remarkable for tight values of the tolerance for which HEM5 and PHEM56 are cheaper than the Runge-Kutta-Fehlberg, about 15 times cheaper than the generalized-α scheme with position constraints, and almost 3 times cheaper than the generalized-α scheme with velocity constraints. We know however that the generalized-α scheme is a low order scheme that is well suited for flexible multibody dynamics where high frequency non-linear dynamics can render the integration difficult.

When the user-defined tolerance is equal to 10^{-10}, the proportion of rejected steps over the accepted steps is higher than the case of the other tolerances, for all the solvers. This is due to the fact that the safety factor 0.9 is too high and must be reduced for very tight tolerances.

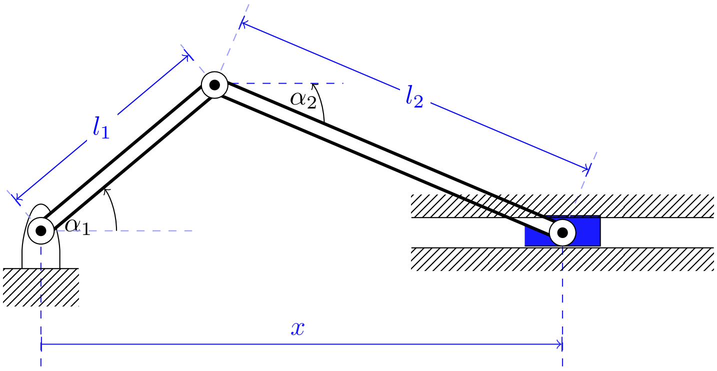


Figure 3.2: The slider-crank system

3.1.2 The slider-crank system

The system and its characteristics are depicted in Figure 3.2. Let m_1, m_2 and l_1, l_2 denote the masses and the lengths of the rods, m_3 the mass of the slider, and let $\mathbf{g}$ denote the gravity acceleration. The system is described with the vector of generalized coordinates $q = [\alpha_1,\ \alpha_2]^T$. The revolute joints lead to a constraint which may be written as

$$g(q) = l_1 \sin(\alpha_1) + l_2 \sin(\alpha_2) = 0. \tag{3.5}$$

The equations of motion can be written in the form of (1.18) with

$$M(q)\ddot{q} = F(q, \dot{q}, t) + G^T(q)\lambda, \tag{3.6}$$

where :

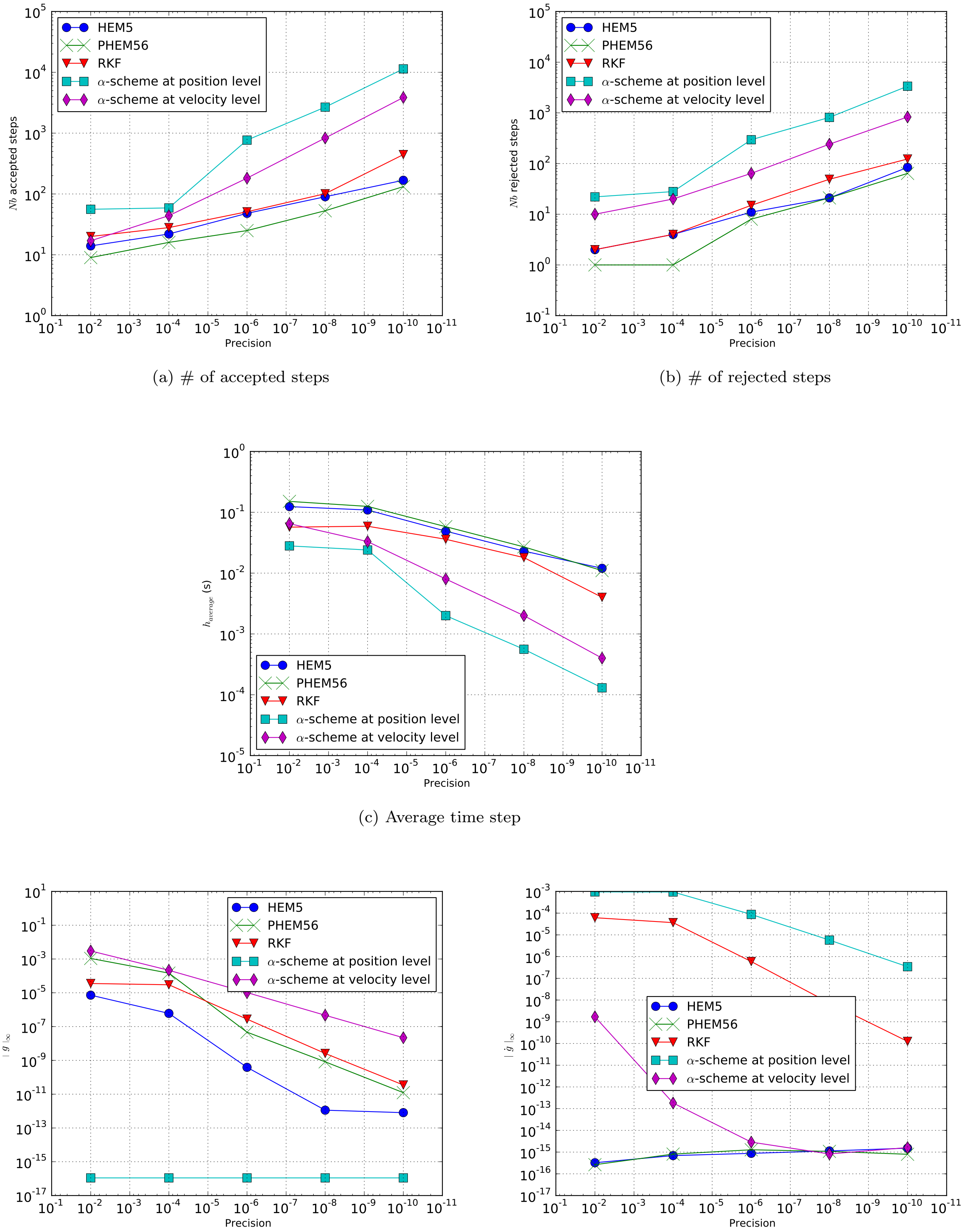
$$M(q) = \begin{pmatrix} I_1 + (0.25m_1 + m_2 + m_3)l_1^2 & (0.5m_2 + m_3)l_1 l_2 \cos(\alpha_2 - \alpha_1) \\ (0.5m_2 + m_3)l_1 l_2 \cos(\alpha_2 - \alpha_1) & I_2 + (0.5m_2 + m_3){l_2}^2 \end{pmatrix}, \tag{3.7}$$

with m_1, m_2 and m_3 being the masses of the crank, the connecting rod and the slider respectively.

$$F(q, \dot{q}, t) = \begin{pmatrix} (0.5m_2 + m_3)l_1 l_2 \sin(\alpha_2 - \alpha_1){\dot{\alpha}_2}^2 - (0.5m_1 + m_2 + m_3)l_1 \mathbf{g} \cos(\alpha_1) \\ -(0.5m_2 + m_3)l_1 l_2 \sin(\alpha_2 - \alpha_1){\dot{\alpha}_1}^2 - (0.5m_2 + m_3)l_2 \mathbf{g} \cos(\alpha_2) \end{pmatrix}, \tag{3.8}$$

$$G(q) = \begin{pmatrix} l_1 \cos(\alpha_1) & l_2 \cos(\alpha_2) \end{pmatrix}. \tag{3.9}$$

The parameters of the simulation are presented in Table 3.4. We start the simulation with initial conditions that are compatible with the constraint manifold $\{(q_0, v_0)|\, g_1(q_0) = 0 \text{ and } \dot{g}_1(q_0) = G(q_0)v_0 = 0\}$. In Fig. 3.3 we present the average time step size and the maximum value of the drift of the constraints at both position and velocity levels as well as the computational work, as functions of the tolerance (precision) on the integration error.



(a) # of accepted steps

(b) # of rejected steps

(c) Average time step

(d) maximum of violation of the position constraint

(e) maximum of violation of the velocity constraint

Figure 3.3: Slider crank: simulation characteristics

Table 3.4: The slider crank characteristics

Initial conditions	$q_0 = [0,0]^T$ rad $v_0 = [4,-2]^T$ rad/s
Gravity $\mathbf{g}$	9.81 m/s^2
Inertial properties	$m_1 = m_2 = m_3 = 1$ kg $I_1 = \frac{m_1 l_1^2}{12}$, $I_2 = \frac{m_2 l_2^2}{12}$
Geometrical properties	$l_1 = 1$ m $l_2 = 1$ m
Simulation characteristics	$t_{end} = 1$ s

With the HEM5 solver, the drift of the constraints at the velocity level is at the order of magnitude of the machine accuracy (Fig.3.3e). The drift of the position constraints is at very acceptable levels for all the values of tolerances on the integration error (Fig.3.3d). The PHEM56 scheme is about 3 times less time consuming than the HEM5 scheme. Once again, for the same user defined tolerance, we observe larger drift of the position constraints with PHEM56 than with HEM5. With the Runge-Kutta-Fehlberg method, the tolerance at the integration error is met with large time step sizes as shown in Fig.3.3c but the drift of the constraints is much higher than with HEM5 or PHEM56.

For the slider crank mechanism, the drift of g (Fig.3.3d) using the α-scheme with position constraints is at the order of magnitude of the machine accuracy. However, the accuracy at the velocity level is quite low. Because of this drift, the detection of the closed contacts defined by the set I_2 in section 3 is affected if the drift is greater than the tolerance used to define I_2. When using the velocity-level α-scheme, the drift is large for tolerances larger than 10^{-6} (Fig.3.3d), and we can notice in Fig.3.5b that the numerical effort is 2 up to 3 times smaller than for the position-level α-scheme.

From the computational effort point of view, PHEM56 and HEM5 are less time-consuming than the Runge-Kutta-Fehlberg scheme, five times less time-consuming than the α-scheme with the position constraints and two times cheaper than the α-scheme with the velocity constraints. Overall, for this example, we draw the same conclusions as for the four-bar mechanism either concerning the computational effort or concerning the enforcement of the constraints.

3.1.3 The slider-crank system with a flexible connecting rod

This system is illustrated in Figure 3.4. This system has been studied in [120] in which the reader

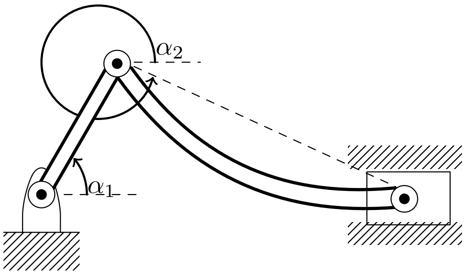


Figure 3.4: The slider-crank with a flexible connecting rod

will find more details about the modeling and the initial conditions as well as an illustration of the mechanism. We performed simulations with different values of the tolerance on the truncation error, with the HEM5 solver, the PHEM56 solver, the Runge-Kutta-Fehlberg scheme and the generalized α-scheme. We started simulations with a safety factor of 0.9, and we noticed that the number of rejected steps was high, so we decreased its value to 0.85. Table 3.5 summarizes the results.

Table 3.5: Flexible slider crank: average time step size, simulation time, number of accepted steps, number of rejected steps and drift of the constraints for different tolerances.

method	tolerance	average h	t_s	accepted	rejected	$\Vert g_1 \Vert_\infty$	$\Vert g_2 \Vert_\infty$	$\Vert g_3 \Vert_\infty$	$\Vert \dot{g}_2 \Vert_\infty$	$\Vert \dot{g}_2 \Vert_\infty$	$\Vert \dot{g}_3 \Vert_\infty$
HEM5	10^{-2}	$1.17\ 10^{-4}$	19.76	961	391	$4.69\ 10^{-14}$	$2.03\ 10^{-10}$	$8.08\ 10^{-14}$	$4.10\ 10^{-15}$	$3.88\ 10^{-15}$	0.0
	10^{-4}	$1.06\ 10^{-4}$	22.84	1066	445	$2.87\ 10^{-7}$	$8.18\ 10^{-8}$	$4.44\ 10^{-14}$	$4.10\ 10^{-15}$	$5.04\ 10^{-15}$	0.0
	10^{-6}	$1.55\ 10^{-4}$	13.97	737	240	$3.71\ 10^{-11}$	$2.05\ 10^{-10}$	$5.32\ 10^{-14}$	$4.08\ 10^{-15}$	$3.88\ 10^{-15}$	0.0
	10^{-8}	$1.06\ 10^{-4}$	21.32	1055	445	$7.30\ 10^{-13}$	$2.03\ 10^{-10}$	$2.13\ 10^{-14}$	$4.08\ 10^{-15}$	$3.48\ 10^{-15}$	0.0
	10^{-10}	$3.20\ 10^{-5}$	70.84	3522	1328	$7.65\ 10^{-15}$	$2.03\ 10^{-10}$	$3.91\ 10^{-14}$	$4.10\ 10^{-15}$	$3.78\ 10^{-15}$	0.0
PHEM56	10^{-2}	$1.03\ 10^{-4}$	13.08	1131	341	$4.02\ 10^{-3}$	$2.11\ 10^{-3}$	$1.42\ 10^{-14}$	$8.00\ 10^{-13}$	$3.14\ 10^{-12}$	0.0
	10^{-4}	$6.29\ 10^{-5}$	19.65	1712	452	$7.06\ 10^{-5}$	$8.01\ 10^{-5}$	$2.13\ 10^{-14}$	$8.17\ 10^{-14}$	$1.78\ 10^{-13}$	0.0
	10^{-6}	$6.19\ 10^{-5}$	20.01	1752	488	$2.29\ 10^{-14}$	$2.03\ 10^{-10}$	$5.50\ 10^{-14}$	$4.08\ 10^{-15}$	$3.92\ 10^{-15}$	0.0
	10^{-8}	$6.20\ 10^{-5}$	20.71	1733	385	$5.86\ 10^{-11}$	$2.12\ 10^{-10}$	$5.68\ 10^{-14}$	$4.02\ 10^{-15}$	$3.80\ 10^{-15}$	0.0
	10^{-10}	$5.77\ 10^{-5}$	21.81	1861	535	$4.31\ 10^{-13}$	$2.03\ 10^{-10}$	$3.37\ 10^{-14}$	$3.98\ 10^{-15}$	$4.39\ 10^{-15}$	0.0
RKF	10^{-2}	$5.67\ 10^{-5}$	58.98	2005	682	$1.25\ 10^{-9}$	$2.10\ 10^{-10}$	$8.08\ 10^{-13}$	$5.76\ 10^{-8}$	$9.87\ 10^{-9}$	0.0
	10^{-4}	$5.06\ 10^{-5}$	85.57	2245	751	$1.84\ 10^{-8}$	$2.34\ 10^{-8}$	$8.17\ 10^{-14}$	$9.80\ 10^{-7}$	$6.88\ 10^{-7}$	0.0
	10^{-6}	$4.12\ 10^{-5}$	104.94	2769	906	$1.18\ 10^{-9}$	$2.17\ 10^{-10}$	$5.06\ 10^{-14}$	$1.20\ 10^{-8}$	$2.83\ 10^{-10}$	0.0
	10^{-8}	$7.27\ 10^{-5}$	234.03	5428	2517	$1.19\ 10^{-9}$	$2.03\ 10^{-10}$	$7.28\ 10^{-14}$	$1.19\ 10^{-8}$	$1.78\ 10^{-11}$	0.0
	10^{-10}	$6.66\ 10^{-5}$	549.06	12989	6659	$1.19\ 10^{-9}$	$2.04\ 10^{-10}$	$1.42\ 10^{-13}$	$1.19\ 10^{-8}$	$9.62\ 10^{-13}$	0.0
α-scheme, position level	10^{-2}	$3.06\ 10^{-4}$	3.60	382	78	$4.05\ 10^{-17}$	$1.33\ 10^{-16}$	0.0	$8.83\ 10^{-5}$	$9.57\ 10^{-5}$	$2.39\ 10^{-9}$
	10^{-4}	$3.28\ 10^{-4}$	3.85	338	10	$4.05\ 10^{-17}$	$1.33\ 10^{-16}$	0.0	$8.83\ 10^{-5}$	$9.57\ 10^{-5}$	$2.39\ 10^{-9}$
	10^{-6}	$4.37\ 10^{-4}$	33.03	2863	580	$4.02\ 10^{-17}$	$1.33\ 10^{-16}$	0.0	$9.47\ 10^{-5}$	$9.97\ 10^{-5}$	$2.51\ 10^{-9}$
	10^{-8}	$9.47\ 10^{-6}$	86.61	11281	3635	$4.09\ 10^{-17}$	$1.34\ 10^{-16}$	0.0	$8.88\ 10^{-5}$	$8.57\ 10^{-5}$	$9.76\ 10^{-8}$
	10^{-10}	$2.20\ 10^{-6}$	369.03	15921	30170	$4.11\ 10^{-17}$	$1.34\ 10^{-16}$	0.0	$8.47\ 10^{-6}$	$4.06\ 10^{-5}$	$7.11\ 10^{-7}$
α-scheme, velocity level	10^{-2}	$2.68\ 10^{-3}$	0.67	42	7	$3.09\ 10^{-6}$	$5.75\ 10^{-6}$	$1.49\ 10^{-13}$	$2.25\ 10^{-14}$	$1.49\ 10^{-14}$	0.0
	10^{-4}	$8.76\ 10^{-4}$	2.04	127	50	$3.09\ 10^{-6}$	$5.75\ 10^{-6}$	$1.49\ 10^{-13}$	$2.25\ 10^{-14}$	$1.49\ 10^{-14}$	0.0
	10^{-6}	$2.79\ 10^{-4}$	7.38	536	213	$1.37\ 10^{-5}$	$2.10\ 10^{-5}$	$3.91\ 10^{-14}$	$1.77\ 10^{-14}$	$8.38\ 10^{-14}$	0.0
	10^{-8}	$6.06\ 10^{-5}$	28.95	2475	741	$6.21\ 10^{-7}$	$9.78\ 10^{-7}$	$3.02\ 10^{-14}$	$7.47\ 10^{-15}$	$8.19\ 10^{-14}$	0.0
	10^{-10}	$1.31\ 10^{-5}$	119.77	11484	3294	$2.87\ 10^{-8}$	$4.56\ 10^{-8}$	$1.10\ 10^{-13}$	$1.03\ 10^{-14}$	$1.76\ 10^{-14}$	0.0

The position constraints g_1, g_2 and g_3, are defined in [120]. This mechanism shows clearly the limitations of (half-)explicit or partitioned schemes against implicit ones when stiff or highly stiff mechanisms are considered. For tolerances larger than (or equal to) 10^{-4}, the generalized α-scheme at the position level is less time-consuming than HEM5, PHEM56 and RKF (column 4 of Table 3.5). On the other hand, for tolerances larger than or equal to 10^{-6}, the α-scheme with the velocity constraints outperforms the other schemes. For tighter tolerances, HEM5 and PHEM56 are more efficient than the α-scheme. For tolerances between 10^{-2} and 10^{-8}, HEM5 and PHEM56 have equivalent simulation times, even if HEM5 uses larger time steps. This can be explained by the fact that HEM5 has 2 more stages than PHEM56. In this example again, dealing with the velocity constraints instead of the position constraints enables to reduce drastically the numerical effort. Indeed, the generalized α-scheme with velocity constraints is almost 5 times cheaper than the one with position constraints. It is also worth noting that the RKF scheme is the most time consuming. Concerning the drift, since all the schemes are dealing with quite tight time steps ($\approx 10^{-5}$s), the drift is kept at acceptable levels for all the values of the tolerances, for all the methods.

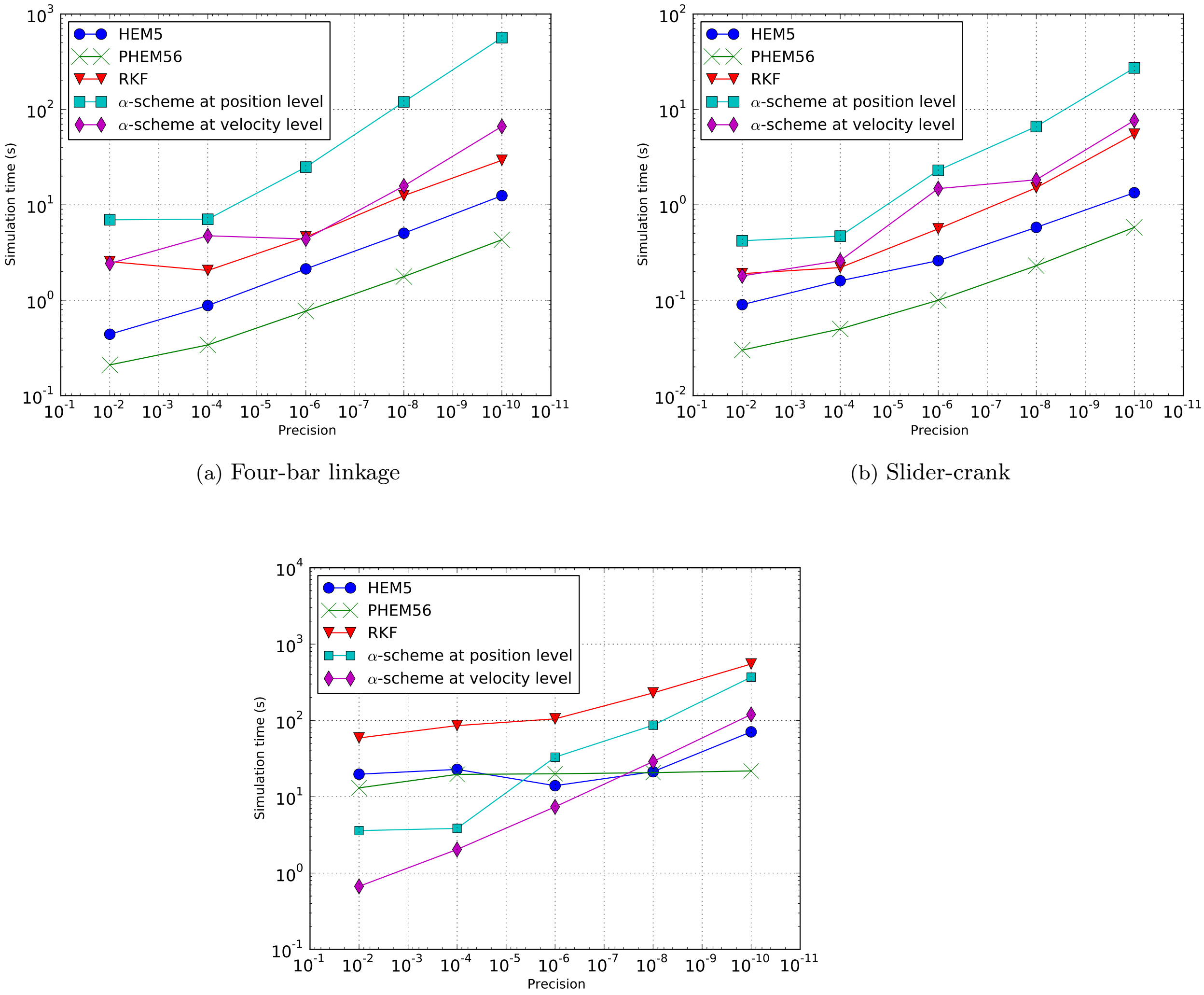
3.1.4 Work-precision diagrams

We present the precision-work diagrams for the three mechanisms in Figures 3.5a, 3.5b and 3.5c. In view of the results presented in all the tables that summarize the computational effort required by PHEM56, HEM5, RKF and the α-scheme, we can say that overall the PHEM56 and HEM5 solvers are the most computationally efficient for rigid mechanisms since they meet the high tolerances on the integration error with a cheaper cost than the other methods. Furthermore, for low tolerances and thus small time step sizes, PHEM56 and HEM5 solver hold the constraints at both position and velocity levels at very acceptable levels compared to the other solvers. Conversely, the generalized-α scheme is the most computationally efficient when stiff mechanical systems and large tolerances are considered (see Figure 3.5c).

3.1.5 Preliminary conclusions on the academic examples

Based on the results presented in the foregoing tables and in Figure 3.5, we could make the following conclusions:

- For rigid systems with bilateral constraints, PHEM56 and HEM5 are the most efficient to save the computational effort while enforcing the velocity constraints at the machine accuracy and maintaining the drift at very low values for a wide range of the precision on the integration error.



(a) Four-bar linkage

(b) Slider-crank

(c) Flexible slider-crank

Figure 3.5: Work/Precision diagrams for the four-bar linkage, the slider-crank, and the flexible slider-crank

Both schemes enforce the constraints at equivalent levels for equivalent time step sizes. With two less stages, the PHEM56 is less time consuming than the HEM5. Given its high order (equal to 4), the RKF scheme gives good results concerning the computation time and the enforcement of the constraints for tolerances larger than or equal to 10^{-6}. Being initially designed for simulating the structural dynamics, the generalized-α scheme is less computationally efficient for this kind of systems. However, it is important to stress that when using the α-scheme with velocity constraints instead of the position constraints, we divide the numerical effort by 3 to 5. Let us also stress that the numerical solver incorporated in the event-driven scheme has to enforce the position and velocity constraints for a good evaluation of the index sets presented in Section 2.1.1.1. Indeed, the correct computation of these index sets, and hence the robustness of the event–driven scheme, is very dependent on the correct computation of the constraint in terms of drift and contact forces. These properties of the numerical time integration scheme are therefore crucial for the development of a robust event–driven solver.

- For stiff mechanisms, implicit schemes outperform (half-)explicit or partitioned schemes for large tolerances. Yet, for tight tolerances, the HEM5 and the PHEM56 schemes prove to be less time consuming than the implicit scheme. For the studied flexible slider crank, the RKF scheme was the more time consuming.

- From the simulations using HEM5, PHEM56 and the generalized-α with velocity constraints, it follows that a better stabilization of the constraints is obtained with a discretization of the constraints on the velocity level. Indeed, these schemes enforce the velocity constraints at the machine accuracy while maintaining the drift at very low levels. HEM5 and PHEM56 obviously outperform the generalized-α scheme with velocity constraints due to their higher order (equal to 5).

3.2 Industrial benchmarks

In this section, the above algorithms are compared on benchmark examples from the ANSYS Rigid Body Dynamics, 16.0 test library (<`http://www.ansys.com/Products/`). Let us mention that this software describes the dynamics using minimal relative coordinates. Newton-Euler formalism is used to describe the dynamics, with a nonsmooth formulation of the contact. The dynamics is then simulated using the above event-driven scheme. The goal is to confirm the preliminary conclusions on real examples that come from industrial applications with non ideal geometries. In order to get some

Table 3.6: Characteristics of the sets

Set	Max # of DOF	Min # of DOF	Max # of joints	Min # of joints	Max # of pairs of contacts
1	19	1	38	2	-
2	18	2	22	3	-
3	11	2	15	2	7
4	31	5	15	4	1

insight into the performance of schemes on large test sets, we use performance profiles. They allow us to draw some general conclusions that are valid in a statistical manner.

3.2.1 Description of the mechanisms

The mechanisms we study are split into four sets described in Table 3.6. The first test set comprises simple problems with a number of degrees of freedom between 1 and 19 and a number of joints between 2 and 38. This set will enable us to test the kinematics and the dynamics with a wide range of joints, without unilateral contacts and impact. The second set comprises tests with large simulation time; the mechanisms have a large number of degrees of freedom and a large number of joints, and no impacts. The third set comprises mechanisms with impacts. It will help to validate the strategy of solving the impact at the impact time and resuming the computation of the dynamics. The fourth set contains systems with flexible beams, with square or rectangular sections; the dynamics of these bodies is simulated under different types of loads and with impacts in some cases. This set contains 9 systems with beam models, the number of degrees of freedom varies between 5 and 31 and the number of kinematic joints varies between 4 and 14. Illustrations of some mechanisms from each set are depicted in Figures 3.6, 3.7, 3.8, and 3.9.

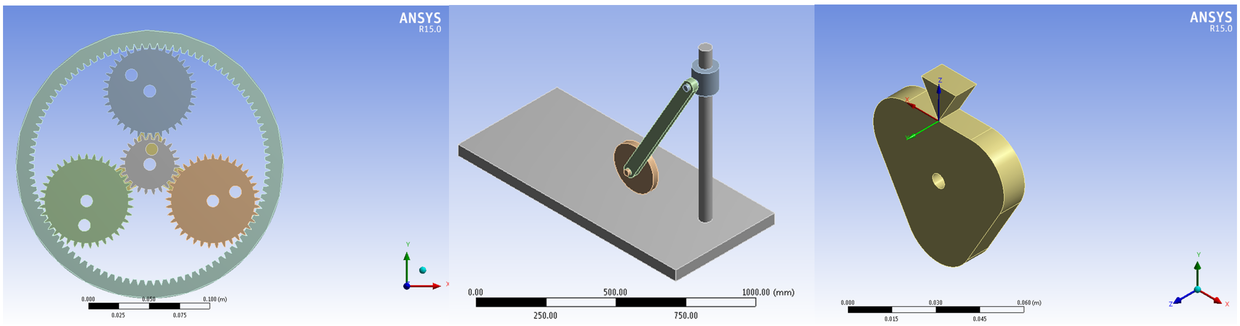
3.2.2 Performance profile

The following metric is introduced in [38]. Let us consider a set P of n_p problems, and a set S of n_s solvers (or numerical methods). For each problem p and solver s we define a performance criteria:

$$t_{p,s} = \text{computing time required for solver } s \text{ to solve problem } p\,. \tag{3.10}$$

We define the performance ratio as

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s},\ s \in S\}}\,. \tag{3.11}$$



(a) Epicyclic gear train

(b) Rotating disk attached to a rod with a collar

(c) A cam mechanism

Figure 3.6: Examples from the first set

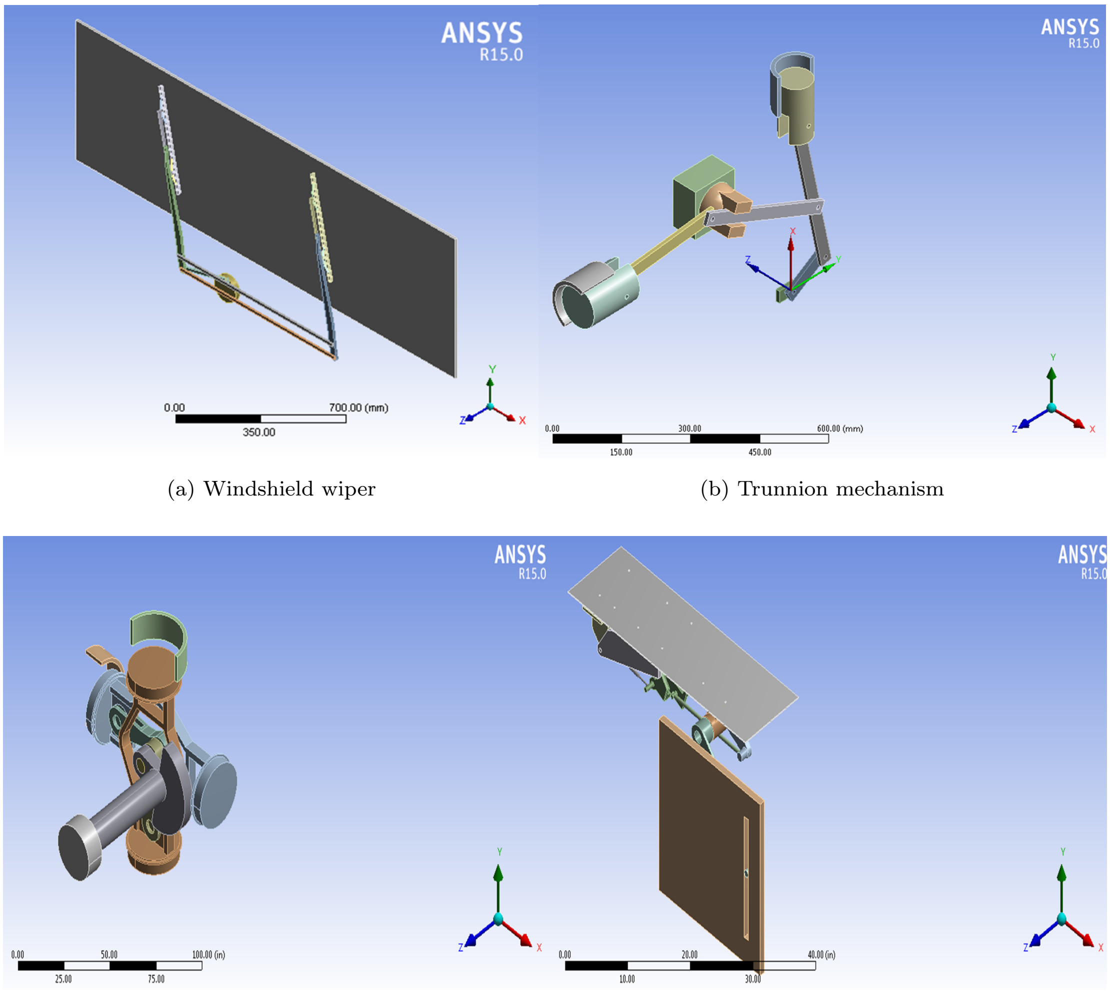
Then, we define the probability $\rho_s(\tau)$ for a solver $s \in S$ that a performance ratio $r_{p,s}$ is below a factor $\tau \in \mathbb{R}$:

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P,\ r_{p,s} \leq \tau\} \leq 1. \tag{3.12}$$

Thus, $\rho_s(\tau)$ is the distribution function for the performance ratio. It is worth noting that $\rho_s(1)$ represents the probability that the solver s beats the other solvers, and $\rho_s(\infty)$ characterizes the robustness of the method. The higher ρ_s is, the better the method is. The term *performance profile* will be used in the sequel for this concept. This concept will be made clearer when used for the analysis of the simulation results presented in section 3.2.3.

3.2.3 Simulations results

For each set, we ran simulations on the mechanisms using HEM5, RK4 and the generalized-α methods. According to the conclusions drawn in the Section 3.1, we choose to use the generalized-α in its index-2 DAE form, because it proved to better stabilize at both the position and velocity levels and to use larger time steps than its index-3 DAE form. At the time this document has been written, the PHEM56 scheme has not yet been implemented in the Ansys solver, therefore no results on the industrial benchmarks are available for this method. The tolerance on the integration error is set to 10^{-4} for all the aforementioned schemes. For each simulation, we collected some simulation data, namely the simulation time, the average time step size and the number of steps performed by each numerical scheme. Figures 3.10, 3.11, 3.12, and 3.13 show the average time step and the number of steps for each problem and for each solver as well as the distribution function $\rho_s(\tau)$ for each solver, with different values of τ between 1 and 2.6. For the first set, we observe that RK4 reaches the



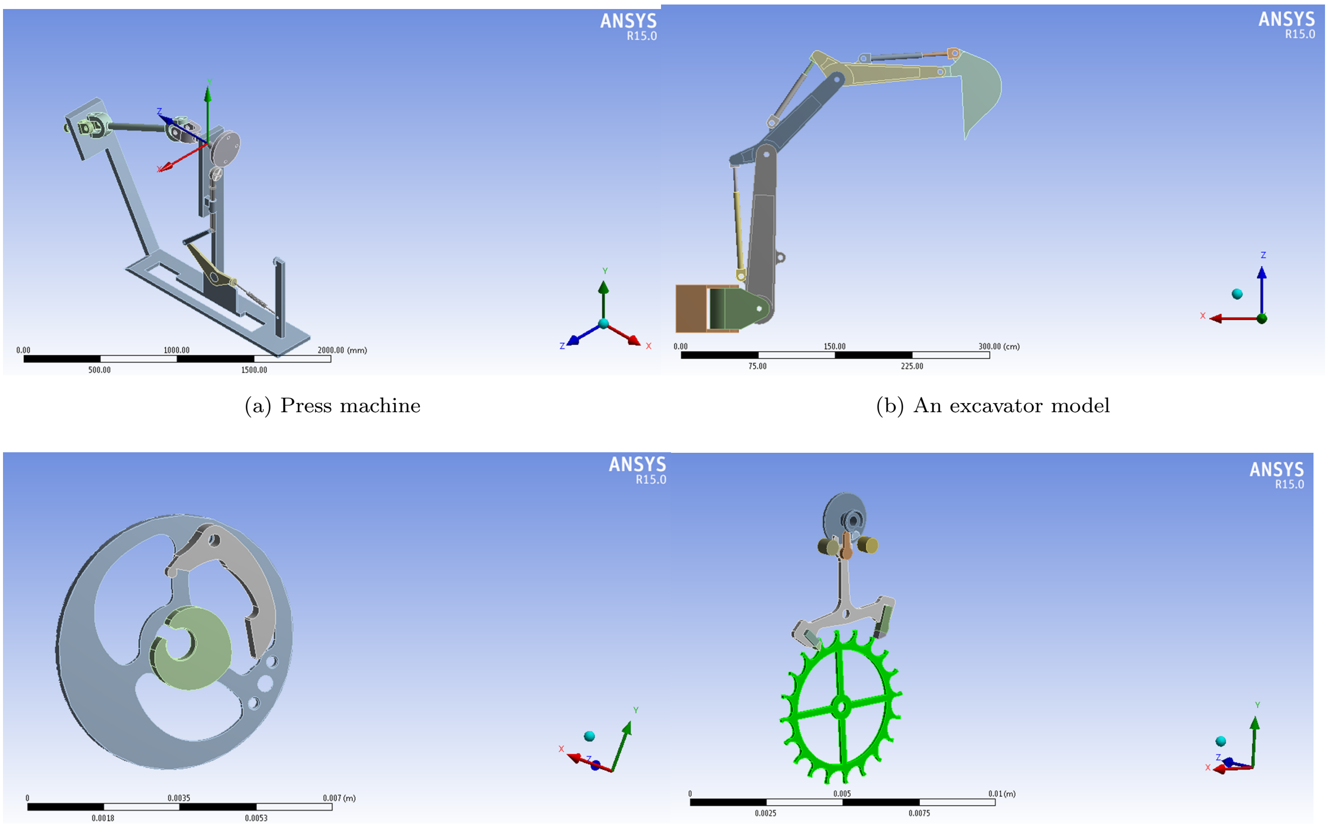
(a) Windshield wiper

(b) Trunnion mechanism

(c) Air piston

(d) Subway door

Figure 3.7: Examples from the second set



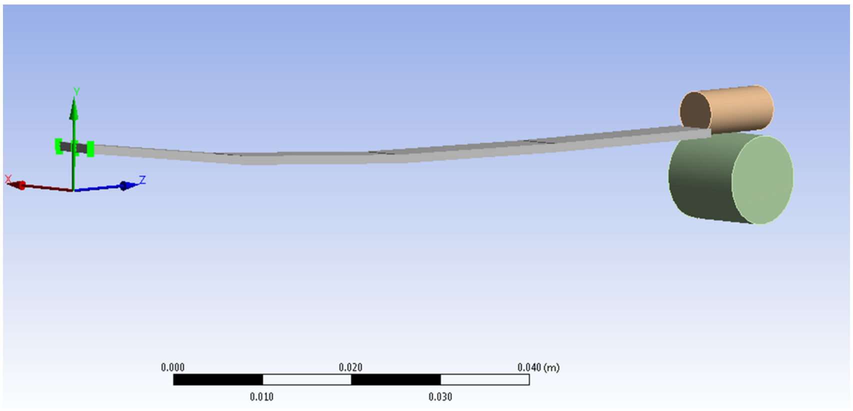
(a) Press machine

(b) An excavator model

(c) A watch mechanism sub-assembly

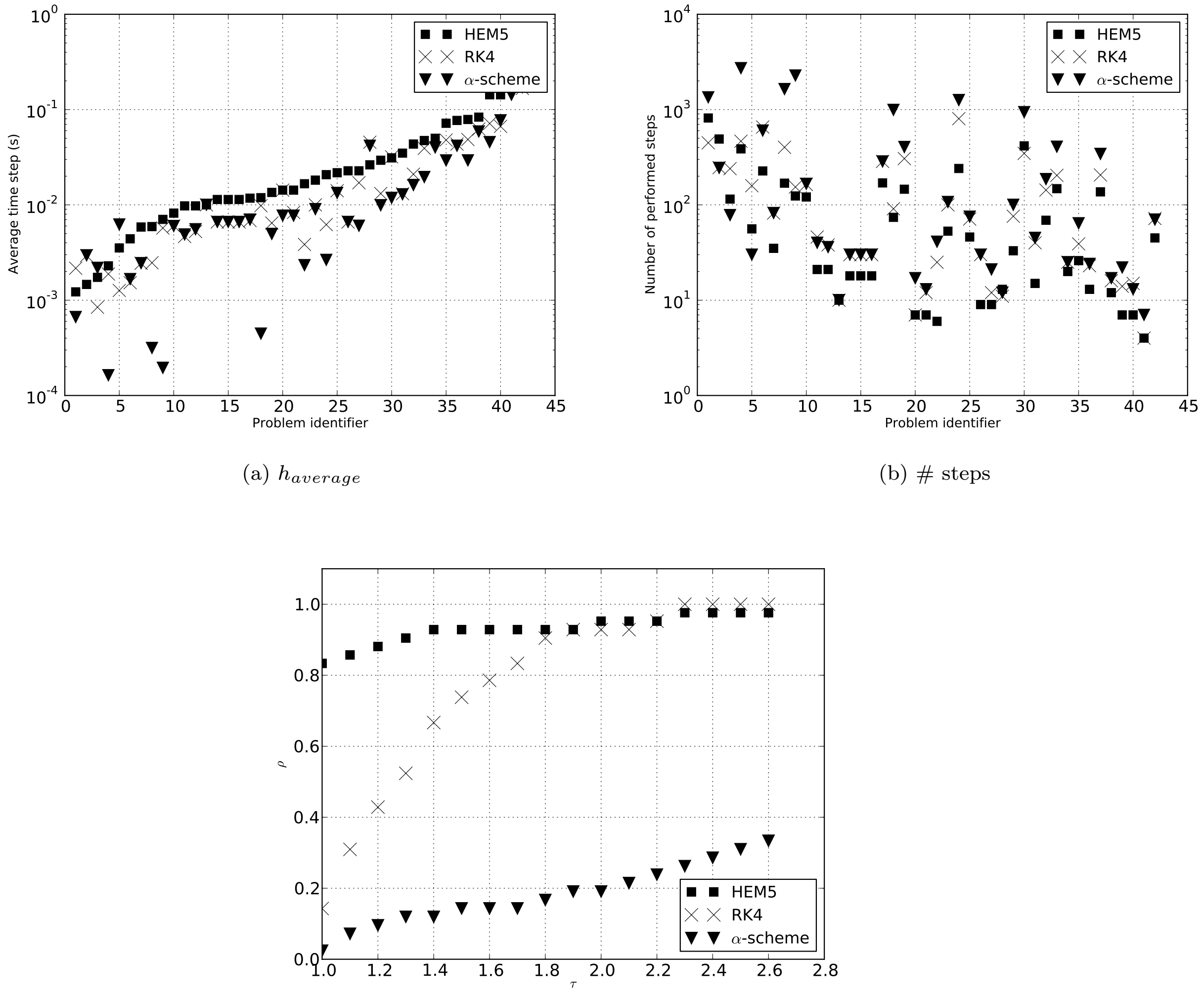
(d) Escapement of a mechanical watch

Figure 3.8: Examples from the third set



(a) Beam under gravity in contact with a cylinder

Figure 3.9: Example from the fourth set



(a) $h_{average}$

(b) # steps

(c) performance profile

Figure 3.10: Average time step, number of iterations and performance profile of the first set

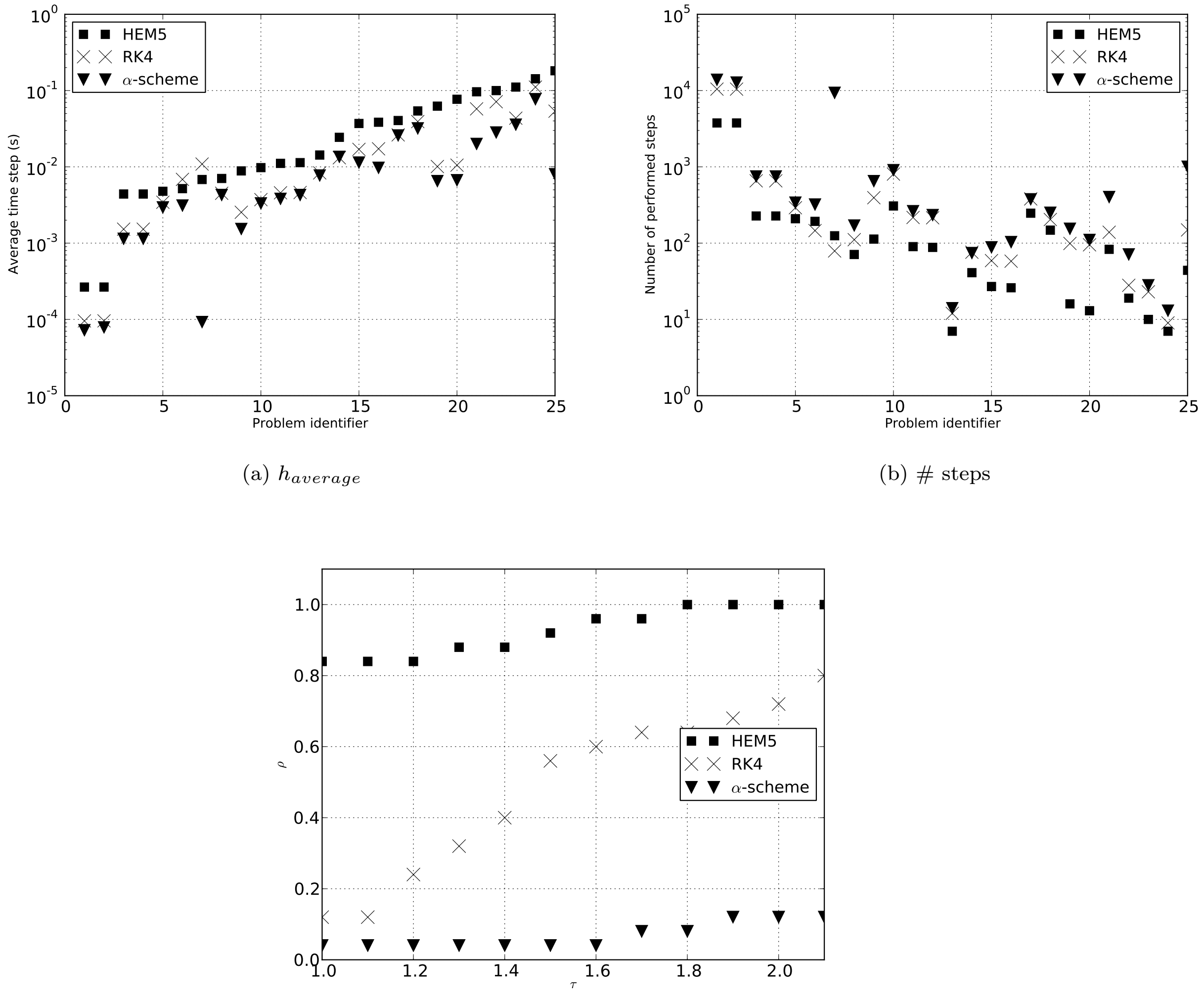
probability of 1 contrary to the other solvers, which means that it is more robust to solve this kind of problems. This can be explained by the fact that this set contains "simple" problems for which using an 8-stage scheme (HEM5) is very time-consuming. The performance profiles of Figures 3.11 and 3.12 show that when the dynamics is more challenging (large number of degrees of freedom, more non linear terms), the HEM5 scheme proves its robustness over the other methods.

From the graphics of Figures 3.10, 3.11 and 3.12, we can see that the generalized-α and RK4 need smaller time step sizes and larger number of steps to solve the dynamics than HEM5. Furthermore, HEM5 stands out for solving more problems than RK4 or the α-scheme for the time ratio τ, even if the problems involve large number of degrees of freedom and constraints. Table 3.7 shows a comparison between the three schemes for $\tau = 1.2$. $\rho_s(1.2)$ defines the percentage of problems solved by a solver s in a ratio of 1.2 with respect to the fastest solver.

Table 3.7: Comparison of ρ between HEM5, RK4 and generalized-α for $\tau = 1.2$, for the various test sets

Set number	$\rho_{HEM5}(1.2)$	$\rho_{RK4}(1.2)$	$\rho_{\alpha-scheme}(1.2)$
1	0.88	0.43	0.01
2	0.84	0.24	0.04
3	0.95	0.48	0.05
4	0.89	0.44	0.11

Most of the tests of the fourth set involve systems with linear beams which dynamics is not very stiff, which explains the fact that the implicit scheme is not outperforming the two other schemes. Indeed, we observe in Fig. 3.13 that RK4 and the generalized-α use equivalent time steps sizes that are even slightly smaller than those used by HEM5; and need an equivalent number of performed steps. However, the α-scheme is more time consuming than the two other schemes, probably due to the computation of the tangent matrices and the resolution of linear systems at each Newton iteration. Nevertheless, we can see that the problem whose identifier is 3 does not follow the pattern. Indeed, unlike the other test cases, its dynamics is actually very stiff and non linear. Let us study this particular example more deeply in the sequel. It consists of an eccentrically rotating beam with 31 degrees of freedom, illustrated in Fig. 3.14. It is subjected to a constant rotation velocity constraint. We ran simulations on this particular system for different values of the integration error tolerance and we collected results that we present in Table 3.8. We can see that the α-scheme uses time step sizes that are 10 to 100 times larger than those used for HEM5 and RK4. It is also 50 to 90 times less

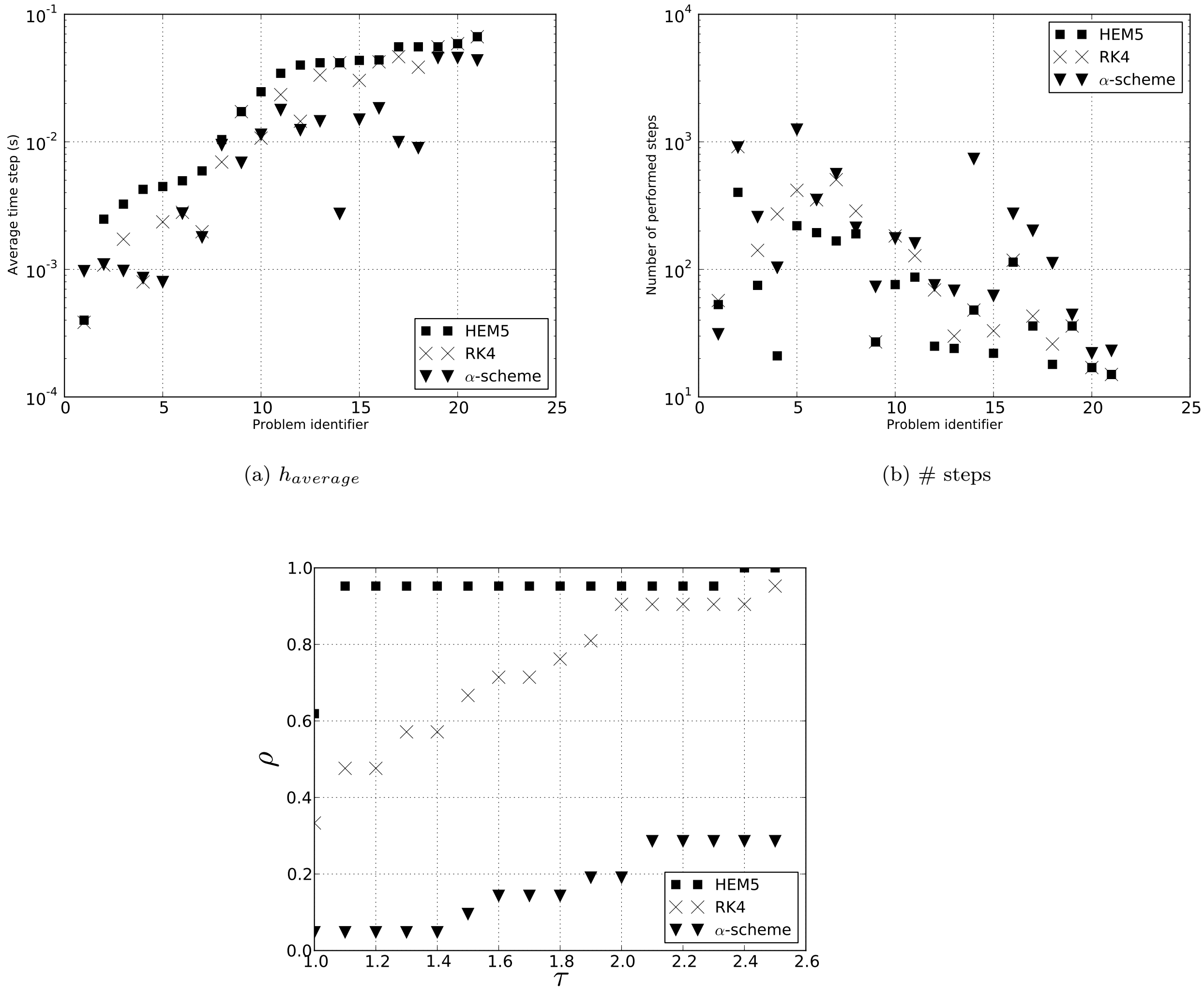


(a) $h_{average}$

(b) # steps

(c) performance profile

Figure 3.11: Average time step, number of iterations and performance profile of the second set

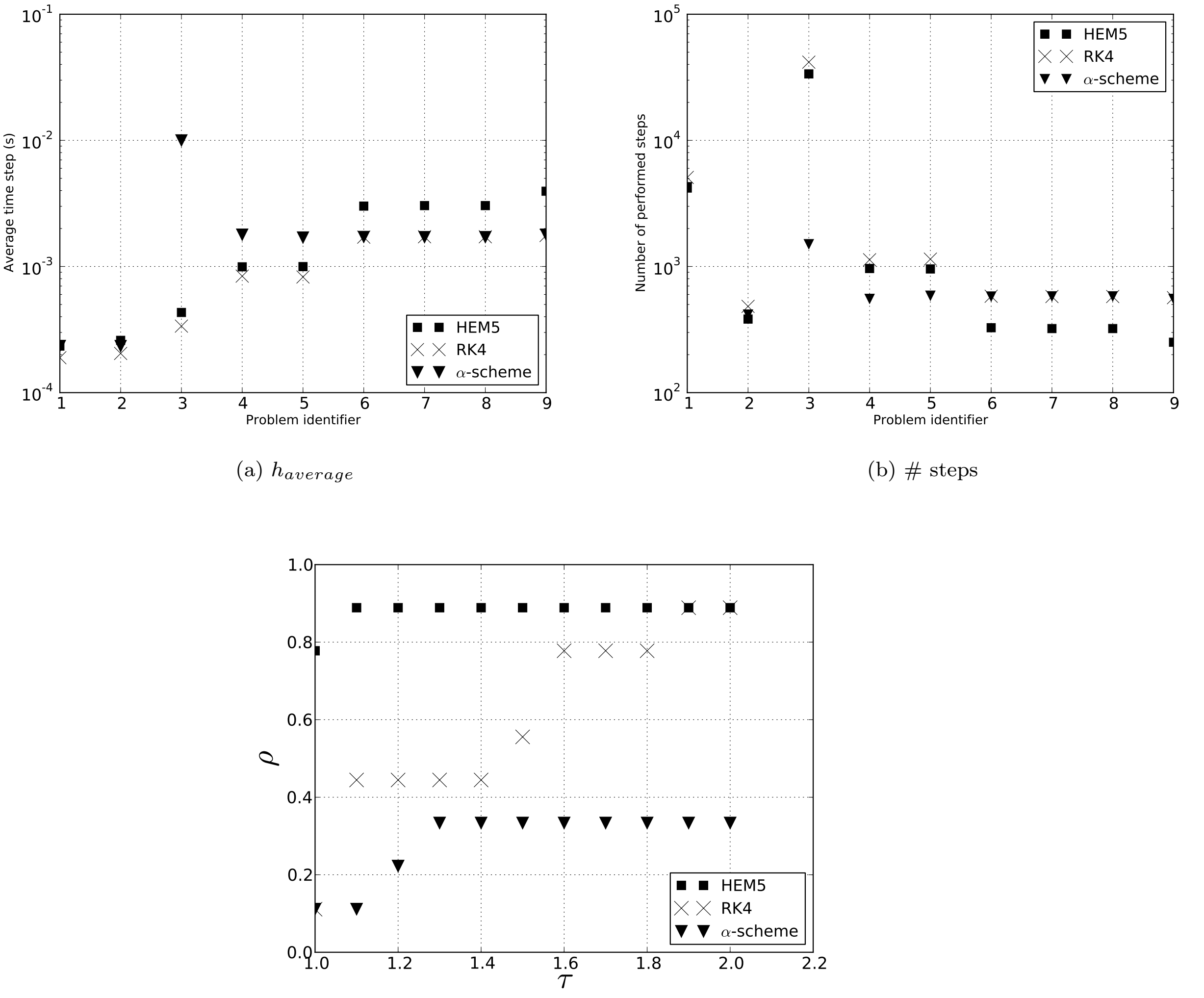


(a) $h_{average}$

(b) # steps

(c) performance profile

Figure 3.12: Average time step, number of iterations and performance profile of the third set



(a) $h_{average}$

(b) # steps

(c) performance profile

Figure 3.13: Average time step, number of iterations and performance profile of the fourth set

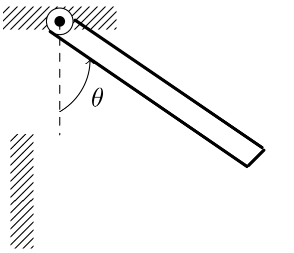


Figure 3.14: Flexible rotating beam

Table 3.8: Eccentrically suspended rotating beam: average time step size, simulation time, number of accepted steps and number of rejected steps for different tolerances.

method	tolerance	average h	t_s	accepted	rejected
HEM5	10^{-2}	$4.22\ 10^{-4}$	548.98	34123	1381
	10^{-4}	$4.20\ 10^{-4}$	549.70	34364	1393
	10^{-6}	$2.40\ 10^{-4}$	985.79	62255	179
RK4	10^{-2}	$3.38\ 10^{-4}$	384.13	41428	2903
	10^{-4}	$3.34\ 10^{-4}$	376.89	41700	2825
	10^{-6}	$3.37\ 10^{-4}$	398.29	41562	2876
α-scheme, $\rho_\infty = 0.99$	10^{-2}	$9.61\ 10^{-2}$	10.55	156	0
	10^{-4}	$8.01\ 10^{-3}$	86.32	1869	3
	10^{-6}	$2.35\ 10^{-3}$	264.80	6371	3
α-scheme, $\rho_\infty = 0.8$	10^{-2}	$9.61\ 10^{-2}$	10.69	156	0
	10^{-4}	$3.16\ 10^{-2}$	24.42	474	0
	10^{-6}	$6.15\ 10^{-3}$	109.83	2437	0
α-scheme, $\rho_\infty = 0.5$	10^{-2}	$9.61\ 10^{-2}$	11.29	156	0
	10^{-4}	$2.74\ 10^{-2}$	28.72	538	0
	10^{-6}	$5.50\ 10^{-3}$	123.09	2714	0

time-consuming than HEM5 and up to 36 times less time-consuming than RK4. Introducing some numerical damping by decreasing the value of the parameter ρ_∞ in (2.54), enables the α-scheme to be more efficient by using larger time steps and reducing the computation time. As expected, fully implicit time stepping schemes are more relevant for stiff mechanical systems.

Let us consider another challenging mechanism consisting of a rotor mechanism that has been analyzed in [87], depicted in Fig. 3.15. The system is composed of a flexible shaft, a symmetric disk, and a bearing. The axis of the rotor is along the y axis and displacements are allowed in the y and z directions. The system exhibits frequencies that are around 10^5Hz. This makes it extremely hard for HEM5 and RK4 to compute the dynamics of the mechanism. Indeed, these two solvers fail to integrate the dynamics with adapted time steps, with a minimum value of 10^{-8}s. We computed the dynamics using the α-scheme using a constant time step of $4.\,10^{-4}$s. The amplitude of the displacement of the rotor with respect to the rotation velocity is shown in Fig.3.16.

Therefore, when the dynamics is stiff, either resulting from some geometric stiffening as in the case of the rotating bar, or when the dynamics involves high frequencies as in the case of the rotor mechanism, explicit and half-explicit schemes use very small time steps in order to stay inside their stability regions. However, the implicit schemes, for instance the generalized-α scheme either for index-3 or index-2 DAE, proves to be efficient. In addition, introducing some numerical damping

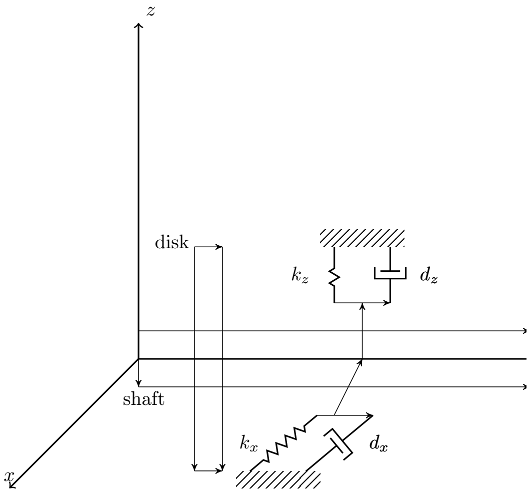


Figure 3.15: Rotor mechanism

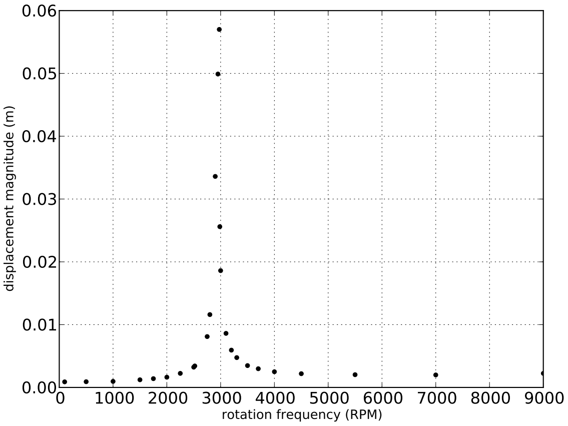


Figure 3.16: Amplitude of the displacement of the rotor

through the the spectral radius ρ_∞ enables to reduce the computation time, as shown in Table 3.8.

Overall, we came across the same conclusions that we drew for the academic examples:

- Explicit and half-explicit schemes outperform the implicit schemes on mechanisms involving unilateral and bilateral constraints, when the dynamics is not stiff.
- The discretization of the constraints at the velocity level proves once again to be a better option than considering position constraints or acceleration constraints. Indeed, it enables a better stabilization of the drift, and therefore saving computation time because it reduces drastically the need of performing calculations to correct the constraints.
- When the dynamics is highly stiff and non-linear, implicit schemes represent the best, if not the only option, to integrate the equations of motion.

3.3 Conclusion

This chapter aimed at performing a thorough comparison between index-3, index-2 and index-1 DAE solvers. To this purpose, we chose several methods: the generalized-α scheme with bilateral constraints discretized at the position level for the first category, the PHEM56, HEM5 and the generalized-α scheme with bilateral constraints discretized at the velocity level for the second one, and the Runge-Kutta-Fehlberg and RK4 methods for the third category. These schemes have been compared on academic and industrial benchmarks. The comparison covers three aspects: stabilization of the drift of the constraints, computational effort and handling the difficulties in case of a stiff dynamics. In order to achieve our objective, we performed simulations on several mechanisms with an adapted time step size to the user defined tolerance on the truncation error. After analyzing the time step sizes used by each solver, the computational cost, the maximum drift of the constraints and the behavior in case of stiff dynamics, the following salient conclusions may be drawn:

- *Computational effort.* Even though the PHEM56 and HEM5 solvers contain more stages (8 and 6 respectively) than the RKF45 and RK4 scheme (5 and 4 respectively), the numerical effort of the semi-explicit solvers is lower than that of the explicit ones when a strategy of time step control is used. Indeed, the order (5) of PHEM56 and HEM5 in addition to their characteristics of reducing the violation of the constraints, enable to use larger step sizes than those used for RKF45 and RK4, and then to reduce the computational effort. For non-stiff mechanisms, both explicit and half-explicit methods give better results than the generalized-α scheme that needs very small time step sizes mainly due to its lower order (2) and the drift at the velocity level when using it with the index-3 DAE formulation. We can also notice that using the α-scheme with velocity constraints enables to do simulations that are less time-consuming than in the case of position constraints.

- *Drift of the constraints.* PHEM56 and HEM5 enforce the constraints at the velocity level, this enables to reduce drastically the drift at the position level. This enables often to perform the integration without any procedure of projection of the constraints on the admissible manifold. This is not the case of the explicit scheme where a projection on the constraints is required in some cases. The generalized-α scheme used with the index-3 DAE formulation enforces the constraints at the position level but the drift at the velocity level may lead to numerically losing the contact depending on the tolerances that are chosen for the index set I_2 of closed contacts. Discretizing the dynamics with a formulation of the constraint at the velocity level seems to be a good compromise to stabilize the drift at acceptable tolerances.

- *Stiff dynamics.* When the mechanisms have a stiff and non-linear dynamics, explicit and half-explicit methods are not numerically efficient compared to the implicit α-scheme. We even saw that when high frequencies are involved, as in the case of the rotor mechanism, HEM5 and RK4 were unable to integrate the dynamics. In such cases, the generalized-α scheme proves its efficiency by handling this high frequency non-linear dynamics by the ability of introducing some numerical damping.

Half-explicit methods (HEM5 and PHEM56 for example) use linear systems (see (2.58) and (2.71) for example) involving jacobians that are evaluated at two different time instants. This represents additional computation time as well as some storage management in order to build and save the jacobian of the previous time. Moreover, when the constraints are redundant, a QR decomposition is performed on the jacobians. For systems such as (2.58) and (2.71), a QR decomposition has to be performed on each jacobian, which represents another numerical effort.

Chapter 4

A hybrid integration method

4.1 Motivation

In this section, we shall address several problems that are frequently encountered in the simulation of constrained multibody systems, and whose simulation requires attention. Namely, accumulation of impacts, accumulation of friction transitions, and C^1 and C^2 discontinuities in CAD geometries which lead to velocity and acceleration jumps. Let us be clear that handling such problems is difficult when using an event-driven strategy as it is the case in the Ansys RBD solver.

Our solution to these issues is a hybrid integration method, consisting of a mixed event-driven/time-stepping integration. The aim is to take advantage of the positive characteristics of each integration family. Indeed, event-driven schemes formulate the dynamics as DAEs over smooth periods, i.e. periods which are free of events. During these periods, root-finding algorithms are performed on the constraints functions to detect as accurately as possible the time of occurrence of an event, time where the event is handled.

Event-driven methods are proved to be efficient for a limited number of events. However, when the transitions or the events are frequent in a short time interval, then the event-detection methods become very time-consuming because of the need to handle every single event. In practice, it may happen that the normal post-impact velocity computed for one of the accumulated impacts does not satisfy the numerical thresholds defined for the index sets, therefore it is not taken into account after and the corresponding unilateral constraint is violated in the following steps, leading to wrong results. Heuristics may be used consisting for example in setting to zero the restitution coefficient for an impact when the velocity is small enough. However this requires introducing additional numerical thresholds that in practice prove to be problem-dependent. Another issue that occurs very often when handling events accumulation with event-driven schemes is the clustering of events. This consists in considering that the times of occurrence of two or more events are numerically close enough to assume that they happen simultaneously and therefore that they have to be solved at the same time in the same LCP (1.21). In the case of accumulation of "too many" events, it happens that some events are not selected in the set of the events to be clustered due to numerical thresholds that are not suitable for all the events. This leads sometimes to non-physical penetration because one or more contacts have not been solved, or to scenarios that are varying greatly when changing the time step.

In the frictional case, detection of sticking/sliding transitions and handling changes of sliding directions is time-consuming for event-driven methods, and very difficult in 3D. In addition, event-driven schemes are very sensitive to the tolerances used to define the index sets.

On the other hand, time-stepping schemes, which operate in the impulse-velocity domain, do not need as many index sets as for event-driven methods. Furthermore, they have proved their robustness for handling large number of events, as well as accumulation of impacts (known as Zeno phenomenon). However, these methods have low order (usually the local order is 1), contrary to event-driven schemes which allow for the use of high-order integration methods over the smooth periods. As discussed in section 2.1.2, several authors have tried to bring some improvements to the time-stepping schemes, either by using extrapolation techniques to enable a step size adjustment during impact-free periods as in [127], or by using high order DAE solvers during smooth periods as in [2]. The authors in [118, 117, 119] proposed to embed the classical Moreau-Jean time-stepping scheme in the discontinuous Galerkin methods in order to provide a high order approximation of the solution during the smooth periods. Another approach proposed in [35] consisted in the separation of the contribution of the nonsmooth variables and that of the smooth variables. This enables to integrate the smooth variables with any high order DAE solver, while the nonsmooth variables are evaluated with the classical Moreau-Jean scheme. This approach has been improved in [31] by taking into account both the position and the velocity constraints, following the GGL approach. For the evaluation of the smooth variables, the generalized-α method has been chosen to enjoy its stability characteristics when dealing with a stiff dynamics. Of course, all the aforementioned techniques do not allow for the increase of the global order of accuracy of the method.

In [43], a mixed even-driven/time-stepping integration method is proposed where classical DAEs solvers are used during smooth impact-free periods. A stop function detects roughly if one or more contacts are closed and enables to switch to the time-stepping scheme. This later is used to compute the dynamics as long as the contacts are closed (depending on the defined thresholds) in order to prevent multiple transitions from an integration scheme to the other and save the computation time used at the reinitialization. However, no precise information is given about the stop function or about the conditions upon which the switch from event-driven to time-stepping (or vice versa) is made. In our work, more accurate conditions are established upon which we switch from an integration scheme to the other. We also discuss the global order of the proposed method and numerical simulations are used to discuss the efficiency of the method.

4.2 The hybrid integration algorithm

4.2.1 Presentation of the algorithm

The proposed algorithm lays on the idea of using the event-driven method with an order p DAE solver until a criterion indicating a switch situation is satisfied. The integration is then resumed with a time-stepping scheme method. When this criterion is no longer valid, we switch back to the event-driven method. This mixed event-driven/time-stepping strategy is illustrated in Fig.4.1.

The switch criteria are discussed below:

- When an impact is detected, its time of occurrence $t_i^\star$ is saved and compared to that of the previous one $t_{i-1}^\star$. The size of the time interval $[t_{i-1}^\star, t_i^\star]$ defined by these two impacts is compared to a critical time step size $h^\star$. When $t_i^\star - t_{i-1}^\star \leq h^\star$, then we consider that there is an accumulation of impacts. In this case, we perform the computation using the Moreau-Jean time-stepping scheme over the time interval $[t_n, t_n + h^\star]$. The time-stepping scheme is then used until the contact is released or in a stable closed state, then we switch back to the event-driven method.

- The case of friction deserves more attention. The difficulty of dealing with friction comes from the difficulty to handle the different transitions: sliding to sticking and sticking to sliding. In the 2D case, the interpolation of $g_T(q(t))$ is doable since $g_T(q(t)) \in \mathbb{R}$ even if it brings an additional numerical effort. Due to the numerical effort related to the resolution of the sticking-sliding transition using the solution proposed by [62], we choose to handle this transition using the time-stepping scheme. Therefore, if the tangential velocity vanishes during a step $[t_n, t_{n+1}]$, then the dynamics during this step is re-computed with the Moreau-Jean time-stepping scheme. By means of interpolation of the normal contact force λ_N^α, we can detect if its sign has changed at some some $\tilde{t}^\star$. A change of the sign of λ_N^α means that the contact has been released (or has become open). If this happens, then we switch back to the event-driven scheme. In the 3D case, handling the transitions is time-consuming and the interpolation of the tangential velocity and

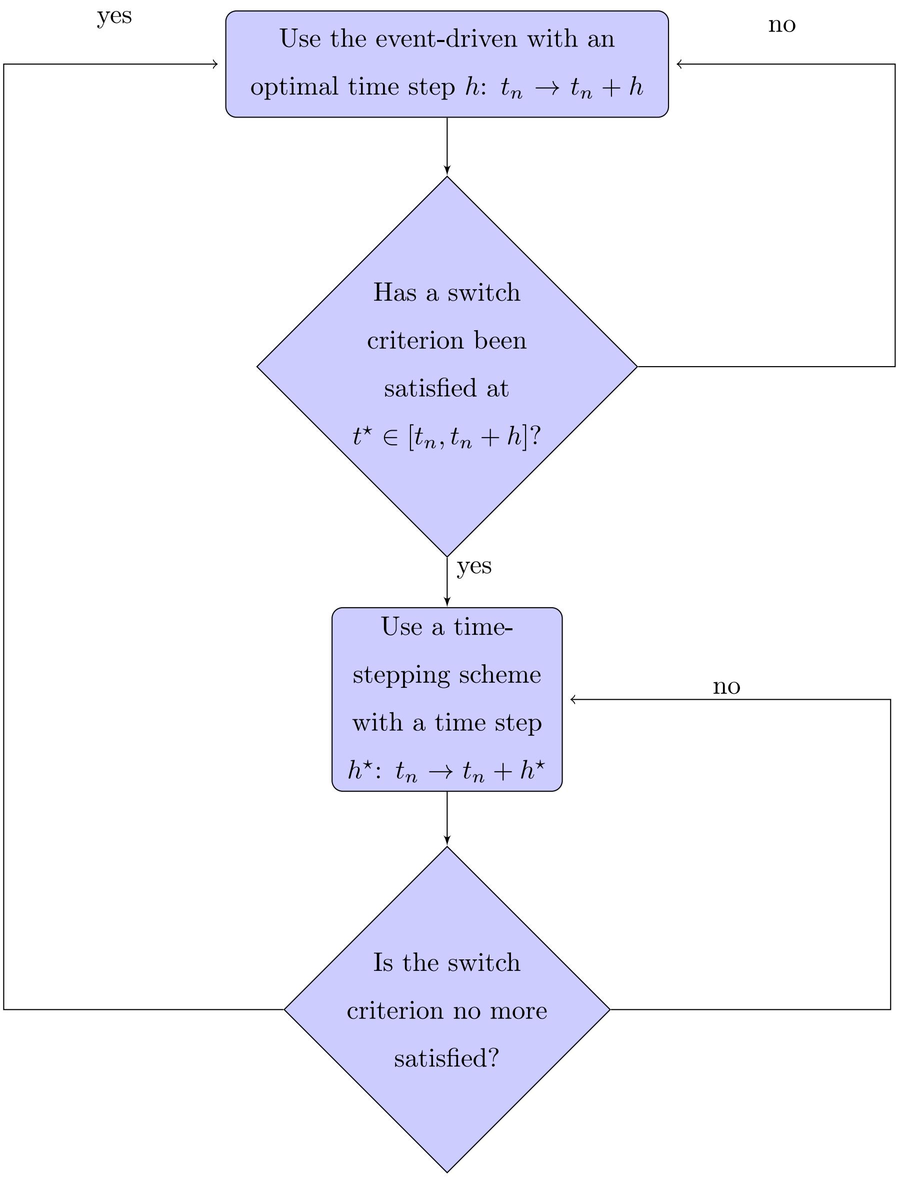

yes
Use the event-driven with an optimal time step h: $t_n \to t_n + h$
no
Has a switch criterion been satisfied at $t^\star \in [t_n, t_n + h]$?
yes
Use a time-stepping scheme with a time step $h^\star$: $t_n \to t_n + h^\star$
no
Is the switch criterion no more satisfied?

Figure 4.1: The mixed event-driven/time-stepping scheme

detecting the time it vanishes is a very hard task. In addition, detecting if the contact force is on the boundary of the friction cone, or inside the cone requires yet another threshold, also due to the numerous thresholds that are used. Therefore, in the 3D, all the steps during which a frictional contact is active are going to be handled using the time-stepping scheme.

- Finally, due to non-perfect geometries given in a CAD description, edges might be present that may render the resolution of the contact difficult. Indeed, in the neighborhood of an edge, the contact solver detects contact points on the left and on the right of the edge. When in addition the constraints are not sufficiently smooth in the neighborhood of the edge, then the systems that formulate the contact are ill-posed, as discussed in Section 4.3.3. Therefore, when the event of crossing such edges is detected during a step $[t_n, t_{n+1}]$, then the dynamics during this step is re-computed with the Moreau-Jean time-stepping scheme. The time-stepping scheme is used to compute the dynamics until we only detect contact pairs on the right or on the left of the transition. The detection of a transition is given by a function *OngoingTransition()* that uses the data from the geometry engine and returns a boolean indicating that there is an ongoing transition.

Let us denote ED the event-driven method and TS the time-stepping scheme. A sketch of the hybrid method is presented in Algorithm 1. Let us recall that I_{Sl} and I_St denote the index sets of sliding and sticking contacts, respectively, and are defined in Section 2.1.1.1.

4.2.2 Order of consistency of the mixed event-driven/time stepping strategy

In section 2.1.1.2, we discussed the consistency and the global order of an event-driven scheme when used with a DAE solver of order p. We consider that the conditions discussed in section 2.1.1.2 are satisfied so that the event-driven method is also of an accuracy order p. When we resume the integration with a time-stepping scheme with a time step size $h^\star$, this time step has to satisfy certain conditions in order not to reduce the order of the DAE method incorporated in the event-driven method. These conditions are addressed below.

Assumption 4.2.1 *The solution is assumed to have positions which are absolutely continuous and right velocities v^+ which are of Locally Bounded Variations, at each instant.*

Assumption 4.2.2 *We assume that:*

- *the inertia matrix $M(q)$ is definite positive and of class C^{p}.*

Algorithm 1 The hybrid integration method

```
Require: $q_n$, $v_n$, $t_n$
Ensure: $q_{n+1}$, $v_{n+1}$, $\lambda_{n+1}$
  $q_{n+1}$, $v_{n+1}$, $\lambda_{n+1}$ = $\mathsf{ED}(q_n, v_n, t_n)$
  Interpolate the position constraints $g(t)$
  Interpolate the tangential velocity constraints $\dot{g}_T(t)$
  Evaluate the index sets $I_2(t)$ and $ISt(t)$ (see section 2.1.1.1)
  case $\exists t_i^\star \mid g^\alpha(t_i^\star) = 0$, and $|t_i^\star - {t^\star}_{i-1}| \leq h^\star$ :
    repeat
      $q_{n+1}$, $v_{n+1}$, $\lambda_{n+1}$ = $\mathsf{TS}(q_n, v_n, t_n)$
    until $\alpha \in I_2$
  case $\Big(\exists \alpha \in I_{Sl}(t_n)\,\text{and}\,\exists t^\star \in [t_n, t_{n+1}] \mid \dot{g}_T^\alpha(t^\star) = 0\Big)$ or $\Big(I_{St}(t_n) \neq \emptyset\Big)$:
    repeat
      $q_{n+1}$, $v_{n+1}$, $\lambda_{n+1}$ = $\mathsf{TS}(q_n, v_n, t_n)$
    until $I_{St}(t_{n+1}) = \emptyset$
  case Edge transition at $t^\star$:
    repeat
      $q_{n+1}$, $v_{n+1}$, $\lambda_{n+1}$ = $\mathsf{TS}(q_n, v_n, t_n)$
    until *OngoingTransition()* returns false
```

- *the force vector $F(q,v,t)$ is of class C^{p}.*

- *the constraints vector $g(q)$ is of class C^{p+1}.*

- *$G(q)$ is of full row rank.*

C^{p} denotes the set of functions f whose derivatives $f^{(1)}, f^{(2)}, \dots f^{(p)}$ exist and are continuous.

Assumption 4.2.3 *The event-driven scheme is of order of consistency* $\boldsymbol{p}$*.*

Assumption 4.2.4 *The Moreau-Jean time-stepping scheme is used with a time step size $h^{\star}$ such that $h^{\star} = h^{p+1}$.*

Proposition 4.2.5 *Under assumptions discussed in section 2.1.1.2, 4.2.1, 4.2.2 4.2.3 and 4.2.4, the order of consistency of the proposed algorithms illustrated in Fig. 4.1 is* $\boldsymbol{p}$*.*

The proof of this proposition can be found in [2, 92].

4.3 Applications

In the following, several applications are presented where the dynamics is computed using the mixed event-driven/time-stepping scheme presented in Fig. 4.1.

4.3.1 Example with accumulation of impacts

The test consists of a classical example of a vertical chain of 3 beads, illustrated in Fig. 4.2. In Fig. 4.3, we show the time step sizes during the simulation time, as well as the evolution of the gap functions.

The integration is started with the event-driven scheme and the HEM5 scheme is used to integrate the dynamics and the step size is adjusted to meet a tolerance of 10^{-4} on the integration error. When the criterion indicating an accumulation of impacts is satisfied, which specifically means that the time interval between two successive events is smaller than a critical time step $h^{\star}$, then the integration is resumed with the classical Moreau-Jean scheme, with a time step $h^{\star}$. As discussed in Section 4.2.2, $h^{\star}$ should satisfy the condition $h^{\star} = h^{6}$, since HEM5 provides a solution of order $\mathsf{p} = 5$. This means that if the event-driven scheme uses a time step $h = 10^{-2}$,then $h^{\star} = 10^{-12}$, this time step is very small and will definitely prevent the simulation from advancing. In practice, we decided to set a minimum time step size $h_{min} = 5.10^{-7}$, if the computed $h^{\star}$ is smaller than h_{min}, then we set $h^{\star} = h_{min}$. This explains that all the steps computed with the Moreau-Jean scheme are run with the

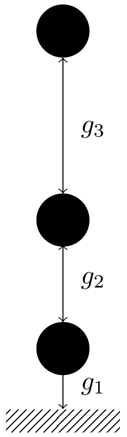


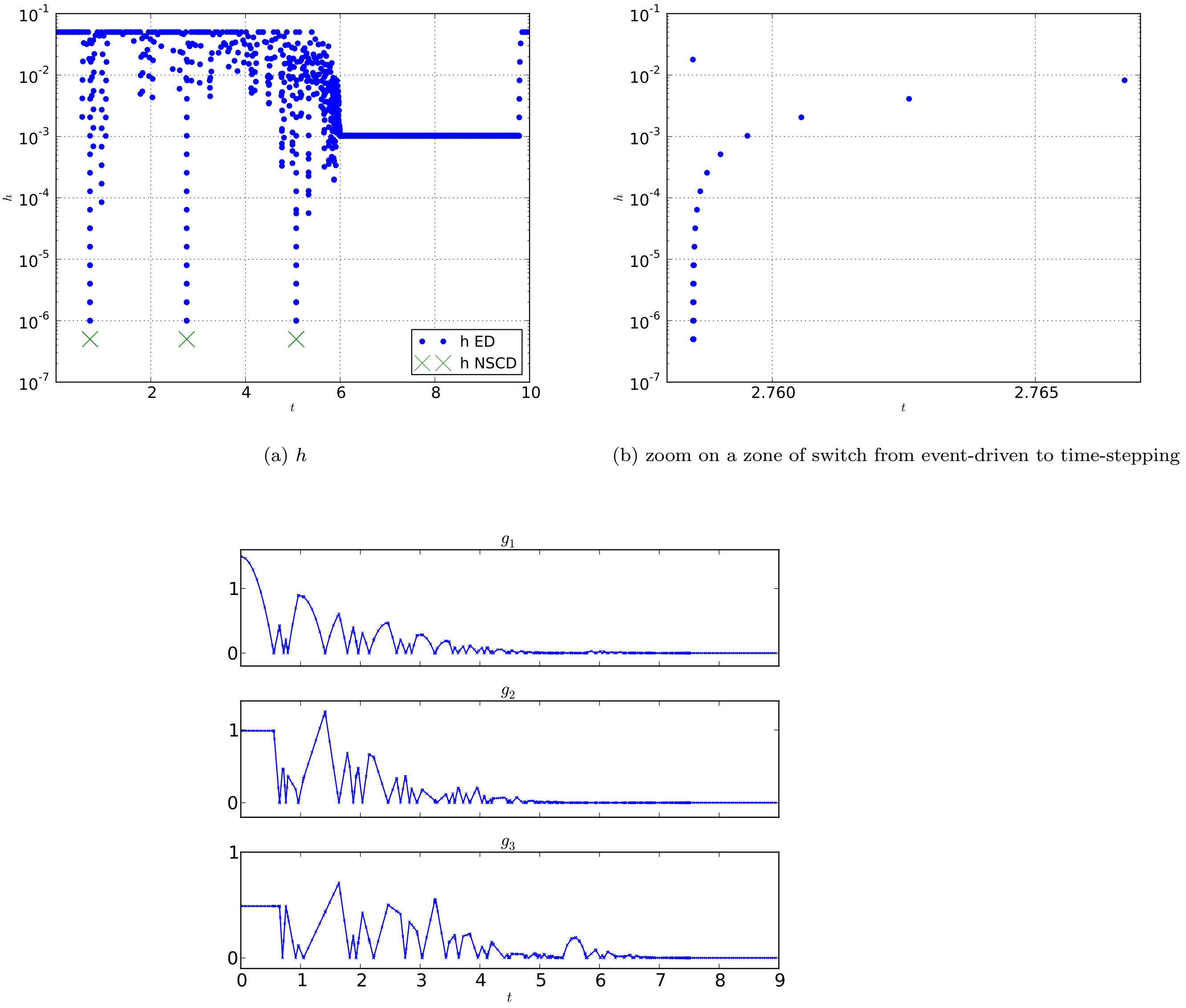
Figure 4.2: Chain of 3 balls

Table 4.1: Chain of balls: maximum violation

tol	maximum violation for g_1	maximum violation for g_2	maximum violation for g_3
10^{-2}	$3.54\,10^{-9}$	$5.26\,10^{-9}$	$4.64\,10^{-9}$
10^{-4}	$3.83\,10^{-9}$	$2.87\,10^{-9}$	$8.54\,10^{-9}$
10^{-6}	$2.44\,10^{-9}$	$2.78\,10^{-9}$	$2.49\,10^{-9}$
10^{-8}	$1.81\,10^{-9}$	$7.11\,10^{-9}$	$1.72\,10^{-9}$

minimum time step h_{min}. After a step size computed with the Moreau-Jean scheme, the integration is resumed with the event-driven strategy for which the time step size is adapted to meet the tolerance on the integration error. Fig.4.3b shows a zoom on the simulation result, namely the step size of an event-driven step, followed by a time-stepping step, and then the time step size is increased following the control strategy defined in (2.120). Finally, the time history of the gap functions of the problem are depicted in Fig.4.3c. Events are accurately detected in the event-driven strategy, the time step used for the time stepping scheme is small enough to prevent any non-physical penetration. When contacts are closed, the dynamics is computed with the event-driven scheme and the HEM5 scheme enforces very well the position constraints.

The maximum violation of the constraints with the hybrid scheme are reported in Table 4.1. We observe that for all the chosen precisions, the violations of the constraints is low and at the same order of magnitude (10^{-9}). This may be explained by the fact that for all the tolerances, the precision on the detection of the events is the same. Furthermore, since no correction of the constraints is performed, the HEM5 scheme in the event-driven method enforces the position constraints but not to a better level than the violation corresponding to the accuracy of the detection of the impacts.



(a) h

(b) zoom on a zone of switch from event-driven to time-stepping

(c) g

Figure 4.3: Chain of 3 balls, time step sizes and constraints with respect to time

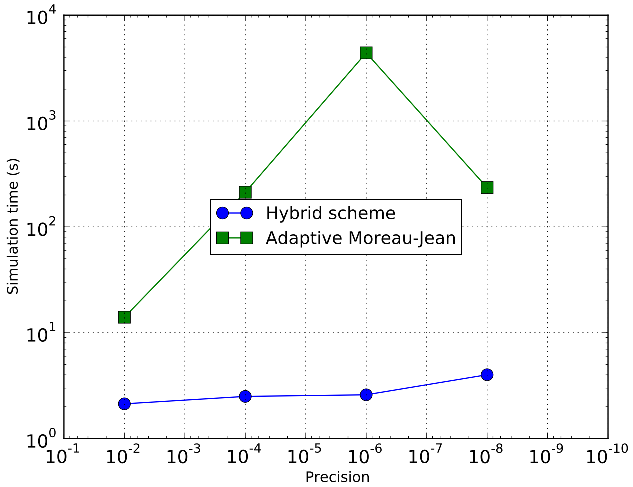


Figure 4.4: Comparison of the hybrid method with the Moreau-Jean scheme on a chain of three balls

We compared the proposed mixed event-driven/time-stepping scheme with the classical Moreau-Jean scheme with an adapted time step strategy. We chose the halved-steps method, discussed in Section 2.2.7 to control the time step size of the time-stepping scheme. The Moreau-Jean scheme is of local order 1 w.r.t.positions and at least of local order 0 w.r.t velocities (since they can undergo jumps during the simulation), we chose to use the positions to approximate the error. If q_2 is the position calculated with the Moreau-Jean scheme with a time step of length h, and $q_{\frac{1}{2}}$ is the approximation calculated after two steps of length $\frac{h}{2}$ each, then the error is $l_n = \| q_{\frac{1}{2}} - q_2 \|$. The optimal time step is computed using (2.120) with $\mathtt{p} = 1$.

For several tolerances on the integration error, we compared the simulation time for both schemes. Results are presented in the work-precision diagram of Fig.4.4.

The Moreau-Jean scheme being of a very low local order, needs to use much smaller time steps than the hybrid scheme where most steps are performed with the event-driven scheme with large time steps. Therefore, the simulation times with the Moreau-Jean scheme are much longer than with the hybrid scheme. It is worth noting that since we set a minimum value for the time step size when using the hybrid scheme, we cannot be sure that the required accuracy is satisfied. In general, it is difficult to evaluate the order of accuracy of the hybrid scheme, unless we have a reference (or analytical) solution that enables us to evaluate the error.

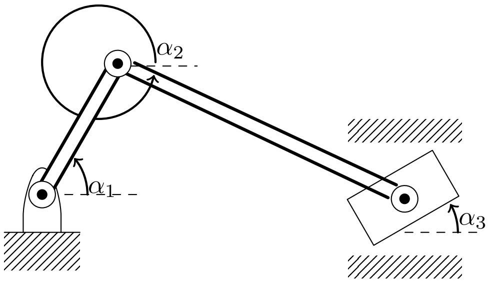


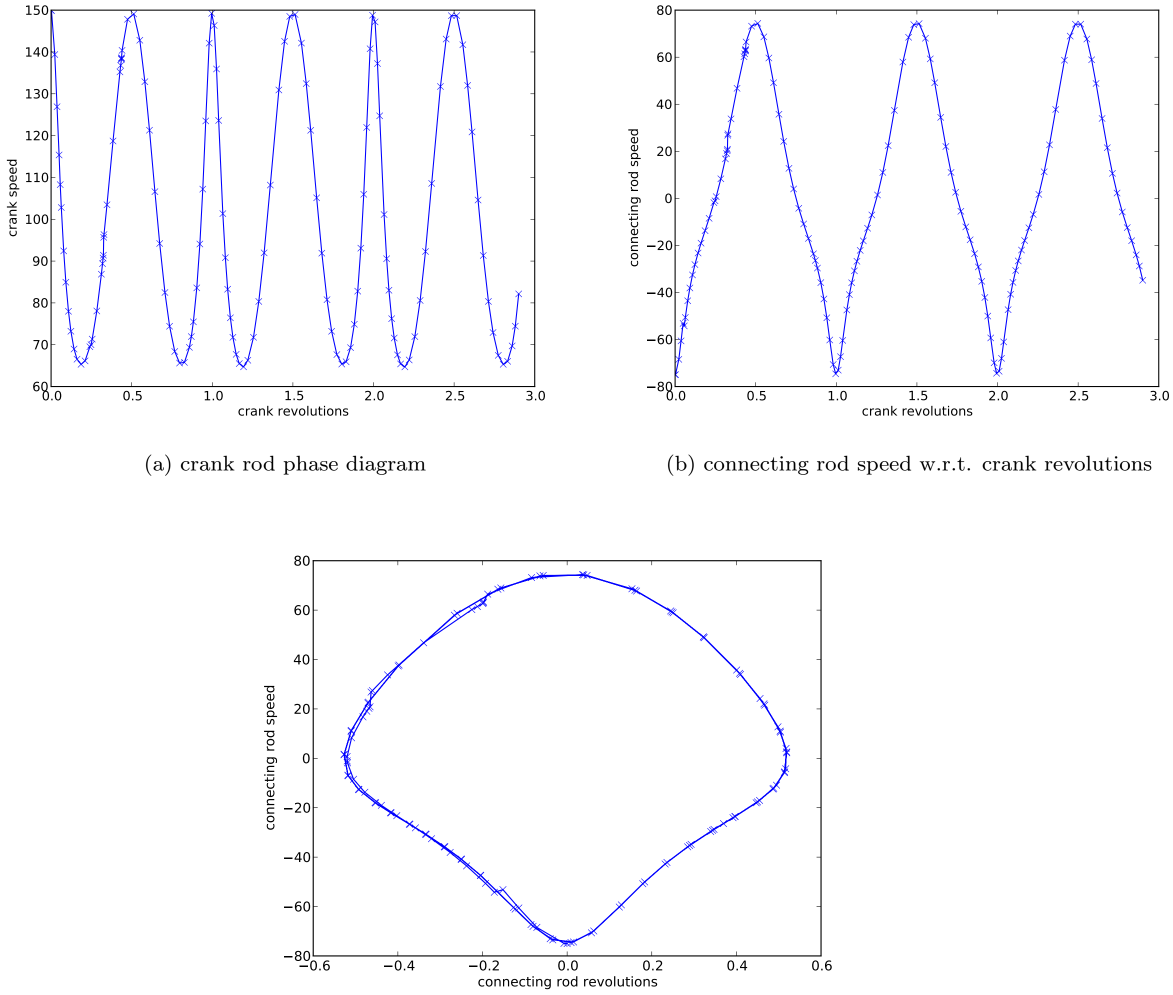
Figure 4.5: A slider-crank mechanism with clearance

4.3.2 Example with friction

As an example to illustrate our method in case of friction, we run a simulation on a rigid slider-crank mechanism with clearance, illustrated in Fig. 4.5. This system has been studied for example in [48, 3]. We used the strategy described in Fig.4.1, the HEM5 method is chosen to be the integration method of the event-driven scheme, the time step size is adjusted to meet a tolerance of 10^{-4} on the integration error. The mechanism is subjected to 4 unilateral constraints with friction.

Figures 4.6a, 4.6b and 4.6c depict the phase diagrams of the crank rod and the connecting rod. These results corroborate those given in [48, 3].

We compared our algorithm with the Moreau-Jean method where the time step has been controlled using the halved steps method discussed in sections 2.2.7 and 4.3.1. For different values of the tolerance on the integration error, we run simulations on the slider-crank mechanism with both methods, and we compared the performances of both algorithms. Figure 4.7 presents the work-precision diagram for both strategies, and Table 4.2 presents the average time step size for each method.



(a) crank rod phase diagram

(b) connecting rod speed w.r.t. crank revolutions

(c) connecting rod phase diagram

Figure 4.6: Slider-crank with clearance and friction, speeds of the crank and connecting rods

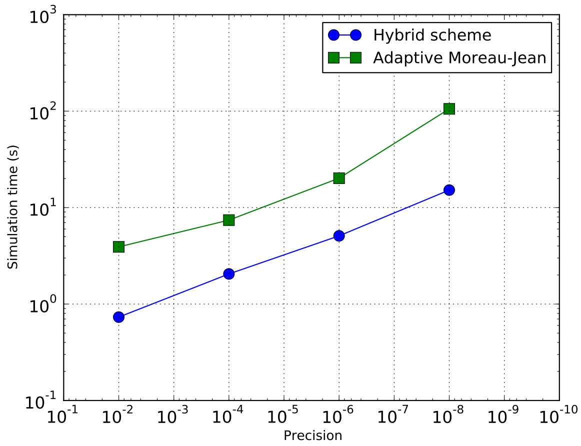


Figure 4.7: Work-precision diagram for the mixed event-driven/time-stepping scheme and the adaptive Moreau-Jean scheme for the Slider-Crank mechanism with clearance

tol	$h_{average}$ with the Hybrid algorithm	$h_{average}$ with the adaptive Moreau-Jean scheme
10^{-2}	1.75 10^{-2}	3.99 10^{-4}
10^{-4}	7.15 10^{-3}	2.66 10^{-3}
10^{-6}	2.97 10^{-3}	4.20 10^{-4}
10^{-8}	1.11 10^{-3}	8.62 10^{-5}

Table 4.2: Comparison of the hybrid algorithm and the adaptive Moreau-Jean scheme on the Slider-crank mechanism with clearance

For all the selected values of the tolerance on the local error, the hybrid integration scheme needs less simulation time than the adaptive Moreau-Jean scheme because (see Fig. 4.7) it takes advantage from the high order DAE solver (HEM5) incorporated in the event-driven method. The local order of the Moreau-Jean scheme being lower, it needs smaller time steps to meet the tolerances than those used for the hybrid method. Once again, we recall that the defined accuracies are imposed on the integration error for the HEM5 scheme incorporated in the event-driven method. For this accuracy to be satisfied for the hybrid scheme, the detection for the events must be "very" accurate for the event-driven method and the time-step size used for the time stepping scheme must satisfy the Assumption 4.2.4. Since we define a minimum value for the time step size ($h_{min} = 10^{-7}$s), this assumption is not satisfied and we cannot be sure that the hybrid scheme accuracy meets the defined tolerance. The accuracy order of the hybrid scheme is not easy to evaluate, unless we know an analytical solution for the problem, which enables us to evaluate the error made.

4.3.3 Bodies with C^0 and C^1 constraints

In this section, we discuss the changes that occur when there are some edges or other discontinuities in a given geometry. These discontinuities make the integration of the equations of motion difficult, and could even lead to incoherent results if they are not correctly treated. In practice, the geometry engine (of Ansys for example) provides pairs of contact points that are within a "touching" tolerance. When the theoretical points are close to the transition between two surfaces separated by an edge, pairs of contact points on both sides meet the contact touching tolerance, also meaning that we have active constraints on both sides of the edge. The data (normal velocities for instance) of these contact points are then used to formulate the contact problem, leading therefore to ill-posed problems as discussed in the sequel.

4.3.3.1 C^0 constraints

Glocker has addressed the problem of jumps in the state at the transitions in [59] for C^0-constraints, starting from the measure equality of Moreau:

$$M(q)dv = Fdt + dR. \tag{4.1}$$

where F collects the external and the Coriolis forces, and dR denotes the impulse of the impact. The post-impact velocity at a given time $t^\star$ is computed by integration Eq.(4.1), which gives

$$M(q^\star)dv = dR = G(q^\star)d\lambda. \tag{4.2}$$

Glocker proposes two solutions to compute the post-impact velocity as well as the impulse. Both methods preserve the kinetic energy of the system and are briefly exposed below.

The time scaling method An artificial time $\tau \in [0,1]$ is introduced such that $dv = vd\tau$ and $dR = Rd\tau$, with $d\tau$ being a Lebesgue measure on $[0,1]$. Therefore, the discontinuous gradient at the connection is transformed into a continuous gradient defined as

$$G(\tau) = (1-\tau)G_1 + \tau G_2,\ \tau \in [0,1]. \tag{4.3}$$

A smooth relative velocity is also defined with

$$\dot{g} = G^T v = 0. \tag{4.4}$$

After differentiating twice the relative velocity, we consider the index-1 problem

$$\begin{cases} Mv = G\lambda \\ \ddot{g} = G^T v + \dot{G}^T v = 0 \end{cases} \tag{4.5}$$

Finally, integrating this equation over $\{t^\star\}$ gives the post impact velocity and the impulse

$$\int_{\{t^\star\}} M dv = \int_0^1 M v(\tau) d\tau, \tag{4.6}$$

$$\int_{\{t^\star\}} dR = \int_0^1 G(\tau)\lambda(\tau) d\tau, \tag{4.7}$$

with $v(\tau = 0) = v^-$ and $v(\tau = 1) = v^+$.

The reflection law Glocker states in [59, Proposition 3] that *"every velocity jump $v^+ \neq v^-$ at which kinetic energy is preserved may be interpreted as a non-dissipative collision, i.e. as a reflexion with v^+ being the mirror image of v^- with respect to a hyperplane with normal in the direction of $M^{-1}R$"*. The starting point of this method is to transform the initial non-convex problem into a convex one by restricting the impact impulse R to lie in the convex cone $\{G_1, G_2\}$. Two variables u and $u^\perp$ are defined as

$$u = v^- + v^+ \tag{4.8}$$

and

$$u^\perp = v^- - v^+. \tag{4.9}$$

To construct the reflexion, the angle bisector of G_1 and G_2 has to be computed. To this aim, two normal vectors n_1 and n_2 are defined as

$$n_1 = \frac{G_1}{\parallel G_1 \parallel} \tag{4.10}$$

and

$$n_2 = \frac{G_2}{\parallel G_2 \parallel}. \tag{4.11}$$

Therefore, one possibility for the bisector is

$$n_P = \frac{n_1 + n_2}{\parallel n_1 + n_2 \parallel}. \tag{4.12}$$

Once the direction n_P of R has been determined, $u^\perp$ is determined using the relation

$$u^\perp = 2M^{-1} n_P n_P^T v^-. \tag{4.13}$$

Finally, the post impact velocity is computed with

$$v^+ = v^- - u^\perp, \tag{4.14}$$

and the impulse is computed with

$$R = M(v^+ - v^-). \tag{4.15}$$

The calculations have not been presented in detail in this section but can be found in [59].

In this section, we analyze the problem starting from the formulation (1.22), Filippov theory is then used to better understand the issue. Next, examples are studied that enable to illustrate the difficulty of numerically computing the dynamics when crossing an edge. Finally, solutions are proposed to solve the issue and applied to the examples.

4.3.3.2 C^1 constraints

In this section, we consider multibody systems with C^1 constraints, meaning that the jacobians are continuous and that there is a jump in the Hessians matrices.

Let us consider two rigid bodies B_A and B_B which can come in contact, as illustrated in Fig.1.6 where C_A and C_B denote the potential contacting points on B_A and B_B respectively, and n^α the normal vector to the contact tangent plane, oriented from B_1 to B_2 and g^α denotes the signed distance between the two bodies. The kinematic equations can be written

$$\begin{cases} g^\alpha(q) \geq 0 \\ \dot{g}^\alpha(q) = \nabla g^\alpha(q)^T \dot{q} \\ \ddot{g}^\alpha(q) = \nabla g^\alpha(q)^T \ddot{q} + \big(\dfrac{d\nabla g^\alpha(q)}{dt}\big)^T \dot{q}. \end{cases} \tag{4.16}$$

In the case of problems with a change in the curvature as depicted in Figure 4.8, and when the contact is closed ($g^\alpha(q) = 0$ and $\dot{g}^\alpha(q) = 0$), the term $\big(\frac{d\nabla g(q)}{dt}\big)^T \dot{q}$ in (4.16) is discontinuous. For the problem illustrated in Fig. 4.8, the "domains" C_1, C_2 and the transition S_d may be defined using some geometrical function $h(q)$ as:

$$\begin{cases} C_1 = \{q \,|\, g_1(q) = 0,\, h(q) > 0\} \\ C_2 = \{q \,|\, g_2(q) = 0,\, h(q) < 0\} \\ S_d = \{q \,|\, g_1(q) = g_2(q) = h(q) = 0\}. \end{cases} \tag{4.17}$$

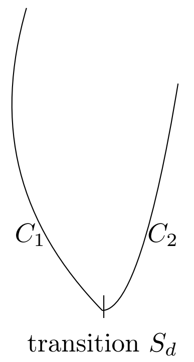


Figure 4.8: Problem with a change in the curvature

When crossing a transition, the solution may have jumps in the velocities and in the accelerations. Indeed, when formulating the dynamics of the constrained system as an index-1 DAE or as an ODE, to compute the accelerations and the contact forces, we have to solve system (1.25) or (1.26). Both systems involve the second derivative of the constraints. However, these systems are not solvable in the case of a system containing a change in the curvature, also meaning that we are considering two bilateral constraints whose second derivatives are not equal at the connection. Let us denote t_d the time at which the integration process arrives at this singularity, meaning that $h(q(t_d)) = 0$. Let us assume that the motion goes from C_1 to C_2. Just before the singularity, at t_d^-, system (1.25) (for example) is written with respect to the first constraint:

$$\begin{pmatrix} M(q(t_d^-)) & -{G_1}^T(q(t_d^-)) \\ G_1(q(t_d^-)) & 0 \end{pmatrix} \begin{pmatrix} \dot{v}(t_d^-) \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} F(t_d^-, q(t_d^-), v(t_d^-)) \\ -\frac{dG_1(q)}{dt} v(t_d^-) \end{pmatrix}. \tag{4.18}$$

Just after the singularity, the system becomes

$$\begin{pmatrix} M(q(t_d^+)) & -{G_2}^T(q(t_d^+)) \\ G_2(q(t_d^+)) & 0 \end{pmatrix} \begin{pmatrix} \dot{v}(t_d^+) \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} F(t_d^+, q(t_d^+), v(t_d^+)) \\ -\frac{dG_2(q)}{dt} v(t_d^+) \end{pmatrix}. \tag{4.19}$$

Therefore we switch from a given DAE to another one. Just at the singularity, if both constraints are active and if we do not take any care regarding the discontinuity the system reads

$$\left(\begin{array}{c|c} \mathrm{M(q(}t_d) & \begin{array}{c} -G_1^T(q(t_d)) \\ -G_2^T(q(t_d)) \end{array} \\ \hline \begin{array}{c} G_1(q(t_d)) \\ G_2(q(t_d)) \end{array} & 0 \end{array}\right) \begin{pmatrix} \dot{v}(t_d) \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} F(t_d, q(t_d), v(t_d)) \\ -\frac{dG_1(q)}{dt} v(t_d) \\ -\frac{dG_2(q)}{dt} v(t_d) \end{pmatrix}.$$

However the right-hand-side may not be in the image of the left block matrix. Numerically, this singularity results in a difficulty when computing the estimations of the accelerations or Lagrange multipliers because the system (1.25) is ill-posed. In the case of implicit schemes, the discontinuity results in a difficulty when solving system (2.84) which becomes ill posed at the singularity.

There is another problem related to the transition of edges when using Runge-Kutta methods in event-driven schemes. Indeed, these methods are based on the estimations of the solution at different moments of the time interval. When a transition happens, then some of the estimations use the constraints related to contact pairs on the left of the edge while others use constraints related to contact pairs on the right of the edge. As a consequence, the resultant approximation at the end of the time step leads to a relatively big truncation error and therefore to using small time step sizes. When the jacobians are equal at the transitions, we hoped that we do not have any problem computing the dynamics when using index-2 DAE solvers since they use the constraints at the velocity level. However, it appeared these schemes are not efficient at the transitions since they use linear systems involving jacobians that are evaluated at different times. As a consequence, we may have systems with jacobians corresponding to constraints on the left and on the right of the transitions, which renders the systems ill-posed (see Section 4.3.3.4).

4.3.3.3 Formulating the problem with C^1 constraints as a Filippov's differential inclusion

The theory of Filippov discussed in this section will help us validate the numerical results obtained in the following sections.

Let us consider a system subject to a set of bilateral constraints of the form $g(q) = 0$. In this case, we define

$$g(q) = \begin{cases} g_1(q), \text{ if } q \in C_1 \\ g_2(q), \text{ if } q \in C_2, \end{cases} \tag{4.20}$$

keeping in mind that $g(q)$ is a C^1 function of q. We showed that when the Delassus operator is invertible, we can evaluate λ and $\ddot{q}$ as

$$\begin{cases} \lambda = -(GM^{-1}G^T)^{-1}(GM^{-1}F + \dfrac{dG}{dt}\dot{q}) \\ \ddot{q} = M^{-1}F + M^{-1}G^T\lambda. \end{cases} \tag{4.21}$$

Let us denote $\frac{dG}{dt}^{-} = \frac{dG_1}{dt}$ and $\frac{dG}{dt}^{+} = \frac{dG_2}{dt}$. In the problem defined in Fig. 4.8, because of the change in the curvature between curves C_1 and C_2, we have

$$M(q)\ddot{q}+F(q,\dot{q},t) = \begin{cases} -G^T(q)((G(q)M^{-1}(q)G^T(q))^{-1}(G(q)M^{-1}(q)F(q,v,t) + \frac{dG}{dt}^{-}\dot{q})) & \text{if } q \in C_1 \\ \\ -G^T(q)((G(q)M^{-1}(q)G^T(q))^{-1}(G(q)M^{-1}(q)F(q,v,t) + \frac{dG}{dt}^{+}\dot{q})) & \text{if } q \in C_2. \end{cases} \tag{4.22}$$

To simplify the expression in (4.22), let us denote $W(q)$ the Delassus operator $G(q)M^{-1}(q)G^T(q),\ i = 1,2$, we then have

$$M(q)\ddot{q} + F(q,\dot{q},t) = \begin{cases} -G^T(q)(W^{-1}(G(q)M^{-1}(q)F(q,v,t) + \frac{dG}{dt}^{-}\dot{q})) & \text{if } q \in C_1 \\ \\ -G^T(q)(W^{-1}(G(q)M^{-1}(q)F(q,v,t) + \frac{dG}{dt}^{+}\dot{q})) & \text{if } q \in C_2. \end{cases} \tag{4.23}$$

Let us consider the case when the dynamics is formulated as an ODE, as in equation (1.26). The presence of a singularity leads to

$$\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{cases} \begin{pmatrix} v \\ M^{-1}(q)F(q,v,t) - M^{-1}(q)G^T(q)W^{-1}\left(G(q)M^{-1}(q)F(q,v,t) + \frac{dG}{dt}^{-}\dot{q}\right) \end{pmatrix} & \text{if } q \in C_1 \\ \\ \begin{pmatrix} v \\ M^{-1}(q)F(q,v,t) - M^{-1}(q)G^T(q)W^{-1}\left(G(q)M^{-1}(q)F(q,v,t) + \frac{dG}{dt}^{+}\dot{q}\right) \end{pmatrix} & \text{if } q \in C_2. \end{cases} \tag{4.24}$$

Since g is C^1 then λ computed with (4.21) may be discontinuous but remains bounded. If in addition the initial conditions (q_0, v_0) satisfy the constraints: $g(q_0) = 0$ and $G(q_0)v_0 = 0$, then, the dynamics evaluated with (4.24) ensures that the submanifold $\{(q,v)\,|\,g(q) = 0,\ G(q)v = 0\}$ is invariant.

Let us denote $X = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$ and $f_i(X) = u(q,\dot{q},t) + w_i(q,v,t)$ where

$$\begin{cases} u(q,v,t) = M^{-1}(q)F(q,v,t) - M^{-1}(q)G^T(q)W^{-1}\left(G(q)M^{-1}(q)F(q,v,t)\right) \\ w_i(q,v,t) = -M^{-1}(q)G^T(q)W^{-1}\left(\dfrac{dG_i}{dt}\dot{q}\right),\ i = 1,2. \end{cases} \tag{4.25}$$

Then the problem can be formulated as

$$\dot{X} = \begin{cases} f_1(X), & \text{if } q \in C_1\,, \\ f_2(X), & \text{if } q \in C_2\,, \end{cases} \qquad X(0) = X_0 \in \mathbb{R}^n. \tag{4.26}$$

This problem is a **piecewise smooth** system, which can be analyzed using the theory of Filippov [46]. Let us consider system (4.26). The state space $\mathbb{R}^n$ is split into the subspaces C_1 and C_2 by the surface S_d defining the zone of the singularity, characterized by a function $s : \mathbb{R}^{2n} \longrightarrow \mathbb{R}$, such that $S_d = \{X \in \mathbb{R}^{2n}\,|\,s(X) = 0\}$. When $s \in \mathsf{C}^{\mathsf{k}}$ (C^{k} is the set of functions with k derivatives), we can define the unit normal vector to the tangent plane to S_d at X as

$$\tilde{n}(X) = \frac{\nabla(s(X))}{\parallel \nabla(s(X)) \parallel}. \tag{4.27}$$

Since $\frac{dG}{dt}^{+}\dot{q} \neq \frac{dG}{dt}^{-}\dot{q}$, the dynamics is not defined at S_d where both constraints are active. A common way to overcome this problem is to extend the discontinuous system into a **Filippov's differential inclusion**, also known as **Filippov convex method** [46], defined as

$$\dot{X} \in \begin{cases} f_1(X), & \text{if } q \in C_1\,, \\ f_2(X), & \text{if } q \in C_2\,, \\ f_H(X), & \text{if } q \in S_d\,, \end{cases} \qquad X(0) = X_0 \in \mathbb{R}^n, \tag{4.28}$$

where H is the closed convex hull defined as $H = \{f_H \mid f_H(X) = (1-\alpha)f_1(X) + \alpha f_2(X)\,, X \in S_d,\ \alpha \in [0\,,\,1]\}$. In the case of a constrained multibody system, this convex hull is given with

$$H = \{f_H \mid f_H(X) = \ (1-\alpha)w_1(X) + \alpha w_2(X) + u(X)\,, X \in S_d,\ \alpha \in [0\,,\,1]\} \tag{4.29}$$

where the functions u, w_1 and w_2 are defined in (4.25). Let $f_{1p} = \tilde{n}^T f_1$ and $f_{2p} = \tilde{n}^T f_2$ be the projections of f_1 and f_2 onto the normal to S_d.

Remark: Filippov's convexification method totally disregards what may happen on the discontinuity surface. It focuses on the right- and left-limits of the vector fields and defines a sliding motion in case the discontinuity surface is attractive.

Different modes can be defined [46, 37] depending on the sign of the product $f_{1p}^T f_{2p}$.

- If $f_{1p}^T f_{2p} > 0$ on S_d, then trajectories leave the surface S_d. We have a **transversal mode**. In this case, a solution in the sense of Carathéodory can be defined [46, 30].

- If $f_{1p}^T f_{2p} < 0$ on S_d, then:

  1. if $f_{1p} > 0$ and $f_{2p} < 0$, we have an **attractive sliding mode** on S_d. Solutions are approaching S_d from both sides as time increases and Filippov provides an extension of the solution on S_d as

  $$f_e(X) = (1-\alpha(X))f_1(X) + \alpha(X)f_2(X), \tag{4.30}$$

  where $\alpha(X) = \frac{f_{1p}}{f_{1p} - f_{2p}}$.

  2. if $f_{1p} < 0$ and $f_{2p} > 0$, we have a **repulsive sliding mode**. The solution can leave the surface S_d at any time, or stay on it.

Fig. 4.11 illustrates the transversal and sliding modes.

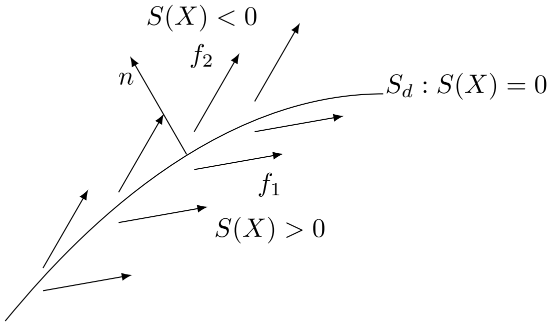


Figure 4.9: Transversal mode

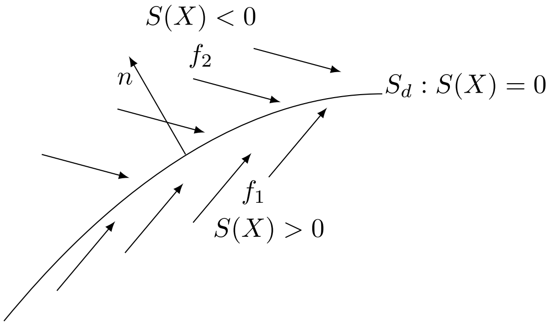


Figure 4.10: Sliding mode

Figure 4.11: Transversal and Sliding modes

The analysis developed in this section leads us to believe that the simulation of the dynamics of systems with edges resulting from non-perfect CAD descriptions would be difficult only when using schemes dedicated to index-1 DAEs or ODEs because these methods solve the constraints on the acceleration level. When the acceleration constraints are discontinuous, these methods fail because the systems to solve to get the estimations are ill-posed. However, we also experienced problems when using solvers for index-2 DAEs, as discussed in the following.

4.3.3.4 Illustrative examples for C^1 constraints

We will now present some examples that illustrate the problems presented in the previous subsection. In each of the following examples, the mechanical system can be considered as being submitted to two different smooth constraints, and the switch between both constraints is made at the "geometrical discontinuity" where they are both active, or as one constraint that is not sufficiently differentiable at a singularity. We choose to consider the first option in this document. Through two simple examples, we try to reproduce the difficulties faced with the Ansys solver when contact pairs on the left and on the right of a transition are detected.

Example 1 Let us consider the case of a ball sliding on a portion of a parabola of equation $y = \frac{1}{2}x^2$ before sliding on the plane of equation $y = 0$, as depicted in Figure 4.12.

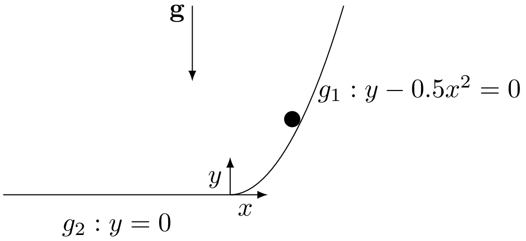


Figure 4.12: Ball sliding on a parabola and then on a plane

The ball is described by the generalized coordinates $[x,\ y]^T$ and is submitted to gravity. Its initial position and velocity are: $[2,\ 2]^T$ and $[-1,\ -2]^T$ respectively. Note that the initial conditions are consistent with the constraints, which are written as

$$\begin{cases} g_1(q) = y - 0.5x^2, \text{ if} x \geq 0 \\ g_2(q) = y, \text{ if} x \leq 0, \end{cases} \tag{4.31}$$

the Jacobians are given by

$$\begin{cases} G_1(q) = [-x\ \ 1], \text{ if} x \geq 0 \\ G_2(q) = [0\ \ 1], \text{ if} x \leq 0, \end{cases} \tag{4.32}$$

and the Hessian matrices are $\begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$ if $x \geq 0$ and $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ if $x \leq 0$. The function h defined in section 4.3.3.2 reads: $h(q) = x$.

The equations of motion can be written as

$$\begin{cases} m\ddot{x} = r_x \\ m\ddot{y} = -m\,\mathbf{g} + r_y, \end{cases} \tag{4.33}$$

where m denotes the mass of the ball, $\mathbf{g}$ denotes the gravity and $r{=}[r_x,\ r_y]{=}G^T(q)\lambda$ is the contact force. By substituting with the expressions of $G(q)$, $F(q, \dot{q}, t)$ and $M(q)$, one gets

$$\begin{cases} r_x = \dfrac{-m\,x}{1+x^2}(\mathbf{g} + \dot{x}^2), \text{ if} x \geq 0 \\ r_x = 0, \text{ otherwise} \end{cases} \tag{4.34}$$

$$\begin{cases} r_y = \dfrac{m}{1+x^2}(\mathbf{g} + \dot{x}^2), \text{ if} x \geq 0 \\ r_y = m\mathbf{g}, \text{ otherwise.} \end{cases} \tag{4.35}$$

At the junction point $(0, 0)$, the second derivatives of the constraints are not equal. Indeed, the Hessian matrices at this point are $\begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$ if $x \geq 0$ and $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ if $x \leq 0$, which means that the term $\frac{dG(q)}{dt}\dot{q}$ is discontinuous at the connection point.

In this case, the numerical solvers experience some issues during the resolution of the dynamics in the neighborhood of the singularity ($x = 0$), assuming that the singularity is detected with a tolerance of 10^{-8}. In the step that follows the detection of this event, the systems used for the computation of the estimations $\dot{V}_i$ and Λ_i become ill-posed. Using the HEM5 scheme in the event-driven strategy, the system becomes:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 6.65031\,10^{-5} & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{V}_1 \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} 0 \\ -9.81 \\ 44.1971 \\ -3.62238\,10^{-2} \end{pmatrix}.$$

We can mention that the Hessian matrices (computed at the plane and at the parabola) are not compatible in the zone of the edge. The 3^{rd} and 4^{th} rows of this matrix show the non consistency of the normals evaluated at Q_1 and Q_2 respectively.

Let us analyze this problem applying the theory presented in Section 4.3.3.3, augmented of the higher order analysis presented in [37]. Let us denote $X = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} = [x,\, y,\, \dot{x},\, \dot{y}]^T$. The equations of motion of this system can be formulated as

$$\dot{X} = \begin{cases} f_1 = \begin{pmatrix} v \\ f_{11}(X,t) \end{pmatrix} & \text{if } x > 0 \\ \\ f_2 = \begin{pmatrix} v \\ f_{22}(X,t) \end{pmatrix} & \text{if } x < 0. \end{cases} \tag{4.36}$$

where $f_{11}(X,t) = \begin{pmatrix} \frac{-x}{1+x^2}(\mathbf{g} + \dot{x}^2) \\ -\mathbf{g} + \frac{\mathbf{g}+\dot{x}^2}{1+x^2} \end{pmatrix}$ and $f_{22}(X,t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. The normal to the switching surface in the state space $\mathbb{R}^4$ at the singularity is defined with $\tilde{n} = [1, 0, 0, 0]^T$. We define the functions $f_{1p} = f_1^T \tilde{n} = \dot{x}$ and $f_{2p} = f_2^T \tilde{n} = \dot{x}$. These vector fields are illustrated in Fig. 4.15 in which the switching line is in red, they are based on a simulation using the HEM5 scheme. We have $f_{1p}|_{x=0}^T f_{2p}|_{x=0} = \dot{x}_s^2 \geq 0$, where $\dot{x}_s$ denotes the x component of the velocity of the ball at the singularity. Different cases must be taken into account.

- If $\dot{x}_s \neq 0$, then we have a *transversal intersection.*
  - If $\dot{x}_s < 0$, the trajectories leave the singularity at $x = 0$ to enter the curve C_2 defined by the constraint $g_2(q) = 0$.
  - If $\dot{x}_s > 0$, the trajectories leave the singularity at $x = 0$ to enter the curve C_1 defined by the constraint $g_1(q) = 0$.
- if $\dot{x}_s = 0$, then the problem is continuous.

Example 2 Let us consider now another example, which is quite close to the one presented in section 4.3.3.4.

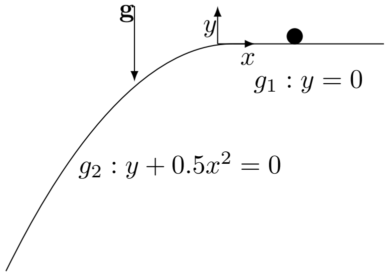


Figure 4.13: Ball sliding on a plane and then on a parabola

The ball is described by the general coordinates $q = [x,\ y]^T$ and is submitted to gravity. Its initial position and velocity are: $[2,\ 0]^T$ and $[-2,\ 0]^T$ respectively. The bilateral constraints read

$$\begin{cases} g_1(q) = y, \text{ if } x \geq 0 \\ g_2(q) = y + 0.5x^2, \text{ if } x \leq 0, \end{cases} \tag{4.37}$$

the Jacobians are given by

$$\begin{cases} G_1(q) = [x \ \ 1], \text{ if} x \geq 0 \\ G_2(q) = [0 \ \ 1], \text{ if} x \leq 0, \end{cases} \tag{4.38}$$

and the Hessian matrices are $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ if $x \geq 0$ and $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ if $x \leq 0$. By substituting with the expressions of $G(q)$, $F(q, \dot{q}, t)$ and $M(q)$, one gets the constraint force $r = G^T \lambda$ given by

$$\begin{cases} r = [0 \quad m\,\mathbf{g}]^T, \text{ if} x \geq 0 \\ r = [\dfrac{-m\,x}{1+x^2}(-\mathbf{g} + \dot{x}^2) \quad - \dfrac{m}{1+x^2}(-\mathbf{g} + \dot{x}^2)]^T, \text{ if} x \leq 0 \end{cases} \tag{4.39}$$

This problem can be formulated as a piecewise smooth system. By taking $X = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} = [x,\, y,\, \dot{x},\, \dot{y}]^T$, we have

$$\dot{X} = \begin{cases} f_1 = \begin{pmatrix} v \\ f_{11}(X,t) \end{pmatrix} & \text{if } x > 0 \\ \\ f_2 = \begin{pmatrix} v \\ f_{22}(X,t) \end{pmatrix} & \text{if } x < 0. \end{cases} \tag{4.40}$$

where $f_{11}(X,t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $f_{22}(X,t) = \begin{pmatrix} \frac{-x}{1+x^2}(-\mathbf{g} + \dot{x}^2) \\ -\mathbf{g} - \frac{-\mathbf{g}+\dot{x}^2}{1+x^2} \end{pmatrix}$. The normal to the switching surface in the state space $\mathbb{R}^4$ at the singularity is defined with $n = [1,0,0,0]^T$. The projections of the vector fields on the switching surface evaluated at the junction are $f_{1p}|_{x=0} = \dot{x}_s$ and $f_{2p}|_{x=0} = \dot{x}_s^2$, where $\dot{x}_s$ denotes the x component of the velocity of the ball at the singularity. Therefore, the analysis developed in Section 4.3.3.4 holds here.

The integration of the dynamics at the singularity fails because it requires to solve problems that are ill-posed. Here is an example of such a system when the HEM5 scheme is used

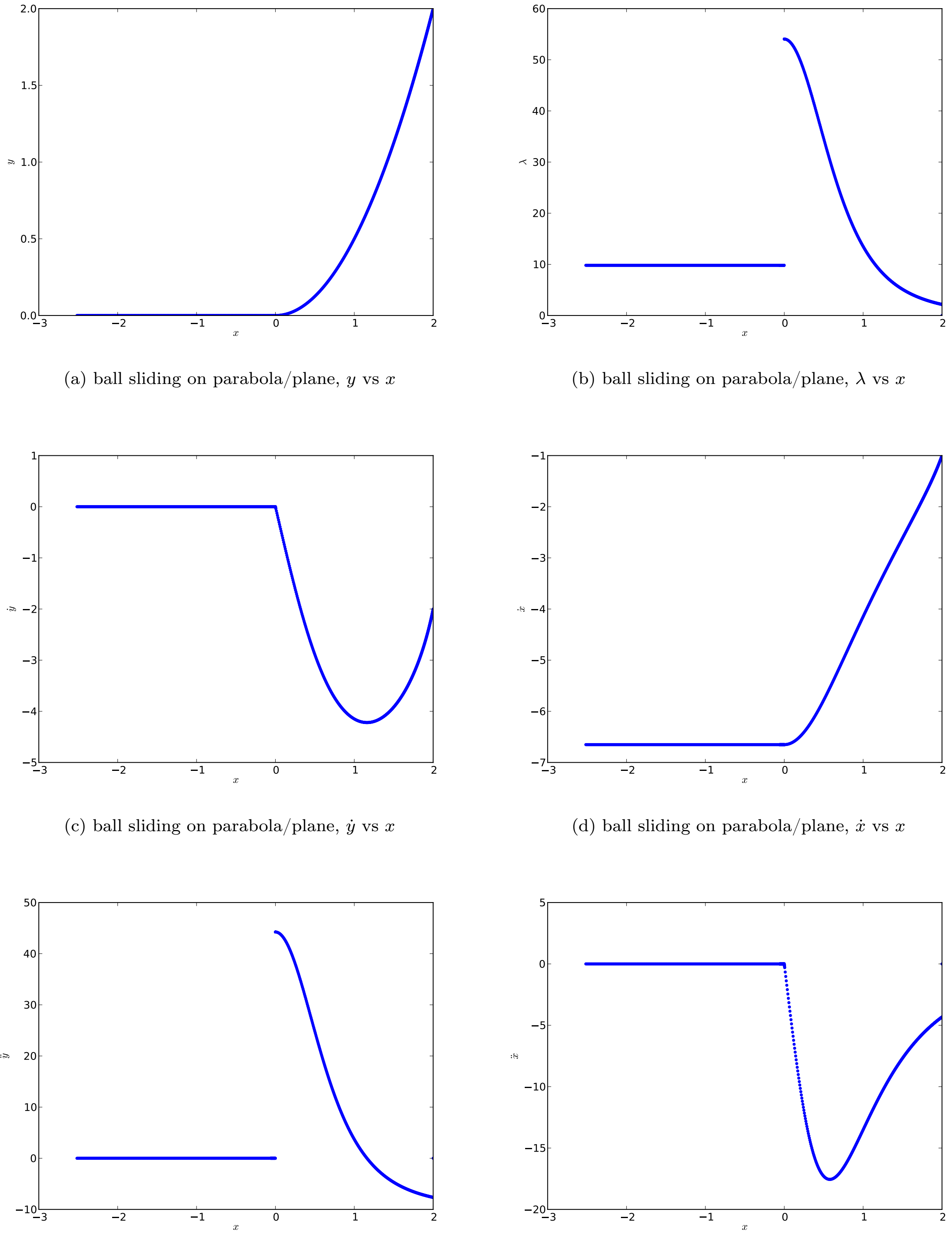
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ -1.99999\,10^{-5} & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{V}_1 \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} 0 \\ -9.81 \\ -3.99999 \\ 0 \end{pmatrix}.$$

The use of the least squares method to overcome this issue does not give satisfactory results. Indeed, with the HEM5 scheme, when it arrives to the edge with both constraints activated, the ball goes back in the direction to the plane instead of going on the parabola. This failure can be explained by the fact the least squares method is suitable for systems with a full rank matrix, which is not the case for example in (4.41).

4.3.3.5 Applying the solution to the issues

In this section, we present the evolution of the dynamics of each of the 2 examples, when applying the hybrid integration method to overcome the problems due to geometry.

Example 1 Figures 4.14a, 4.14b, 4.14c, 4.14d, 4.14e and 4.14f depict the evolution of some variables with respect to the x position. The constraints defined for this system have jacobians which are equal at the edge. Therefore, the velocities are continuous as illustrated in Figures 4.14c and 4.14d. However,



(a) ball sliding on parabola/plane, y vs x

(b) ball sliding on parabola/plane, λ vs x

(c) ball sliding on parabola/plane, $\dot{y}$ vs x

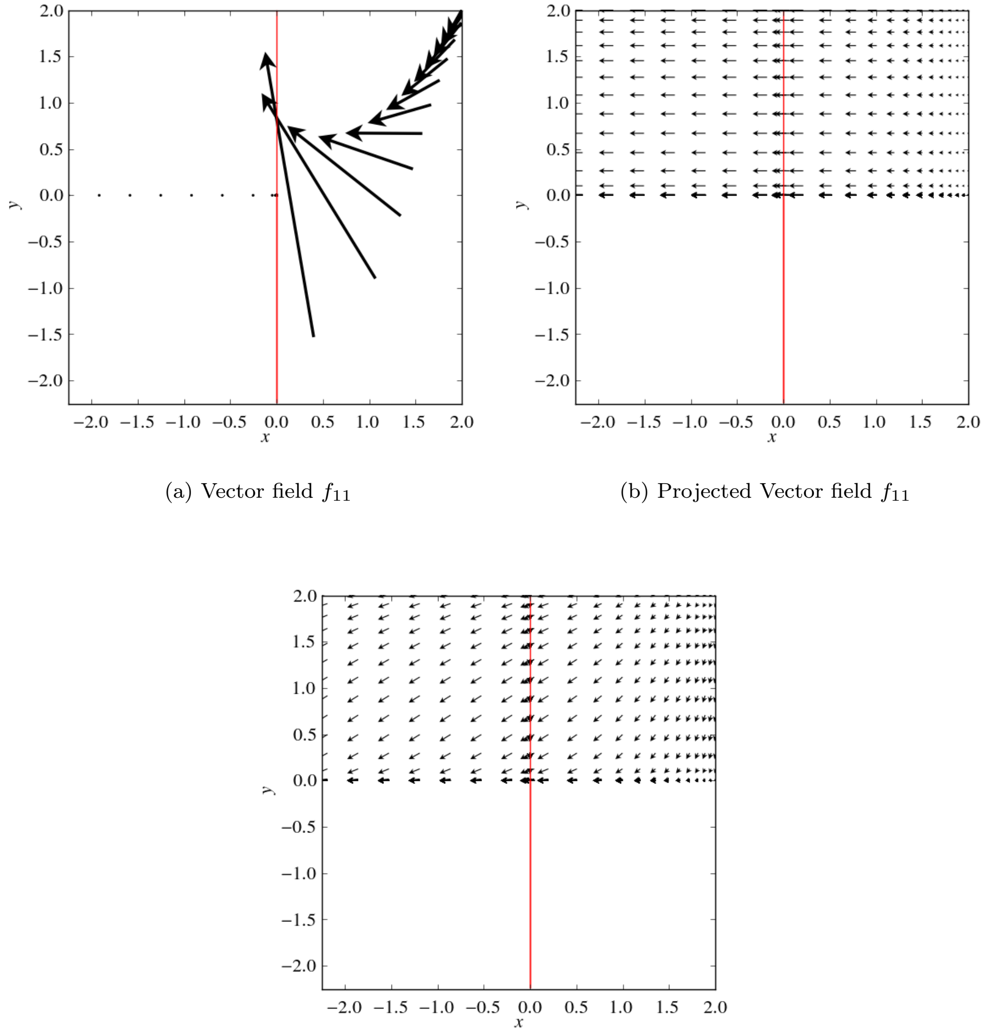
(d) ball sliding on parabola/plane, $\dot{x}$ vs x

(e) ball sliding on parabola/plane, $\ddot{y}$ vs x

(f) ball sliding on parabola/plane, $\ddot{x}$ vs x

Figure 4.14: parabola/plane example, dynamical variables with respect to x

their second derivatives are not equal. This leads to a jump in the accelerations and therefore in the contact forces, as showed in 4.14e and 4.14b. The vector fields of Fig. 4.15 show indeed a transverse mode where the velocities enable to cross the edge.



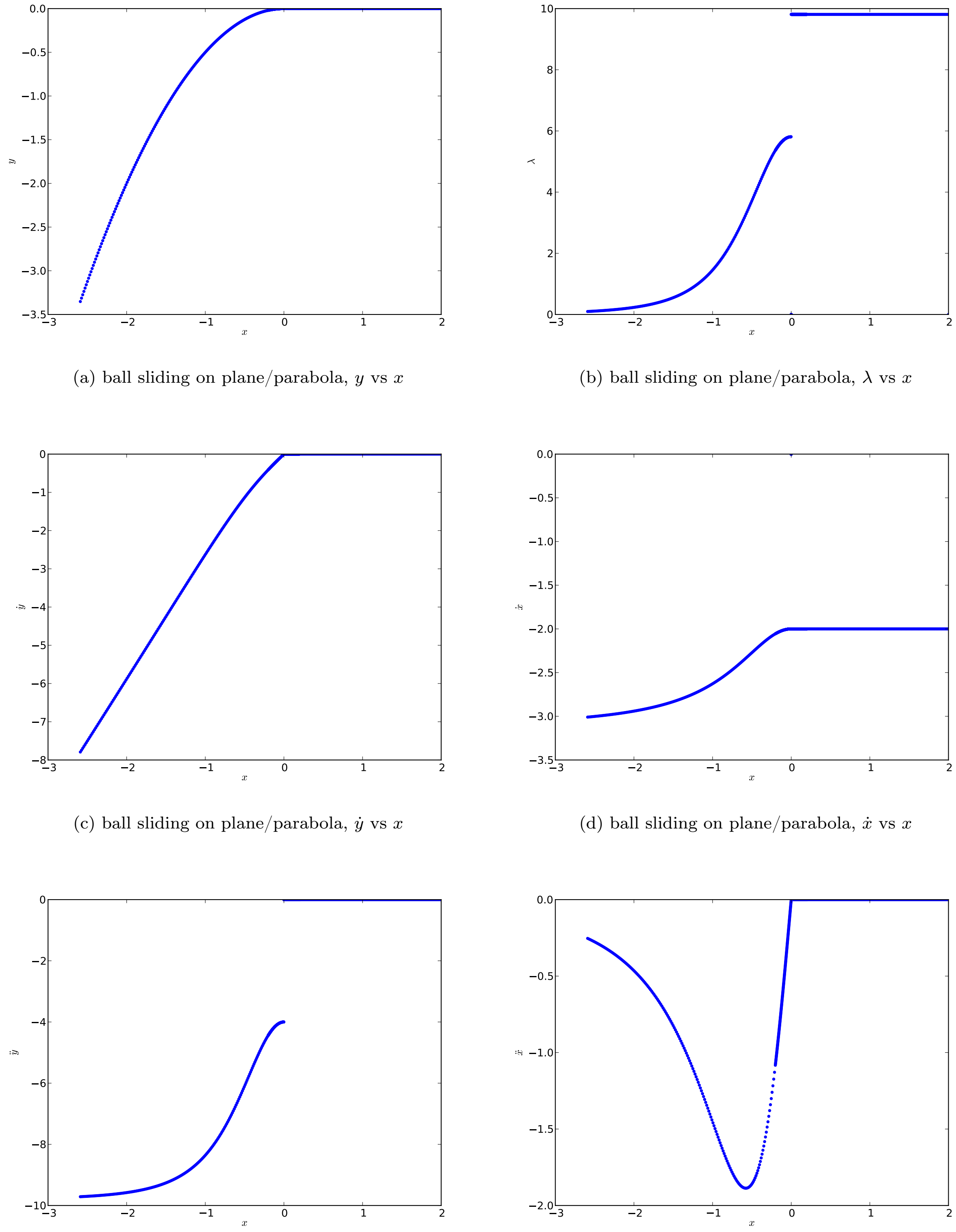
(a) Vector field f_{11}

(b) Projected Vector field f_{11}

(c) Velocities field

Figure 4.15: Vector fields describing the dynamics of the parabola/plane example

Example 2 Numerically, the edge is detected at $x \simeq -1.755.10^{-15}$. Figures 4.16a, 4.16b, 4.16c, 4.16d, 4.16e, and 4.16f depict the evolution of some variables with respect to time. We can draw the same conclu-



(a) ball sliding on plane/parabola, y vs x

(b) ball sliding on plane/parabola, λ vs x

(c) ball sliding on plane/parabola, $\dot{y}$ vs x

(d) ball sliding on plane/parabola, $\dot{x}$ vs x

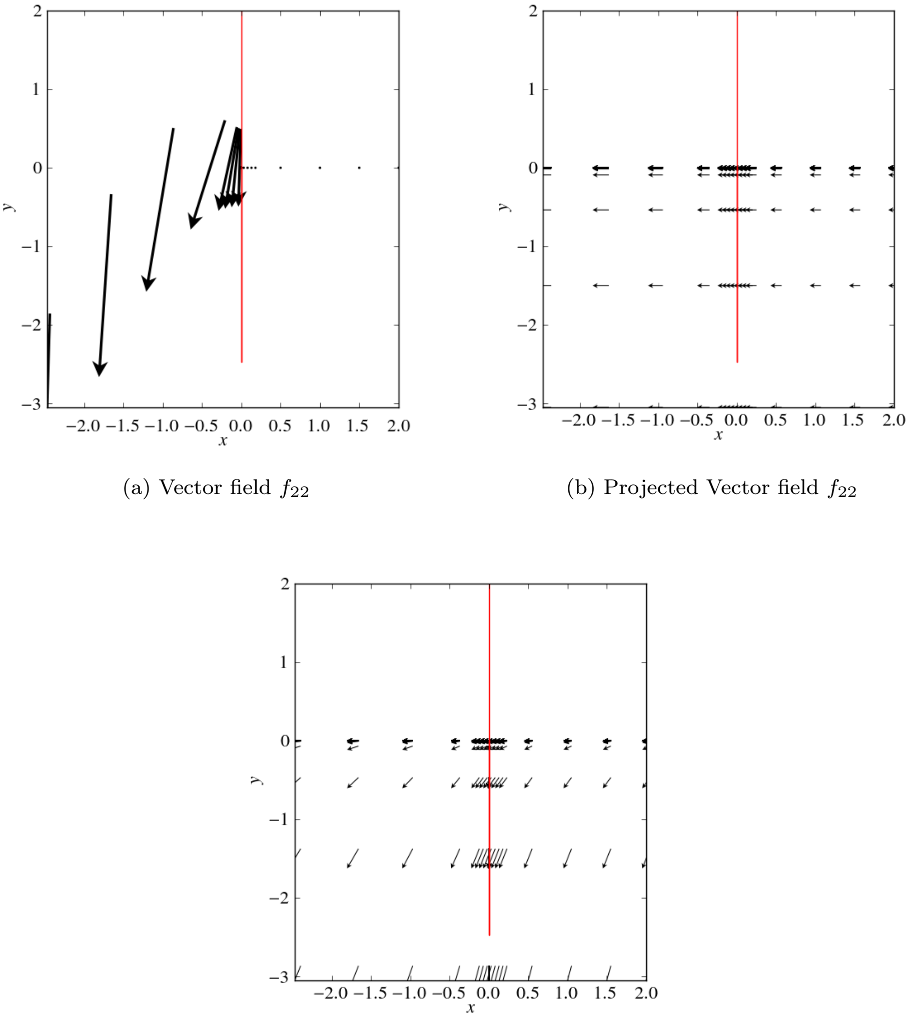
(e) ball sliding on plane/parabola, $\ddot{y}$ vs x

(f) ball sliding on plane/parabola, $\ddot{x}$ vs x

Figure 4.16: unversed parabola/plane example, dynamical variables with respect to x

sions as for the previous example. The velocities of the bead are continuous because the constraints to which the system is subjected are equal at the connection. However, we observe a jump on the $\ddot{y}$ component of the acceleration and on the Lagrange multiplier. These jumps result from the fact the hessians of the constraints are not equal at the edge.

Fig. 4.17 illustrates these vector fields when using the hybrid integration method for the simulation. These vector fields are consistent with a transverse mode, the velocity field in Fig. 4.17c allow for crossing the transition at $x = 0$.



(a) Vector field f_{22}

(b) Projected Vector field f_{22}

(c) Velocities field

Figure 4.17: Vector fields describing the dynamics of the plane/parabola example

4.4 Conclusion

In this chapter, we addressed several problems that are frequently met in practical applications of the multibody dynamics field. The problems refer to handling accumulation of events, friction and edge transitions in a CAD description. The proposed solution takes advantage of event-driven methods and time-stepping schemes. The algorithm is a mixed event-driven/time-stepping method where the event-driven is used with a high order DAE solver until a criterion indicating a switch condition is satisfied. The integration is then resumed with a time-stepping method. Conditions have been discussed in 4.2.2 that have to be satisfied to prevent the breakdown of the global order when switching from the event-driven to the time-stepping. In practice, the condition on the size of the time step of the time-stepping scheme cannot always be satisfied because it leads to very small time steps that prevent the simulation from advancing quickly. Therefore, in practical applications, we set a minimum time step size for the time-stepping method, even if the global order is not maintained.

Finally, several examples have been treated in order to prove the robustness and the numerical efficiency of the proposed hybrid scheme. Our solution has also been compared to a Moreau-Jean time stepping scheme with an adaptive strategy of the time step size. As expected, the hybrid integration is less time-consuming than the adaptive Moreau-Jean method since it can use large time steps because the incorporated DAE solver is of high order.

The implementation of the hybrid scheme is ongoing in the Ansys solver and the preliminary results make it one of the priorities for the future developments in the Ansys Rigid Body solver.

Chapter 5

Conclusion

This thesis is dedicated to improve and build numerical schemes for the simulation of the dynamics of nonsmooth multibody systems, i.e. multibody systems with impacts and friction. An extensive state of the art has been presented in Chapter 2 in which we discussed numerical schemes dedicated to the resolution of index-3, index-2 and index-1 DAEs. Based on this state of the art, several numerical schemes have been selected to run simulations over several sets of mechanical systems.

In chapter 3, four numerical schemes have been used to run simulations on several academic examples and a wide range of industrial benchmarks, in the context of event-diven schemes. Namely,

- the implicit generalized-α scheme in its index-3 and index-2 DAE forms,
- the HEM5 schemes, a half-explicit method for index-2 DAEs,
- the PHEM56 method, a half-explicit partitioned method, also dedicated to index-2 DAEs,
- and the RK4 and RKF methods for the resolution of ODEs.

These methods were compared from the point of view of their ability to enforce the constraints, their efficiency and their robustness in the case of stiff dynamics. The salient conclusions from the simulations are the following:

- the numerical experiments support the theory on the relevance of discretizing the constraints on the velocity level. Indeed, the schemes for index-2 DAEs (HEM5 and PHEM56) exactly solve the velocity constraints while keeping the violation of the position constraints to low levels. The generalized-α scheme in its index-3 DAE form solves exactly the position constraints, however we observe a high violation of the velocity constraints. The schemes for ODEs enforce the acceleration constraints while the position and velocity constraints are violated. However, the chosen methods (RK4 and RKF) are of high order (4), therefore when they are used with a "tight" tolerance on the integration error, they enforce the position and velocity constraints to acceptable levels. Let us mention that in the context of event-driven schemes, we are mainly interested in the position and velocity constraints to evaluate the index sets describing the status of the contacts. These index sets are sensitive to the defined thresholds. Therefore, the best method is the one that drastically reduces the violation of the constraints at both position and velocity. For this, index-2 DAE solvers have proved their superiority over other schemes. Of course one can propose to correct the violated constraints by some means of stabilization or projection techniques. Nevertheless, from a numerical point of view, this correction represents and additional computational work.

- index-2 DAE methods are more time-efficient than the other methods. Indeed, the selected methods (HEM5 and PHEM56) being of high order (5) and almost not requiring any correction of the constraints, are able to solve problems in a less time than the other schemes. The generalized-α in its index-2 DAE form is less time-consuming than in its index-3 DAE form. When it is used in its index-3 DAE form, the generalized-α needs to use short time steps sizes to maintain the drift at low levels. In addition, correction of the constraints are most of the time required when the system involves nonlinear constraints. RK4 and RKF proved to be efficient but less than HEM5 and PHEM56 because they require the use of methods to correct the constraints.

- when the dynamics is stiff, implicit methods prove their numerical superiority over the other methods. With its ability to introduce a numerical damping, the generalized-α method is able to solve the dynamics of systems with high frequencies with reasonable time step sizes. Half-explicit and explicit schemes either need very tight time steps to deal with a stiff dynamics, or completely fail when the involved frequencies are too high.

The second segment with this thesis deals with several problems that are frequent in the simulation of multibody systems: accumulation of impacts, friction and handling the numerical singularities due to imperfect geometries in a CAD description. To deal with these issues, we proposed a hybrid event-driven/time-stepping integration method. The aim is to profit from the advantages of both integration families (event-driven and time-stepping). This mixed integration consists in using the event-driven scheme with a high order DAE integrator until a switch criterion is satisfied, then the integration is resumed with the Moreau-Jean time-stepping scheme. In order not to reduce the high order of the event-driven scheme when using the Moreau-Jean method, this latter has to use small time steps satisfying the conditions established in Section 4.2.2. In practice, these conditions lead to very small time steps that prevent the simulation from advancing quickly. This is why we set a minimum value for the time steps of the time-stepping method, even if the order of the DAE solver incorporated in the event-driven method breaks down. We used this hybrid integration on several academic examples, and in some cases our algorithm has been compared to the Moreau-Jean scheme with an adaptive strategy for the time step size. The Moreau-Jean method having a local order 1, it needs much smaller time steps to achieve the tolerances on the integration error than the mixed event-driven/time stepping scheme.

As a perspective to this work, the hybrid method should be tested on industrial benchmarks, with several degrees of freedom and several contact points, to prove its robustness. Furthermore, the switch criteria have to be precised and adapted to an industrial development context. The implementation of this mixed algorithm is being performed in the Ansys solver, but this work is not advanced enough to run simulation on some industrial tests.

Appendix A

Discretization of some numerical method

A.1 Some Runge-Kutta schemes

There are a multitude of Runge-Kutta methods, but RK4, Runge-Kutta-Fehlberg and Dormand-Prince are probably one of the most popular. In the following, we present their discretization.

A.1.1 RK4

RK4 is based on 4 estimations of the derivatives as

$$\begin{cases} Y_1 = hf(y_n, t_n) \\ Y_2 = hf(y_n + \frac{1}{2}Y_1, t_n + \frac{h}{2}) \\ Y_3 = hf(y_n + \frac{1}{2}Y_2, t_n + \frac{h}{2}) \\ Y_4 = hf(y_n + Y_3, t_n + h). \end{cases} \tag{A.1}$$

The solution at the end of the time step is given with

$$y_{n+1} = y_n + \frac{1}{6}(Y_1 + 2Y_2 + 2Y_3 + Y_4). \tag{A.2}$$

A.1.2 Runge-Kutta-Fehlberg

This scheme is based on 6 estimations of the derivatives

$$\begin{cases} Y_1 = hf(y_n, t_n) \\ Y_2 = hf(y_n + \frac{1}{4}hY_1, t_n + \frac{1}{4}h) \\ Y_3 = hf(y_n + h(\frac{3}{32}Y_1 + \frac{9}{32}Y_2), t_n + \frac{3}{8}h) \\ Y_4 = hf(y_n + h(\frac{1932}{2197}Y_1 - \frac{7200}{2197}Y_2 + \frac{7296}{2197}Y_3), t_n + \frac{12}{13}h) \\ Y_5 = hf(y_n + h(\frac{439}{216}Y_1 - 8Y_2 + \frac{3680}{513}Y_3 - \frac{845}{4104}Y_4), t_n + h) \\ Y_6 = hf(y_n + h(-\frac{8}{27}Y_1 + 2Y_2 - \frac{3544}{2565}Y_3 + \frac{1859}{4104}Y_4 - \frac{11}{40}Y_5), t_n + h) \end{cases} \tag{A.3}$$

Then, a 4^{th} order approximation of the solution at the end of the step can be defined as

$$y_{n+1}^{(4)} = y_n + \frac{25}{216}Y_1 + \frac{1408}{2565}Y_3 + \frac{2197}{4101}Y_4 - \frac{1}{5}Y_5 \tag{A.4}$$

and a 5^{th} order one as

$$y_{n+1}^{(5)} = y_n + \frac{16}{135}Y_1 + \frac{6656}{12825}Y_3 + \frac{28561}{56430}Y_4 - \frac{9}{50}Y_5 + \frac{2}{55}Y_6. \tag{A.5}$$

There is a debate on whether the 4^{th} or the 5^{th} order estimation should be chosen as an approximation of the solution at the end of the time step. The 4^{th} order estimation seems to be more stable than the other one, therefore it will be chosen in the work presented in this report. The value $y_{n+1}^{(5)} - y_{n+1}^{(4)}$ is used to estimate the integration error and compute the optimal time step size, as explained in Section 2.2.7.4.

A.1.3 Dormand-Prince scheme

Dormand-Prince method is based on 7 estimations of the derivatives as follows

$$\begin{cases} Y_1 = hf(y_n, t_n) \\ Y_2 = hf(y_n + \frac{1}{5}hY_1, t_n + \frac{1}{5}h) \\ Y_3 = hf(y_n + h(\frac{3}{40}Y_1 + \frac{9}{40}Y_2), t_n + \frac{3}{10}h) \\ Y_4 = hf(y_n + h(\frac{44}{45}Y_1 - \frac{56}{15}Y_2 + \frac{32}{9}Y_3), t_n + \frac{4}{5}h) \\ Y_5 = hf(y_n + h(\frac{19372}{6561}Y_1 - \frac{25360}{2187}Y_2 + \frac{64448}{6561}Y_3 - \frac{212}{729}Y_4), t_n + \frac{8}{9}h) \\ Y_6 = hf(y_n + h(\frac{9017}{3168}Y_1 - \frac{355}{33}Y_2 - \frac{46732}{5247}Y_3 + \frac{49}{472}Y_4 - \frac{5103}{18656}Y_5), t_n + h) \\ Y_7 = hf(y_n + h(\frac{35}{384}Y_1 + \frac{500}{1113}Y_3 + \frac{125}{192}Y_4 - \frac{2187}{6784}Y_5 + \frac{11}{84}Y_6), t_n + h) \end{cases} \tag{A.6}$$

Order	Formula
1	$y_{n+1} = y_n + hf_{n+1}$
2	$y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf_{n+1}$
3	$y_{n+1} = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6}{11}hf_{n+1}$
4	$y_{n+1} = \frac{48}{25}y_n - \frac{36}{25}y_{n-1} + \frac{16}{25}y_{n-2} - \frac{3}{25}y_{n-3} + \frac{12}{25}hf_{n+1}$
5	$y_{n+1} = \frac{300}{137}y_n - \frac{300}{137}y_{n-1} + \frac{200}{137}y_{n-2} - \frac{75}{137}y_{n-3} + \frac{12}{137}y_{n-4} + \frac{30}{137}hf_{n+1}$
6	$y_{n+1} = \frac{360}{147}y_n - \frac{450}{147}y_{n-1} + \frac{400}{147}y_{n-2} - \frac{225}{147}y_{n-3} + \frac{72}{147}y_{n-4} - \frac{10}{147}y_{n-5} + \frac{60}{147}hf_{n+1}$

Table A.1: BDF schemes

As in the case of the RKF scheme, we have a 4^{th} and a 5^{th} order estimations defined with

$$y_{n+1}{}^{(4)} = y_n + \frac{35}{384}Y_1 + \frac{500}{1113}Y_3 + \frac{125}{195}Y_4 - \frac{2187}{6784}Y_5 + \frac{11}{84}Y_6, \tag{A.7}$$

and

$$y_{n+1}{}^{(5)} = y_n + \frac{5179}{576000}Y_1 + \frac{7571}{16695}Y_3 + \frac{393}{640}Y_4 - \frac{92097}{339200}Y_5 + \frac{187}{2100}Y_6 + \frac{1}{40}Y_7. \tag{A.8}$$

The difference between these two estimations is used to evaluate the integration error and to control the time step size.

A.2 Some multistep methods

There are three families of linear multistep methods: BDF schemes, Adams-Bashforth methods and Adams-Moulton methods. These methods are presented below for different accuracy orders.

A.2.1 BDF schemes

The BDF formulas for different orders are presented in Table A.1.

Order	Formula
1	$y_{n+1} = y_n + hf_n$
2	$y_{n+1} = y_n + h(\frac{3}{2}f_n - \frac{1}{2}f_{n-1})$
3	$y_{n+1} = y_n + h(\frac{23}{12}f_n - \frac{16}{12}f_{n-1} + \frac{5}{12}f_{n-2})$
4	$y_{n+1} = y_n + h(\frac{55}{24}f_n - \frac{59}{24}f_{n-1} + \frac{37}{24}f_{n-2} - \frac{9}{24}f_{n-3})$
5	$y_{n+1} = y_n + h(\frac{1901}{720}f_n - \frac{2774}{720}f_{n-1} + \frac{2616}{720}f_{n-2} - \frac{1274}{720}f_{n-3} + \frac{251}{720}f_{n-4})$

Table A.2: Adams-Bashforth formulas

A.2.2 Adams-Bashforth formulas

Adams-Bashforth formulas for different orders are presented in Table A.2

A.2.3 Adams-Moulton formulas

Adams-Moulton formulas for different orders are presented in Table A.3.

A.3 The Moreau-Jean algorithm

In this section we present the algorithm of the Moreau-Jean time-stepping scheme as implemented in the Ansys solver.

$G_B(q)$ and $G_U(q)$ stand for the jacobians of the bilateral and the unilateral constraints, respectively.

Algorithm 2 Moreau-Jean time-scheme implementation

```
Require: $q_n$, $v_n$
Require: $\theta$, $\gamma$
Require: $K_t = -\frac{\partial F(q,v,t)}{\partial q}$, $C_t = -\frac{\partial F(q,v,t)}{\partial v}$
Ensure: $q_{n+1}$, $v_{n+1}$
  $q = q_n$, $u = v_n$
  $i = 0$
  repeat
    $\hat{M} = M(q, t_{n+1}) + h\theta C(q, u, t_{n+1}) + h^2\theta^2 K(q, u, t_{n+1})$
    $\tilde{F} = (1-\theta)F(q_n, v_n, t_n) + \theta F(q, u, t_{n+1})$
    $R_0 = -M(u - v_n) + h\tilde{F}$
    $v_{free} = u + \hat{M}^{-1} R_0$
    Compute the index set $I_1 = \{i \,|g_i(q) = 0\}$
    if $I_1 \neq \emptyset$ then
      Solve a linear problem for the bilateral constraints
        $$\begin{pmatrix} \hat{M}(q,u) & G_B^T(q) \\ G_B(q) & 0 \end{pmatrix} \begin{pmatrix} \Delta V \\ \lambda_B \end{pmatrix} = \begin{pmatrix} \tilde{F} \\ -\dot{g}_B \end{pmatrix} \tag{A.9}$$
      Solve the nonsmooth problem (for a frictionless impact) for the unilateral impact :
        $$\begin{pmatrix} \hat{M}(q,u) & G_U^T(q) \\ G_U(q) & 0 \end{pmatrix} \begin{pmatrix} \Delta V \\ \lambda_U \end{pmatrix} = \begin{pmatrix} 0 \\ U_{free} + eU_n \end{pmatrix} \tag{A.10}$$
      with $U_{free} = G_U(q)v_{free}$ and $U_n = G_U(q_n)v_n$
    end if
    Update velocities: $u = v_{free} + \Delta V$ (if no unilateral constraints, then $\Delta V = 0$)
    Update positions: $q = q_n + h((1-\theta)v_n + \theta u)$
    Compute the dynamics residual: $R = -M(\tilde{q}, t_{n+1})(u - v_n) + h\big((1-\theta)F(q_n, v_n, t_n) + \theta F(q, u, t_{n+1})\big) + G^T(q)\lambda$
    Compute the error: $error = \parallel R \parallel_\infty$
    $i = i + 1$
  until $error < tol$
  Update the state: $q_{n+1} = q$, $v_{n+1} = u$
```

Order	Formula
1	$y_{n+1} = y_n + hf_{n+1}$
2	$y_{n+1} = y_n + h(\frac{1}{2}f_{n+1} + \frac{1}{2}f_n)$
3	$y_{n+1} = y_n + h(\frac{5}{12}f_{n+1} + \frac{8}{12}f_n - \frac{1}{12}f_{n-1})$
4	$y_{n+1} = y_n + h(\frac{9}{24}f_{n+1} + \frac{19}{24}f_n - \frac{5}{24}f_{n-1} + \frac{1}{24}f_{n-2})$
5	$y_{n+1} = y_n + h(\frac{251}{720}f_{n+1} + \frac{646}{720}f_n - \frac{264}{720}f_{n-1} + \frac{106}{720}f_{n-2} - \frac{19}{720}f_{n-3})$

Table A.3: Adams-Moulton formulas

Appendix B

A few words about the industrial code implementation

The work presented in this thesis is based on a python code which I developed, as well as a C++ implementation in the Ansys Rigid Body Dynamics solver performed together with the team in Ansys.

B.1 Python implementation

The python code is based on major 10 scripts:

- *LagrangianSystem* contains the class defining a Lagrangian system (inertia matrix, applied forces, position, velocity...),
- *IndexSets* contains the definition of the index sets defined in Section 2.1.1.1,
- *DenseOutput* contains the methods to interpolate the position and velocity constraints as well as the methods to find the time at which these variables vanish, as the dichotomy method for example,
- *HEM5*,*PHEM56*, *RKF* and *GeneralizedAlpha* which contain the discretizations of the HEM5, PHEM56, Runge-Kutta-Fehlberg and generalized-α methods, respectively,
- *EventDriven* contains the event-driven strategy. It calls the DAE method requested by the user, and the *LagrangianSystem*, *IndexSets*, *DenseOutput* scripts,
- *MoreauJean* contains the Moreau-Jean time-stepping method,

- *MixedEDTS*, contains the hybrid event-driven/time-stepping method. It contains the switch conditions discussed in Chapter 2 and uses the *EventDriven* and *MoreauJean* python scripts.

To run a simulation for a mechanism, a python script must be created which contains the characteristics of the system: inertia matrix, forces, constraints, jacobian of the constraints and the second derivatives of the constraints. The user also defines the initial conditions, the initial time step size, the accuracy tolerance and the integration strategy (event-driven, time-stepping, hybrid event-driven/time-stepping).

B.2 Implementation in the Ansys Rigid Body software

About 6 months of my thesis consisted in implementing the HEM5, generalized-α and the Moreau-Jean methods in the Ansys software. The work has been performed together with the team in charge of the development of the Rigid Body Dynamics module of Ansys. The generalized-α method is now available in the 16.0 commercial release of the Ansys software. The implementation of the hybrid integration method is ongoing and the preliminary results obtained on academic benchmarks make it sufficiently promising to be part of the priority future developments to do in the Rigid Body Dynamics module.

Bibliography

[1] M. Abadie. Dynamic Simulation of Rigid Bodies: Modelling of Frictional Contact. In B. Brogliato, editor, *Impacts in Mechanical Systems*, volume 551 of *Lecture Notes in Physics*, pages 61–144. Springer, 2000.

[2] V. Acary. Higher order event capturing time–stepping schemes for nonsmooth multibody systems with unilateral constraints and impacts. *Applied Numerical Mathematics*, 62:1259–1275, 2012.

[3] V. Acary. Projected event-capturing time-stepping schemes for nonsmooth mechanical systems with unilateral contact and Coulomb's friction. *Computer Methods in Applied Mechanics and Engineering*, 256:224–250, 2013.

[4] V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics*, volume 35 of *Lecture Notes in Applied and Computational Mechanics*. Springer Verlag, 2008.

[5] V. Acary and F. Cadoux. Applications of an existence result for the Coulomb friction problem. In Georgios E. Stavroulakis, editor, *Recent Advances in Contact Mechanics*, volume 56 of *Lecture Notes in Applied and Computational Mechanics*, pages 45–66. Springer, 2013.

[6] V. Acary, F. Cadoux, C. Lemaréchal, and J. Malick. A formulation of the linear discrete Coulomb friction problem via convex optimization. *Zeitschrift für Angewandte Mathematik und Mechanik*, 91(2):155–175, 2011.

[7] L.-E. Andersson. Existence results for quasistatic contact problems with Coulomb friction. *Applied Mathematics and Optimization*, 42(2):169–202, 2000.

[8] M. Anitescu and G.D. Hart. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *International Journal for Numerical Methods in Engineering*, 60(14):2335–2371, 2004.

[9] M. Anitescu, F. Potra, and D. Stewart. Time-stepping for three-dimensional rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 177(3):183–197, 1999.

[10] M. Anitescu and F.A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, 1997.

[11] M. Anitescu and F.A. Potra. A time-stepping method for stiff multibody dynamics with contact and friction. *International Journal for Numerical Methods in Engineering*, 55(7):753–784, 2002.

[12] M. Arnold. Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2. *BIT Numerical Mathematics*, 38(3):415–438, 1998.

[13] M. Arnold and A. Murua. Non-stiff integrators for differential–algebraic systems of index 2. *Numerical Algorithms*, 19(1-4):25–41, 1998.

[14] P. Ballard. A counter-example to uniqueness in quasi-static elastic contact problems with small friction. *International Journal of Engineering Science*, 37(2):163–178, 1999.

[15] P. Ballard and S. Basseville. Existence and uniqueness for dynamical unilateral contact with Coulomb friction : a model problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):59–78, 2005.

[16] O.A. Bauchau. *Flexible Multibody Dynamics*, volume 176. Springer Science+ Business Media, 2011.

[17] O.A. Bauchau and A. Laulusa. Review of contemporary approaches for constraint enforcement in multibody systems. *Journal of Computational and Nonlinear Dynamics*, 3(1):011005, 2008.

[18] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1(1):1–16, 1972.

[19] E.J. Berger. Friction modeling for dynamic system simulation. *Applied Mechanics Reviews*, 55(6):535–577, 2002.

[20] W. Blajer, W. Schiehlen, and W. Schirm. A projective criterion to the coordinate partitioning method for multibody dynamics. *Archive of Applied Mechanics*, 64(2):86–98, 1994.

[21] A. Blumentals, B. Brogliato, and F. Bertails-Descoubes. The contact problem in Lagrangian systems subject to bilateral and unilateral constraints, with or without sliding Coulomb's friction: A tutorial. Technical report, 2014.

[22] M. Borri, C.L. Bottasso, and P. Mantegazza. Equivalence of Kane's and Maggi's equations. *Meccanica*, 25(4):272–274, 1990.

[23] C.L. Bottasso, O. A. Bauchau, and A. Cardona. Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations. *SIAM Journal on Scientific Computing*, 29(1):397–414, 2007.

[24] C.L. Bottasso, D. Dopico, and L. Trainelli. On the optimal scaling of index three DAEs in multibody dynamics. *Multibody System Dynamics*, 19(1-2):3–20, 2008.

[25] V. Brasey. A Half-explicit Runge-Kutta method of order 5 for solving constrained mechanical systems. *Computing*, 48(2):191–201, 1992.

[26] V. Brasey. HEM5 User's guide. Technical report, Université de Genève, 1994.

[27] V. Brasey. *Méthodes demi-explicites pour équations différentielles algébriques semi-explicites d'index 2.* PhD thesis, 1994.

[28] V. Brasey and E. Hairer. Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *SIAM Journal on Numerical Analysis*, 30(2):538–552, 1993.

[29] K.E. Brenan, S.L. Campbell, and L. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, volume 14. Siam, 1996.

[30] B. Brogliato. *Nonsmooth Mechanics: Models, Dynamics and Control.* Springer, 1999.

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

[32] O. Brüls and M. Arnold. Convergence of the generalized-α scheme for constrained mechanical systems. *Multibody System Dynamics*, 18(2):185–202, 2007.

[33] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods.* Wiley-Interscience, New York, NY, USA, 1987.

[34] A. Cardona and M. Géradin. Time integration of the equations of motion in mechanism analysis. *Computers & Structures*, 33(3):801–820, 1989.

[35] Q.Z. Chen, V. Acary, Ge. Virlez, and O. Brüls. A nonsmooth generalized-alpha scheme for flexible multibody systems with unilateral constraints. *International Journal for Numerical Methods in Engineering*, 96(8):487–511, November 2013.

[36] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.

[37] L. Dieci and L. Lopez. Sliding motion in Filippov differential systems: Theoretical results and a computational approach. *SIAM Journal on Numerical Analysis*, 47(3):2023–2051, 2009.

[38] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[39] R. Dzonou, M.DP Monteiro Marques, and L. Paoli. A convergence result for a vibro-impact problem with a general inertia operator. *Nonlinear Dynamics*, 58(1-2):361–384, 2009.

[40] R. Dzonou and M. DP Monteiro Marques. A sweeping process approach to inelastic contact problems with general inertia operators. *European Journal of Mechanics-A/Solids*, 26(3):474–490, 2007.

[41] Ch. Eck, J. Jarusek, and M. Krbec. *Unilateral contact problems: variational methods and existence theorems.* CRC Press, 2005.

[42] E. Eich-Soellner and C. Führer. *Numerical Methods in Multibody Dynamics.* Teubner, 1998. Reprint Lund university 2002.

[43] B. Esefeld and H. Ulbrich. A hybrid integration scheme for nonsmooth mechanical systems. In *Multibody Dynamics 2011, Eccomas Thematic Conference, Brussels, 4th July until 7th July 2011*, 2011.

[44] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research*, 2(1):13–30, 1983.

[45] R. Featherstone and D. Orin. Robot dynamics: Equations and algorithms. In *ICRA*, pages 826–834, 2000.

[46] A.F. Filippov. *Differential equations with discontinuous right-hand sides.* Mathematics and Its Applications, Kluwer Academic, Dordrecht, 1988.

[47] P. Fisette and B. Vaneghem. Numerical integration of multibody system dynamic equations using the coordinate partitioning method in an implicit newmark scheme. *Computer Methods in Applied Mechanics and Engineering*, 135(1):85–105, 1996.

[48] P. Flores, R. Leine, and C. Glocker. Modeling and Analysis of Rigid Multibody Systems with Translational Clearance Joints Based on the Nonsmooth Dynamics Approach. In Krzysztof Arczewski, Wojciech Blajer, Janusz Fraczek, and Marek Wojtyra, editors, *Multibody Dynamics*, volume 23 of *Computational Methods in Applied Sciences*, page 107â130. Springer Netherlands, 2011.

[49] C.Glocker F.Pfeiffer. *Multibody Dynamics with Unilateral Contacts.* Wiley-Interscience, 1996.

[50] K. Funk and F. Pfeiffer. A time-stepping algorithm for stiff mechanical systems with unilateral constraints. *PAMM*, 2(1):228–229, 2003.

[51] J. García de Jalón. Twenty-five years of natural coordinates. *Multibody System Dynamics*, 18(1):15–33, 2007.

[52] J. García de Jalón and E. Bayo. *Kinematic and Dynamic of Multibody Systems: the Real–Time Challenge.* Springer Verlag, 1994.

[53] J. García de Jalón, A. Callejo, and A.F. Hidalgo. Efficient solution of Maggi's equations. *Journal of Computational and Nonlinear Dynamics*, 7(2):021003–021012, 2012.

[54] J. García de Jalón and M.D. Gutiérrez-López. Multibody dynamics with redundant constraints and singular mass matrix: existence, uniqueness, and determination of solutions for accelerations and constraint forces. *Multibody System Dynamics*, pages 1–31, 2013.

[55] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1971.

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

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

[58] C Glocker. Formulation of spatial contact situations in rigid multibody systems. *Computer Methods in Applied Mechanics and Engineering*, 177(3):199–214, 1999.

[59] C. Glocker. Velocity jumps induced by c0-constraints. In *Proceedings of the 1999 ASME Design Engineering Technical Conferences, Las Vegas, Nevada*, 1999.

[60] C. Glocker, E. Cataldi-Spinola, and R. Leine. Curve squealing of trains: Measurement, modelling and simulation. *Journal of Sound and Vibration*, 324(1):365–386, 2009.

[61] Ch. Glocker. *Set-valued force laws: dynamics of non-smooth systems*, volume 1. Springer Science & Business Media, 2001.

[62] Ch. Glocker and F. Pfeiffer. Complementarity problems in multibody systems with planar friction. *Archive of Applied Mechanics*, 63(7):452–463, 1993.

[63] S. Greenhalgh, V. Acary, and B. Brogliato. On preserving dissipativity properties of linear complementarity dynamical systems with the θ-method. *Numerische Mathematik*, 125(4):601–637, December 2013.

[64] M. Haddouni, V. Acary, and J.D. Beley. Comparison of index-3, index-2 and index-1 DAE solvers for nonsmooth multibody systems with unilateral and bilateral constraints. In *Multibody Dynamics 2013*, Zagreb, Croatia, July 2013. Eccomas.

[65] M. Haddouni, V. Acary, J.D. Beley, and S.e Garreau. Comparison of index-2 and index-1 DAE solvers for nonsmooth multibody systems with unilateral or bilateral constraints. In *CSMA 2013 - 11e Colloque National en Calcul des Structures*, Giens, France, May 2013. CSMA.

[66] E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I, Nonstiff Problems*. Springer, 2000.

[67] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer, 2002.

[68] J. Haslinger. Least square method for solving contact problems with friction obeying the Coulomb law. *Applications of Mathematics*, 29(3):212–224, 1984.

[69] J. Haslinger and J.C. Nedelec. Approximation of the Signorini problem with friction, obeying the Coulomb law. *Mathematical Methods in the Applied Sciences*, 5(1):422–437, 1983.

[70] E.J. Haug, D. Negrut, and C. Engstler. Implicit Runge-Kutta integration of the equations of multibody dynamics in descriptor form. *Journal of Structural Mechanics*, 27(3):337–364, 1999.

[71] E.J. Haug, D. Negrut, and M. Iancu. A state-space-based implicit integration algorithm for differential-algebraic equations of multibody dynamics. *Journal of Structural Mechanics*, 25(3):311–334, 1997.

[72] N.J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.

[73] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3):283–292, 1977.

[74] O. Janin and C. H. Lamarque. Comparison of several numerical methods for mechanical systems with impacts. *International Journal for Numerical Methods in Engineering*, 51(9):1101–1132, 2001.

[75] J. Jarušek. Contact problems with bounded friction. Coercive case. *Czechoslovak Mathematical Journal*, 33(2):237–261, 1983.

[76] J. Jarušek. Contact problems with bounded friction. Semicoercive case. *Czechoslovak Mathematical Journal*, 34(4):619–629, 1984.

[77] J. Jarušek and Ch. Eck. Dynamic contact problems with small Coulomb friction for viscoelastic bodies: Existence of solutions. *Mathematical Models and Methods in Applied Sciences*, 9(1):11–34, 1999.

[78] L.O. Jay and D. Negrut. Extensions of the HHT-method to differential-algebraic equations in mechanics. *Electronic Transactions on Numerical Analysis*, 26:190–208, 2007.

[79] L.O. Jay and D. Negrut. A second order extension of the generalized–α method for constrained systems in mechanics. In C. L. Bottasso, editor, *Multibody Dynamics. ECCOMAS Thematic Conference on Multibody Dynamics*, volume 12 of *Computational Methods in Applied Sciences*, pages 143–158. Springer Netherlands, 2008.

[80] M. Jean. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3):235–257, 1999.

[81] M. Jean, V. Acary, and Y. Monerie. Non-smooth contact dynamics approach of cohesive materials. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1789):2497–2518, 2001.

[82] M. Jean and J.J. Moreau. Dynamics in the Presence of Unilateral Contacts and Dry Friction: A Numerical Approach. In G. Del Piero and F. Maceri, editors, *Unilateral Problems in Structural Analysis â 2*, volume 304 of *International Centre for Mechanical Sciences*, pages 151–196. Springer Vienna, 1987.

[83] M. Jean and J.J. Moreau. Unilaterality and dry friction in the dynamics of rigid body collections. In *Proceedings of Contact Mechanics International Symposium*, pages 31–48. Presses Polytechniques et Universitaires Romandes, 1992.

[84] A. Klarbring. A mathematical programming approach to three-dimensional contact problems with friction. *Computer Methods in Applied Mechanics and Engineering*, 58(2):175–200, 1986.

[85] A. Klarbring and J.S. Pang. Existence of solutions to discrete semicoercive frictional contact problems. *SIAM Journal on Optimization*, 8(2):414–442, 1998.

[86] S. Krenk. Energy conservation in Newmark based time integration algorithms. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6110–6124, 2006.

[87] M. Lalanne and G. Ferraris. *Rotordynamics Prediction in Engineering, 1998*. Wiley, New York.

[88] A. Laulusa and O. A. Bauchau. Review of classical approaches for constraint enforcement in multibody systems. *Journal of Computational and Nonlinear Dynamics*, 3(1):011004, 2008.

[89] R. Leine, D.H. Van Campen, and Ch. Glocker. Nonlinear dynamics and modeling of various wooden toys with impact and friction. *Journal of Vibration and Control*, 9(1-2):25–78, 2003.

[90] R.I. Leine, B. Brogliato, and H. Nijmeijer. Periodic motion and bifurcations induced by the Painlevé paradox. *European Journal of Mechanics-A/Solids*, 21(5):869–896, 2002.

[91] C. Lunk and B. Simeon. Solving constrained mechanical systems by the family of Newmark and α–methods. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 86(10):772–784, 2006.

[92] R. Mannshardt. One-step methods of any order for ordinary differential equations with discontinuous right-hand sides. *Numerische Mathematik*, 31(2):131–152, 1978.

[93] L. Mariti, N. P. Belfiore, E. Pennestri, and P. P. Valentini. Comparison of solution strategies for multibody dynamics equations. *International Journal for Numerical Methods in Engineering*, 88(7):637–656, 2011.

[94] N.H. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. *Automatic Control, IEEE Transactions*, 33(5):419–426, 1988.

[95] J.J. McPhee. On the use of linear graph theory in multibody system dynamics. *Nonlinear Dynamics*, 9(1-2):73–90, 1996.

[96] J.J. McPhee and S.M. Redmond. Modelling multibody systems with indirect coordinates. *Computer Methods in Applied Mechanics and Engineering*, 195(50):6942–6957, 2006.

[97] M. Möller and Ch. Glocker. Rigid body dynamics with a scalable body, quaternions and perfect constraints. *Multibody System Dynamics*, 27(4):437–454, 2012.

[98] J.J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In J.J. Moreau and P.D. Panagiotopoulos, editors, *Nonsmooth Mechanics and Applications*, volume 302 of *International Centre for Mechanical Sciences*, pages 1–82. Springer Vienna, 1988.

[99] J.J. Moreau. Numerical aspects of the sweeping process. *Computer Methods in Applied Mechanics and Engineering*, 177(3):329–349, 1999.

[100] J.J. Moreau. Facing the plurality of solutions in nonsmooth mechanics. *Nonsmooth/Nonconvex Mechanics with Applications in Engineering, II. NNMAE*, 2006:3–12, 2006.

[101] A. Murua. Partitioned half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *Computing*, 59(1):43–61, 1997.

[102] D. Negrut, E.J. Haug, and H.C. German. An implicit Runge–Kutta method for integration of differential-algebraic equations of multibody dynamics. *Multibody System Dynamics*, 9(2):121–142, 2003.

[103] D. Negrut, R. Rampalli, G. Ottarsson, and A. Sajdak. On the use of the HHT method in the context of index 3 differential algebraic equations of multi-body dynamics. In *Proc. of the ECCOMAS Conf. on Advances in Computational Multibody Dynamics, Madrid, Spain*, 2005.

[104] N.M. Newmark. A method of computation for structural dynamics. In *Proc. ASCE*, volume 85, pages 67–94, 1959.

[105] J. Nocedal and S. J.Wright. *Numerical Optimization*, volume 2. Springer New York, 1999.

[106] J.S. Pang and D.E. Stewart. A unified approach to discrete frictional contact problems. *International Journal of Engineering Science*, 37(13):1747–1768, 1999.

[107] J.S. Pang and J.C. Trinkle. Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with coulomb friction. *Mathematical Programming*, 73(2):199–226, 1996.

[108] L. Paoli. A proximal-like method for a class of second order measure-differential inclusions describing vibro-impact problems. *Journal of Differential Equations*, 250(1):476–514, 2011.

[109] L. Paoli and M. Schatzman. A numerical scheme for impact problems i: The one-dimensional case. *SIAM Journal on Numerical Analysis*, 40(2):702–733, 2002.

[110] L. Paoli and M. Schatzman. A numerical scheme for impact problems ii: The multidimensional case. *SIAM Journal on Numerical Analysis*, 40(2):734–768, 2002.

[111] F. Pfeiffer. The idea of complementarity in multibody dynamics. *Archive of Applied Mechanics*, 72(11-12):807–816, 2003.

[112] F. Pfeiffer, M. Foerg, and H. Ulbrich. Numerical aspects of non-smooth multibody dynamics. *Computer Methods in Applied Mechanics and Engineering*, 195(50):6891–6908, 2006.

[113] F. Pfeiffer and M. Hajek. Stick-slip motion of turbine blade dampers. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 338(1651):503–517, 1992.

[114] M. Renouf, V. Acary, and G. Dumont. 3D frictional contact and impact multibody dynamics. A comparison of algorithms suitable for real-time applications. In *ECCOMAS Thematic Conference Multibody Dynamics 2005*, Madrid, June 2005.

[115] R. Rocca and M. Cocu. Existence and approximation of a solution to quasistatic Signorini problem with local friction. *International Journal of Engineering Science*, 39(11):1233–1255, 2001.

[116] T. Roßmann, F. Pfeiffer, and Ch. Glocker. Efficient algorithms for non-smooth dynamics. In *Proceedings of: ASME International Mechanical Engineering Congress and Exposition, Dallas, Texas*, pages 115–122. ASME, 1997.

[117] T. Schindler and V. Acary. Application of timestepping schemes based on time discontinuous Galerkin methods to multi-dimensional examples. In *Euromech 514 - New trends in Contact Mechanics*, Cargèse, France, March 2012. Michel Raous, Peter Wriggers.

[118] T. Schindler and V. Acary. Timestepping schemes for nonsmooth dynamics based on discontinuous Galerkin methods: definition and outlook. *Mathematics and Computers in Simulation*, 95:180–199, 2014.

[119] T. Schindler, S. Rezaei, J. Kursawe, and V. Acary. Half-explicit timestepping schemes on velocity level based on time-discontinuous Galerkin methods. Research Report 8623, INRIA Grenoble, October 2014.

[120] B. Simeon. Modelling a flexible slider crank mechanism by a mixed system of DAEs and PDEs. *Mathematical Modelling of Systems*, 2(1):1–18, 1996.

[121] B. Simeon. *Computational Flexible Multibody Dynamics: a Differential-Algebraic Approach.* Springer, 2013.

[122] D. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[123] D.E. Stewart. Existence of solutions to rigid body dynamics and the Painlevé paradoxes. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6):689–693, 1997.

[124] D.E. Stewart. Rigid-body dynamics with friction and impact. *SIAM review*, 42(1):3–39, 2000.

[125] D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with Coulomb friction. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 162–169. IEEE, 2000.

[126] C. Studer and C. Glocker. Simulation of non-smooth mechanical systems with many unilateral constraints. *Proceedings ENOC-2005 Eindhoven*, 2005.

[127] C. Studer, R. Leine, and C. Glocker. Step size adjustment and extrapolation for time-stepping schemes in non-smooth dynamics. *International Journal for Numerical Methods in Engineering*, 76(11):1747–1781, 2008.

[128] F.E. Udwadia. On constrained motion. *Applied Mathematics and Computation*, 164(2):313–320, 2005.

[129] F.E. Udwadia and R.E. Kalaba. A new perspective on constrained motion. *Proceedings of the Royal Society of London: Mathematical, Physical and Engineering Sciences*, 439(1906):407–410, 1992.

[130] R.A. Wehage and E.J. Haug. Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. *Journal of Mechanical Design*, 104(1):247–255, 1982.

[131] R.A. Wehage, E.J. Haug, and R.R. Beck. Generalized coordinate partitioning in dynamic analysis of mechanical systems. Technical report, DTIC Document, 1981.

[132] J. Wittenburg. *Dynamics of Multibody Systems*. Springer, 2008.

[133] J. Yen, L. Petzold, and S. Raha. A time integration algorithm for flexible mechanism dynamics: The DAE α-method. *Computer methods in applied mechanics and engineering*, 158(3):341–355, 1998.