



**HAL**  
open science

# Comparison of several formulations and integration methods for the resolution of DAEs formulations in event-driven simulation of nonsmooth frictionless multibody dynamics

Mounia Haddouni, Vincent Acary, Stéphane Garreau, Jean-Daniel Beley,  
Bernard Brogliato

## ► To cite this version:

Mounia Haddouni, Vincent Acary, Stéphane Garreau, Jean-Daniel Beley, Bernard Brogliato. Comparison of several formulations and integration methods for the resolution of DAEs formulations in event-driven simulation of nonsmooth frictionless multibody dynamics. *Multibody System Dynamics*, 2017, pp.1-31. 10.1007/s11044-017-9570-y . hal-01562700v1

**HAL Id: hal-01562700**

**<https://inria.hal.science/hal-01562700v1>**

Submitted on 17 Jul 2017 (v1), last revised 11 Sep 2017 (v2)

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

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

# Comparison of several formulations and integration **methods** for the resolution of DAEs formulations in event-driven simulation of nonsmooth frictionless multibody dynamics

Mounia Haddouni<sup>2</sup>, Vincent Acary<sup>1</sup>, Stéphane Garreau<sup>2</sup>, Jean-Daniel Beley<sup>2</sup>, and Bernard Brogliato<sup>1</sup>

<sup>1</sup>INRIA Bipop team

<sup>2</sup>ANSYS, France

<sup>1</sup>{*vincent.acary, bernard.brogliato*}@inria.fr

<sup>2</sup>{*mounia.haddouni, stephane.garreau, jean-daniel.beley*}@ansys.com

March 6, 2017

## Abstract

This article is devoted to the comparison of numerical integration methods for nonsmooth multibody dynamics with joints, unilateral contacts and impacts in an industrial context. With an event-driven strategy, the smooth dynamics, which is integrated between two events, may be equivalently formulated as a Differential Algebraic Equation (DAE) of index 1, 2 or 3. It is well-known that these reformulations are no longer equivalent when a numerical time-integration technique is used. The drift-off effect and the stability of the numerical scheme strongly depend on the index of the formulation. But, besides the standard properties of accuracy and stability of the DAE solvers, the event-driven context imposes some further requirements that are crucial for a robust and efficient event-driven strategy. In this article, several state-of-the-art numerical time integration methods for each formulation are compared: the generalized- $\alpha$  scheme for index-3 formulation and stabilized index-2 formulation, (Partitioned) Runge-Kutta Half-Explicit Method of order 5 (HEM5 and PHEM56) for index-2 DAEs with projection techniques, and Runge-Kutta explicit scheme of order 5, the Dormand-Prince scheme (DOPRI5), for index-1 DAEs with projection techniques (MDOP5). We compare these schemes in terms of efficiency, violation of the constraints and the way they handle stiff dynamics on numerous industrial benchmarks, where a CAD software is in this loop. One of the major conclusions is that the index-2 DAEs solvers prove to be better than other schemes to maintain low violations at position and acceleration levels. The best compromise allows us to design efficient event-driven solvers. When the dynamics is stiff, implicit schemes outperform explicit and half-explicit methods which

are sometimes unable to compute the dynamics when the system's frequency range is wide. Furthermore, in industrial context, some solvers fail to reproduce the properties that they enjoy in theory. This is particularly true for half-explicit schemes when the Jacobian of the constraints has not full rank.

## 1 Introduction

The aim of this article is the comparison of numerical integration methods of nonsmooth multibody dynamics with joints, unilateral contacts and impacts in an industrial context. There are two major integration strategies that are used to solve the nonsmooth dynamics of multi-body dynamical systems: *event-detecting* time-stepping schemes (a.k.a. event-driven schemes) and *event-capturing* schemes (a.k.a. simply time-stepping schemes). In time-stepping schemes [?, ?, ?, ?, ?], the formulation of the dynamics in terms of measure differential inclusions enables to simultaneously handle the smooth dynamics and the non-smooth events. In event-driven schemes, the events at discrete times (impact, take-off, transitions sliding/sticking if friction is taken into account) and the non impulsive dynamics are separated. The detection of the occurrence of an event is a crucial step. The smooth dynamics between two events may be equivalently formulated as Differential Algebraic Equations (DAE) of index 1, 2 or 3. The resulting formulation is then integrated with any classical numerical scheme [?, ?]. We refer to [?, ?] for a detailed comparison between these two strategies. In this paper, we choose to study the numerical time-integration in the context of an event-driven strategy. This choice is motivated by the needs of high accuracy simulations of the smooth phases between two events.

It is well-known that the reformulations as index 1, 2 or 3 DAEs are no longer equivalent when a time-integration technique is used [?]. The drift-off effect and the stability of the numerical scheme strongly depend on the index of the formulation. But, besides the standard properties of accuracy and stability of the DAE solvers, the event-driven context imposes some further requirements that are crucial for a robust and efficient event-driven strategy. For multibody systems with contact, the index reduction technique, obtained by differentiating the constraints, requires sometimes further smoothness assumptions of the constraints that are not necessarily satisfied in practical situations with contact. From this point of view, low kinematic level formulations, that is, high index formulation, are more suitable. Furthermore, in an event-driven strategy, unilateral constraints are classified into several index sets that describe the status of the contact: open, closed, closed with a vanishing velocity, *etc.* The correct approximations and stability of the index sets depend on the ability of the numerical time-stepping schemes to control the violations of both position and velocity level constraints and the accuracy and the stability of the Lagrange multipliers, that are related to the constraints at the acceleration level.

### 1.1 State of the art

The application of event-driven approach to the nonsmooth multibody systems with contacts, using any Ordinary Differential Equations (ODE) or DAE **solver** for the non impulsive phases and using an accurate detection of the events thanks to root

finding procedures [?], dates back to the 1990's. In [?], a variable structure system is introduced that corresponds to the introduction of the index sets of active constraints. In [?, ?, ?], the extensive use of complementarity approach enables the design of efficient event-driven algorithms at least for planar systems with contact and Coulomb's friction. All of these references use the standard index-1 formulation of the dynamics with explicit Runge-Kutta solvers. Except in the original work of [?] where the comparison between several ODE solvers is made, and the influence of the precision of the root-finding procedure is studied in an event-driven context, the comparison of ODE/DAE solvers for multibody systems in an event-driven context has not been further studied.

Besides the root-finding procedures, the efficiency of the event-driven scheme relies on the efficiency of the time-stepping schemes within the smooth phases. In the context of bilateral constraints, this question has been deeply discussed in [?, ?, ?, ?] and comprehensive numerical comparisons can be found in [?, Section VII.7] and the test set for IVP solvers suite [?]. Let us summarize the main approaches in the following paragraphs.

When an index-1 formulation is chosen, any ODE solver may virtually be applied to the system. For non stiff systems, the use of explicit high order Runge-Kutta method with projection techniques to control the drift-off effect is a popular choice. Let us cite for instance the DOPRI5 solver [?] with its modification taking into account the projection MDOP5 [?]. We have retained **the latter** solver for our comparisons. We can also cite the use of extrapolation techniques for index-1 formulation together with projection techniques as in [?, ?]. Comparisons of these approaches have been done in [?]. In [?, ?, ?], the authors transform the initial DAE describing the dynamics of a multibody system into a State Space ODE (SSODE) by using the technique of coordinate partitioning. The obtained SSODE, describing the dynamics using only the independent coordinates, is solved using the SDIRK (Singly Diagonally Implicit Runge Kutta) algorithm [?].

An alternative approach, also for non stiff systems, is the use of half-explicit Runge-Kutta schemes for the index-2 **formulations**. The approach tackles the problem at the velocity level and avoids the differentiation of the constraints and their evaluation at the acceleration level. One of the main solvers of this family dedicated to Lagrangian mechanical systems is a Runge-Kutta Half-Explicit Method of order 5 (HEM5) developed in [?]. It enables a drastic reduction of the drift-off effect [?, ?]. In [?], Arnold makes a classification of half-explicit methods, and explains the limitations of some of them to deal with the algebraic constraints, which leads to an order drop, and proposes a new class of methods that handles this issue that yields Partitioned Half-Explicit Method as PHEM56 [?, ?].

For stiff systems, the Backward Differentiation Formulae (BDF) method with index-2 stabilized formulation, based on the Gear-Gupta-Leimkuhler(GGL) method [?], is the most popular method in industrial simulation software for smooth constrained systems. Nevertheless, higher order multi-step methods are not the most relevant methods in the context of an event-driven approach, since the change in the active closed contacts implies a reinitialization of the time-stepping schemes. Concerning the one-step methods, index-2 stabilized formulations based on the Gear-Gupta-Leimkuhler(GGL) method have also been integrated by implicit Runge-Kutta solvers such as RADAU5 [?] and the generalized- $\alpha$  schemes [?, ?, ?, ?]. The RADAU5 was set aside mainly because the targeted applications of stiff systems come from the flexible multi-body

dynamics. We have preferred the generalized- $\alpha$  scheme for its lower memory requirement and its ability to control the numerical damping, which is an important feature for end-users in mechanical engineering. Furthermore, these methods, due to their low order and their ability to treat index-2 formulation, and even index-3 formulations, are good candidates for a time-stepping scheme for stiff systems in an event-driven framework.

## 1.2 Objectives and outline of the paper

In this article, several state-of-the-art numerical time integration methods for each formulation are compared: the generalized- $\alpha$  scheme for the index-3 formulation and the stabilized index-2 formulation, (Partitioned) Runge-Kutta Half-Explicit Method of order 5 (HEM5 and PHEM56) for index-2 DAEs with projection techniques, and Runge-Kutta explicit scheme of order 5, the Dormand-Prince scheme (DOPRI5), for index-1 DAEs with projection techniques (MDOP5).

We compare these schemes in terms of efficiency, violation of the constraints and the way they handle stiff dynamics. Points of comparison include the violation 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 numerous industrial benchmarks that contain several nonsmooth 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 motivation behind this work is twofold:

- Firstly, we want to compare several numerical schemes in terms of their efficiency when they are used to compute the dynamics of complex mechanical systems with a large number of degrees of freedom, joints and contacts.
- Secondly, we want to analyze these schemes in terms of violation of the constraints **at the position, velocity and acceleration levels**.

We would like to draw a particular attention to the latter point. The violation at the position level means that the joint constraints are not satisfied anymore. From a numerical standpoint, the violation of the position and velocity constraints can lead to serious problems when using an event-driven scheme, because of the unavoidable necessary numerical thresholds for the evaluation of the index sets. Indeed, event-driven schemes are sensitive to these thresholds; this point will be made clear later. In addition, using a CAD description for the geometry makes it sometimes difficult for the detection of impact, and therefore for the update of the set of active constraints. At the other end of the spectrum, imposing the constraints at the position and using index-3 formulations may lead to numerical artifacts and oscillations in the Lagrange multipliers [?, ?]. Such artifacts yield wrong forecasts for the release and activation of the unilateral constraints.

This work focuses more on the practical use of numerical integration methods in the field of multibody dynamics, than on their theoretical aspects and convergence properties, by studying several methods on a wide range of multibody mechanisms. Very few papers discuss the performances of numerical methods on industrial benchmarks, among them let us cite the work

of Negrut [?] and Arnold [?]. On industrial examples, theoretical properties are sometimes not sufficient to make a robust solver in practice, particularly when the geometric constraints are evaluated through a CAD library and when the Jacobian matrix of the constraints has not full rank (see discussion in Section 4.5).

The outline of the article is as follows. In Section 2, we present a description of the principles of the event-driven scheme that is used. A brief presentation of the numerical methods to be used for the simulations is discussed in Section 3. In Section 4, we present a deeper comparison on industrial benchmarks. Conclusions end the article in Section 5.

## 2 Strategies of the integration of the dynamics with contact

In this section, we recall the problem, then we introduce the definition of the index sets describing the status of the contact, and finally, we present the general algorithm of the event-driven scheme.

### 2.1 Problem setting

Using the Lagrangian formalism, the dynamics of a mechanical system submitted to  $m$  unilateral and bilateral constraints, with initial conditions at an initial time  $t_0$  given by  $q(t_0) = q_0$  and  $v(t_0) = v_0$ , can be described by the following equations of motion:

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q, v, t) + G^T(q)\lambda \\ g^\alpha(q) = 0, \alpha \in \mathbb{B} \\ 0 \leq g^\alpha(q) \perp \lambda^\alpha \geq 0, \alpha \in \mathbb{U}, \end{cases} \quad (1)$$

where :

- $q \in \mathbb{R}^n$  denotes the vector of generalized coordinates and  $v = \dot{q}$  is the vector of generalized velocities,
- $M(q) \in \mathbb{R}^{n \times n}$  is the symmetric positive definite matrix of inertia,
- $F(q, v, t)$  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,  $\mathbb{B} \subset \mathbb{N}$  denotes the index set of bilateral constraints resulting from the joint conditions, while  $\mathbb{U} \subset \mathbb{N}$  denotes the set of unilateral constraints describing the contact conditions. The set of all constraints is denoted by  $I_0 = \mathbb{B} \cup \mathbb{U}$ .
- $G(q)$  is the Jacobian matrix of the constraints:

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

◦  $\lambda \in \mathbb{R}^m$  is the Lagrange multipliers vector associated with the constraints.

It is worth noting that the Newton-Euler formalism can be formulated equivalently with the introduction of an operator  $T(q)$  of appropriate dimension such that  $\dot{q} = T(q)v$  where  $q$  contains the parameterization of the finite rotation.

The complementarity condition in (1) illustrates the fact that there is a reaction force for an unilateral constraint  $\alpha \in \mathbb{U}$  only when the bodies are in contact,  $g^\alpha(q) > 0 \Rightarrow \lambda^\alpha = 0, \alpha \in \{\mathbb{U}\}$ . Furthermore, since there is no adhesion effect, the reaction force has to be non negative.

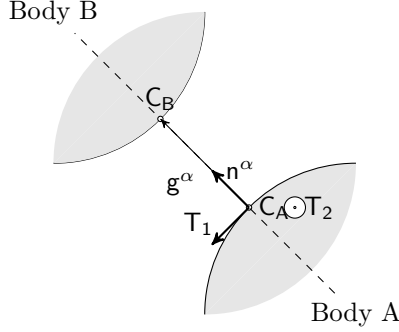


Figure 1: Signed distance between two bodies  $A$  and  $B$  at contact  $\alpha$

Fig.1 illustrates one unilateral constraint  $g^\alpha$  defined as the signed distance between  $C_A$  and  $C_B$ . In this case,  $g^\alpha = (C_B - C_A) \cdot n^\alpha$ , where  $n^\alpha$  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 Moreau impact law. For the closed contacts index set  $I_1(q) = \{\alpha \in I_0 \mid g^\alpha(q) = 0\}$ , we compute the impulse  $p(t_n)$  and the post-impact velocity  $v^+(t_n)$  by solving the Newton impact equations at the time of impact  $t_n$

$$\left\{ \begin{array}{l} M(q(t_n))(v^+(t_n) - v^-(t_n)) = p(t_n) \\ U_N^{\alpha,+}(t_n) = G^\alpha(q(t_n))v^+(t_n) \\ U_N^{\alpha,-}(t_n) = G^\alpha(q(t_n))v^-(t_n) \\ p(t_n) = \sum_{\alpha \in I_1} G^{\alpha T}(q(t_n))P_N^\alpha(t_n) \\ 0 = U_N^{\alpha,+}(t_n) = U_N^{\alpha,-}(t_n), \alpha \in \mathbb{B}, \\ 0 \leq U_N^{\alpha,+}(t_n) + eU_N^{\alpha,-}(t_n) \perp P_N^\alpha(t_n) \geq 0, \alpha \in \mathbb{U}, \end{array} \right. \quad (2)$$

where

$$\left\{ \begin{array}{l} v^+(t_n) = \lim_{\tau \rightarrow t_n, \tau > t_n} v(\tau) \\ v^-(t_n) = \lim_{\tau \rightarrow t_n, \tau < t_n} v(\tau). \end{array} \right. \quad (3)$$

The complementarity condition in (2) describes the Signorini condition written at the velocity level and augmented by the Moreau impact law [?]. In (2),  $U_N^\alpha$  is the vector of normal relative velocities and  $P_N^\alpha$  is the vector of local impulses at the contact points. The scalar  $e \in [0, 1]$  is the global kinematic coefficient of restitution. When the inertia matrix is invertible, problem (2) can be reduced to local unknowns  $U_N^\alpha$  and  $P_N^\alpha$ , computed by solving the following Linear Complementarity Problem (LCP)

$$\begin{cases} U_N^+(t_n) = G(q(t_n))M^{-1}(q(t_n))G^T(q(t_n))P_N(t_n) + U_N^-(t_n) \\ 0 \leq U_N^+(t_n) + eU_N^-(t_n) \perp P_N(t_n) \geq 0. \end{cases} \quad (4)$$

The LCP matrix  $G(q(t_n))M^{-1}(q(t_n))G^T(q(t_n))$  is called the Delassus matrix. Equation (4) describes the so-called *contact-impact LCP*.

## 2.2 Definition of the index of a DAE

In the literature [?, ?], 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 in terms of the independent variables  $q(t), v(t)$  and  $\lambda(t)$ . Let us define the index set of closed contacts with a vanishing relative velocity

$$I_2(t) = \mathbb{B} \cup \{\alpha \in \mathbb{U} \mid g^\alpha(q(t)) = 0, G^\alpha(q(t))v(t) = 0, \lambda^\alpha(t) \geq 0\} \quad (5)$$

For a constrained multibody system, when the contacts are closed for a non trivial period of time, that is for the contact  $\alpha \in I_2(t)$  for all  $t \in [t_a, t_b], t_b > t_a$ , the dynamics of the system is described with the following index-3 DAE

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = F(q, v, t) + G^T(q)\lambda \\ g^\alpha(q) = 0, \alpha \in I_2 \\ g^\alpha(q_0) = 0, \alpha \in I_2 \\ \lambda^\alpha = 0, \alpha \in I_0 \setminus I_2, \end{cases} \quad (6)$$

For the sake of readability, we omit the time argument  $t$  in the index sets  $I_0, I_2$  as we have already done for the variables that depend on time. It is well known that index-3 differential algebraic equations are difficult to handle numerically [?]. Therefore, the dynamics is usually solved as 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

$$\left\{ \begin{array}{l} \dot{q} = v \\ M(q)\dot{v} = F(q, v, t) + G^T(q)\lambda \\ G^\alpha(q)v = 0, \alpha \in I_2 \\ g^\alpha(q_0) = 0, \alpha \in I_2 \\ G^\alpha(q_0)v_0 = 0, \alpha \in I_2 \\ \lambda^\alpha = 0, \alpha \in I_0 \setminus I_2, \end{array} \right. \quad (7)$$

where  $v_0$  is the initial velocity.

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

$$\left\{ \begin{array}{l} \dot{q} = v \\ M(q)\dot{v} = F(q, v, t) + G^T(q)\lambda \\ G^\alpha(q)\dot{v} + \frac{dG^\alpha(q)}{dt}v = 0, \alpha \in I_2 \\ g^\alpha(q_0) = 0, \alpha \in I_2 \\ G^\alpha(q_0)v_0 = 0, \alpha \in I_2 \\ G^\alpha(q_0)\dot{v}_0 + \frac{dG^\alpha}{dt}(q_0)v_0 = 0, \alpha \in I_2 \\ \lambda^\alpha = 0, \alpha \in I_0 \setminus I_2. \end{array} \right. \quad (8)$$

The system (8) can be written using matrices as

$$\begin{pmatrix} M(q) & -G^{\alpha T}(q) \\ G^\alpha(q) & 0 \end{pmatrix} \begin{pmatrix} \dot{v} \\ \lambda^\alpha \end{pmatrix} = \begin{pmatrix} F(q, v, t) \\ -\frac{dG^\alpha(q)}{dt}v \end{pmatrix}, \quad (9)$$

where  $\alpha \in I_2$ . When the Delassus operator  $G^\alpha(q)M^{-1}(q)G^{\alpha T}(q)$  is invertible, the DAE (6) is of index 3, and the index-1 DAE (8) can be rewritten as an ODE of the form  $\dot{y} = f(y, t)$ , or more explicitly

$$\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} v \\ -M^{-1}(q)G^{\alpha T}(q)(G^\alpha(q)M^{-1}(q)G^{\alpha T}(q))^{-1}(\frac{dG^\alpha}{dt}v + G^\alpha(q)M^{-1}(q)F(q, v, t)) + M^{-1}(q)F(q, v, t) \end{pmatrix}. \quad (10)$$

It can be checked that the dynamics in (10) renders the manifold  $\{(q, v) \mid g^\alpha(q) = 0, G^\alpha(q)v = 0, \alpha \in I_2\}$  invariant. Under the Lipschitz continuity of the **right-hand** side in (10), a unique solution  $(q, v)$  of (10) 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 (10) satisfies the initial problem in (6).

In the case of unilateral constraints, a *relative degree* can be defined as in [?] 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 **level**. Indeed, the complementarity relation  $0 \leq g(q) \perp \lambda \geq 0$  can be formulated at the velocity level as

$$\begin{aligned} &\text{if } g(q) = 0, \text{ then } 0 \leq G(q)v \perp \lambda \geq 0, \\ &\text{else, } \lambda = 0. \end{aligned} \tag{11}$$

or at the acceleration level as

$$\begin{aligned} &\text{if } g(q) = 0, \ G(q)v = 0, \ \text{then } 0 \leq \frac{d^2g(q)}{dt^2} \perp \lambda \geq 0. \\ &\text{else, } \lambda = 0. \end{aligned} \tag{12}$$

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

### 2.3 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:

$$I_1(t) = \mathbb{B} \cup \{\alpha \in \mathbb{U} \mid g^\alpha(q(t)) = 0\}, \tag{13}$$

- the index set  $I_2$  of contacts that are active in position and velocity, or **closed contacts**, defined in (5).

The occurrence of an event leads to a change in the index sets  $I_1$  and  $I_2$ . 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 continue the time integration up to the next event. The event-driven scheme used in this paper is illustrated in Fig.2. For more details on event-driven schemes, we refer to [?, ?, ?].

Some steps in the algorithm in Fig.2 deserve further consideration:

- *Update the index sets.* Some numerical thresholds are required to evaluate equalities and inequalities in the definition of the index sets  $I_1$  and  $I_2$  in (13) and (5). For that purpose, we write in the numerical practice  $g^\alpha(q) \leq \tau_0$ ,  $G^\alpha(q)v \leq \tau_1$  and  $\lambda^\alpha \geq \tau_3$  for some positive scalar values  $\tau_i, i = 1, 2, 3$ . Besides the problems induced by the floating point arithmetic, such thresholds are mandatory to stabilize index sets when some approximation errors are made by the numerical integrators. This makes the event-strategy particularly sensitive to the violation of the constraints.
- *Correct the violated constraints.* For a closed constraint over a nontrivial period of time, that is  $\alpha \in I_2(t)$ , some violations may appear due to some approximation errors. The violated constraints have to be brought back to the admissible manifold. As an example, the strategy of Lubich [?, Chapter 7] may be used. Similar projection techniques can be found in [?, ?, ?]. The same procedure is also applied after an impact and a change of  $I_1$ .

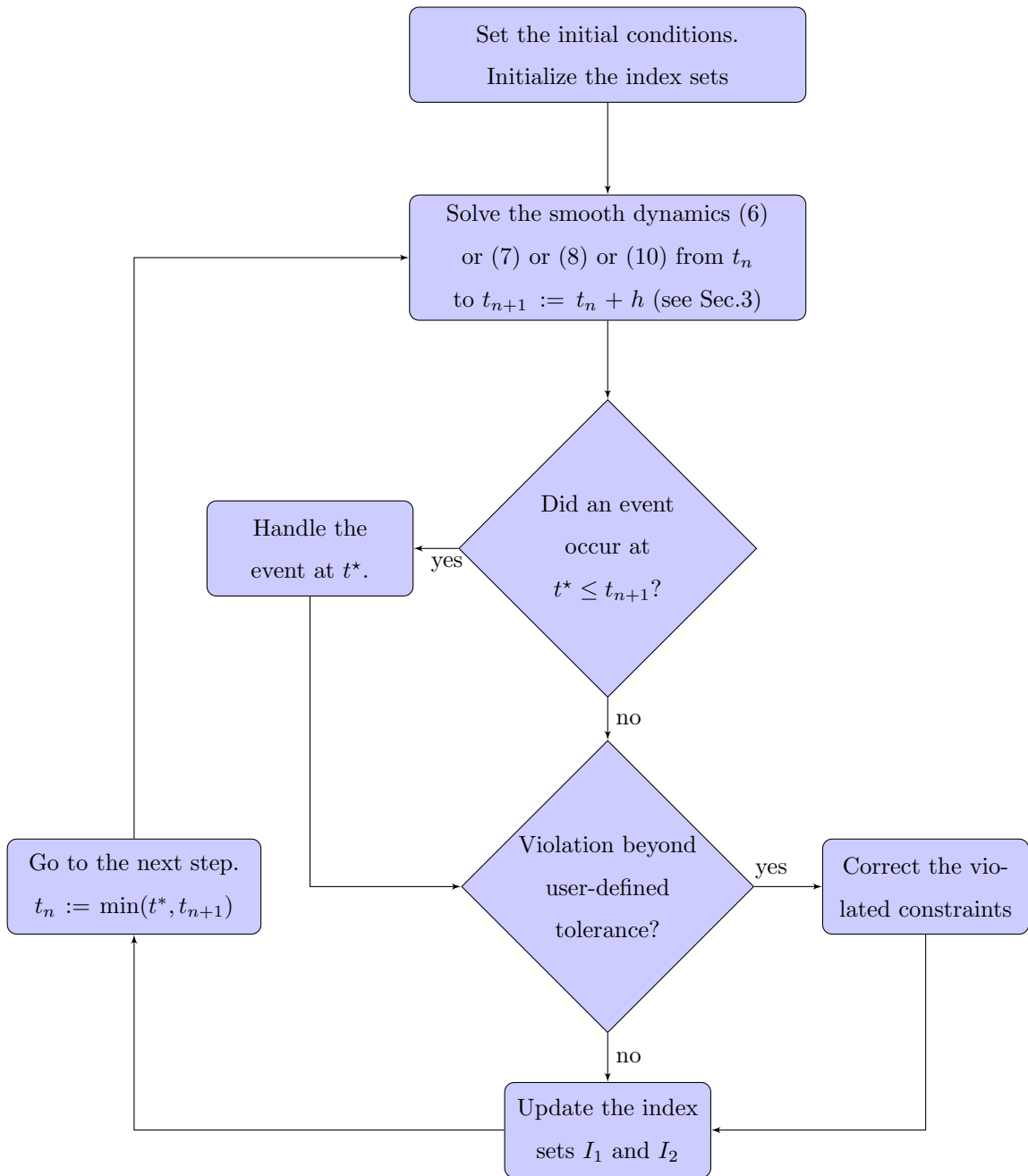


Figure 2: The event-driven scheme. Synopsis of a one-step integration.

- *Handle the event.* If many events are detected during the time step, then their times of occurrence are compared and the first event to happen is the first to be handled. There are many types of events, including:

1. change in the loading conditions.
2. change in the sign of a Lagrange multiplier for a unilateral constraint, indicating a take-off event.
3. impacts (activation of constraints), in this case we solve the impact LCP (4) at  $t^*$ .
4. change in the normal vector that leads to a jump in the constraints and contact forces.

### 3 The time integration schemes

Three numerical methods have been selected for this study. Namely, generalized- $\alpha$  scheme, HEM5 and DOPRI5.

#### 3.1 An index-3 and index-2 DAE integrator: the generalized- $\alpha$ scheme

The Newmark family of schemes [?, ?], widely used in the field of linear structural dynamics, has also been applied to the simulations of multibody systems. This second-order accuracy scheme is unconditionally stable for a linear dynamics, for a correct choice of parameters. However, the use of dissipation leads to a drop in its order. In [?, ?, ?], Chung and Hulbert, and then other authors **proposed** the generalized- $\alpha$  methods, a family of schemes that generalizes all the HHT methods and whose main characteristic is to properly deal with the high frequencies, while keeping the  $2^{nd}$  order of the method. Cardona and Géradin [?] adapted the method to compute the dynamics of multibody systems with joints. Contributions in the development and application of these schemes in multibody dynamics are found in [?] for an analysis of convergence of the index-3 version, and in [?, ?, ?, ?, ?, ?] for convergence analysis and stabilization techniques. Since there exist slightly different versions in the literature, let us recall there the variant that we use in the comparison. The dynamics (6) is written at time  $t_{n+1}$  as

$$\begin{cases} \dot{q}_{n+1} = v_{n+1} \\ M(q_{n+1})\ddot{q}_{n+1} = F(q_{n+1}, v_{n+1}, t_{n+1}) + G^T(q_{n+1})\lambda_{n+1} \\ g(q_{n+1}) = 0. \end{cases} \quad (14)$$

Let us introduce the acceleration-like variable  $a_n$  defined by the recurrence relation

$$(1 - \alpha_m)a_{n+1} + \alpha_m a_n = (1 - \alpha_f)\ddot{q}_{n+1} + \alpha_f \ddot{q}_n. \quad (15)$$

At the beginning of the simulation, this variable is initialized as  $a_0 = \ddot{q}_0$ . The following difference equations relate  $q_{n+1}$ ,  $v_{n+1}$  and  $a_n$ ,  $a_{n+1}$ :

$$\begin{cases} q_{n+1} = q_n + hv_n + h^2(\frac{1}{2} - \beta)a_n + h^2\beta a_{n+1} \\ v_{n+1} = v_n + h(1 - \gamma)a_n + h\gamma a_{n+1}, \end{cases} \quad (16)$$

where the constants  $\alpha_f$ ,  $\alpha_m$ ,  $\beta$  and  $\gamma$  are suitably chosen so that the scheme is stable for the linear dynamics. The numerical damping is made easier by introducing the so-called *spectral radius parameter*  $\rho_\infty$ . Indeed,  $\rho_\infty = 0$  corresponds to asymptotic annihilation of the high frequencies, while  $\rho_\infty = 1$  corresponds to no numerical damping. The algorithm is unconditionally stable [?] if the coefficients are chosen such that  $\alpha_m = (2\rho_\infty - 1)/(\rho_\infty + 1) < 1/2$ ,  $\alpha_f = \rho_\infty/(\rho_\infty + 1) < 1/2$ ,  $\gamma = 1/2 + \alpha_f - \alpha_m > 1/2$ ,  $\beta = 1/4(\gamma + 1/2)^2$  for  $\rho_\infty < 1$ . Since the generalized- $\alpha$  scheme is inherently a linear multi-step method, this choice is only valid for a fixed time-step and yields an order drop otherwise. For a variable time-step, we refer to [?, ?] for an improved rule.

The scheme is based on prediction and correction steps where some Newton iterations are performed in order to solve the dynamics and the constraint residuals  $R_d$  and  $R_{c,q}$  defined by

$$\begin{cases} R_d(q_{n+1}, v_{n+1}, t_{n+1}, \lambda_{n+1}) = M(q_{n+1})\ddot{q}_{n+1} - F(q_{n+1}, v_{n+1}, t_{n+1}) - G^T(q_{n+1})\lambda_{n+1} \\ R_{c,q}(q_{n+1}) = g(q_{n+1}), \end{cases} \quad (17)$$

where  $\ddot{q}_{n+1}$  and  $v_{n+1}$  are replaced by  $q_{n+1}$  thanks to (15) and (16). The Newton iterations amount to solving the following linear system at the iteration  $k$

$$S_q(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}) \begin{pmatrix} \Delta q \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} R_d(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}^k) \\ R_{c,q}(q_{n+1}^k) \end{pmatrix}, \quad (18)$$

with the iteration matrix

$$S_q(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}) = \begin{pmatrix} \beta' M(q_{n+1}^k) + \gamma' C_t(q_{n+1}^k, v_{n+1}^k, t_{n+1}) + K_t(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}^k) & -G^T(q_{n+1}^k) \\ G(q_{n+1}^k) & 0 \end{pmatrix} \quad (19)$$

and where  $\beta' = \frac{1-\alpha_m}{h^2\beta(1-\alpha_f)}$ ,  $\gamma' = \frac{\gamma}{h\beta}$ ,  $K_t = \frac{\partial(M\ddot{q}-F+G^T\lambda)}{\partial q}$  is the stiffness matrix and  $C_t = -\frac{\partial F}{\partial \dot{q}}$  is the damping matrix. Note that this algorithm aims at maintaining the constraints at the position level, but it can also be reformulated to write the constraints on the velocity level or on the acceleration level [?]. When considering the constraints at the velocity level, the problem is formulated as an index-2 DAE whose discretization is given with the first equation of (14) together with

$$R_{c,v}(q_{n+1}, v_{n+1}) = G(q_{n+1})v_{n+1} = 0. \quad (20)$$

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

$$S_v(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}) \begin{pmatrix} \Delta v \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} R_d(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}^k) \\ G(q_{n+1}^k)v_{n+1}^k \end{pmatrix}, \quad (21)$$

with the iteration matrix

$$S_v(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}) = \begin{pmatrix} \beta' M(q_{n+1}^k) + C_t(q_{n+1}^k, v_{n+1}^k, t_{n+1}) + \gamma' K_t(q_{n+1}^k, v_{n+1}^k, t_{n+1}, \lambda_{n+1}^k) & -G^T(q_{n+1}^k) \\ G(q_{n+1}^k) & 0 \end{pmatrix} \quad (22)$$

where  $\beta' = \frac{1-\alpha_m}{h\gamma(1-\alpha_f)}$  and  $\gamma' = \frac{h\beta}{\gamma}$ .

The generalized- $\alpha$  method can be adapted to enforce both position and velocity constraints, based on the stabilization technique also called the Gear–Gupta–Leimkuhler(GGL) method [?] (see for instance [?, ?, ?]). Since there is no consensus on the generalized- $\alpha$  scheme for index-2 stabilized formulation, we recall here the scheme that has been used in the comparison. This has also been done in the context of the simulation of the dynamics of multibody systems with both unilateral and bilateral constraints in [?]. Let us assume a system only subject to bilateral constraints. An additional vector of Lagrange multipliers  $\mu$  associated with the position constraints is introduced. The augmented equations of dynamics read:

$$\begin{cases} M(q)\dot{q} = M(q)v + G^T(q)\mu \\ M(q)\ddot{q} = F(q, v, t) + G^T(q)\lambda \\ g(q) = 0, \mu \\ G(q)v = 0, \lambda \end{cases} \quad (23)$$

Let us denote:  $\tilde{q} = \int_{[t_n, t_{n+1}]} v dt$ ,  $U = q - \tilde{q}$ ,  $\dot{\tilde{q}} = \tilde{v}$ , and  $\nu = \int_{[t_n, t_{n+1}]} \mu dt$ . In [?], the time discretization in an event-capturing context is discussed. Following their conclusions on the efficiency of the algorithm, the method used in that paper is based on the decoupling of the resolution at the velocity level given by the system (21) and the resolution at the position level. The final algorithm is based on two steps:

- a correction step consisting in the resolution of (21).
- enforcement of the position constraints by solving:

$$\begin{pmatrix} M(q_{n+1}^k) & G^T(q_{n+1}^k) \\ G(q_{n+1}^k) & 0 \end{pmatrix} \begin{pmatrix} \Delta q \\ \Delta \nu \end{pmatrix} = - \begin{pmatrix} M(q_{n+1}^k)U_{n+1}^k - G^T(q_{n+1}^k)\nu_{n+1}^k \\ g(q_{n+1}^k) \end{pmatrix}, \quad (24)$$

Algorithm 1 presents a sketch of the method.

Other options are discussed in [?]. Especially, the question of the coupling strategies between the position level and the velocity level. A very interesting analysis is also done in [?] in a similar but different context. The authors consider an index-3 DAE formulation and solve a constrained mechanical system with a generalized- $\alpha$  Lie group scheme. Since the constraints are enforced at the position level, the hidden constraints at the velocity level are not satisfied by the time-stepping scheme. They consider two alternative solution procedures: one is based on the stabilization of the constraints at the velocity level as it is developed in [?], and the other one, a simple projection of the manifold defined by the velocity constraints. It is shown that the latter method induces instabilities and order reduction. In the scheme described in Algorithm 1, we attempt to avoid this drawback by iterating on both levels up to convergence.

To estimate the integration error, we follow the methodology proposed by G eradin and Cardona in [?] and Negrut and al. in [?] for the HHT schemes family. The exact value of the positions vector can be approximated by a Taylor series

---

**Algorithm 1** The generalized- $\alpha$  method with stabilization of the position and the velocity constraints

---

**Require:**  $q_n, v_n, t_n$

**Ensure:**  $q_{n+1}, v_{n+1}, \lambda_{n+1}$

$$a_{n+1} \leftarrow \frac{1}{1-\alpha_m}(\alpha_f \ddot{q}_n - \alpha_m a_n)$$

$$v_{n+1} \leftarrow v_n + h(1-\gamma)a_n + h\gamma a_{n+1}$$

$$q_{n+1} \leftarrow q_n + hv_n + h^2(\frac{1}{2} - \beta)a_n + \beta h^2 a_{n+1}$$

$$\lambda_{n+1} \leftarrow 0, \nu_{n+1} \leftarrow 0, U_{n+1} \leftarrow 0$$

**repeat**

$$\Delta v, \Delta \lambda \leftarrow \text{solution of (21)}$$

$$v_{n+1} \leftarrow v_{n+1} + \Delta v$$

$$q_{n+1} \leftarrow q_{n+1} + \gamma' \Delta v$$

$$\lambda_{n+1} \leftarrow \lambda_{n+1} + \Delta \lambda$$

$$\Delta q, \Delta \nu \leftarrow \text{solution of (24)}$$

$$U_{n+1} \leftarrow U_{n+1} + \Delta q$$

$$q_{n+1} \leftarrow q_{n+1} + \Delta q$$

$$\nu_{n+1} \leftarrow \nu_{n+1} + \Delta \nu$$

**until**  $R_d(q_{n+1}, v_{n+1}, t_{n+1}, \lambda_{n+1}) \leq \text{tol}$  and  $G(q_{n+1})v_{n+1} \leq \text{tol}$  and  $g(q_{n+1}) \leq \text{tol}$  (tol is a user defined tolerance)

---

development around  $t_n$ :

$$q_e(t_n + h) = q_n + h\dot{q}_n + \frac{h^2}{2}\ddot{q}_n + \frac{h^3}{6}\dddot{q}_n + \mathcal{O}(h^4). \quad (25)$$

The integration error is computed as

$$e = q_{n+1} - q_e(t_n + h). \quad (26)$$

By substituting the expression of  $q_{n+1}$  from (16) into (26), we get:

$$e = h^2\left(\frac{1}{2} - \beta\right)a_n + h^2\beta a_{n+1} - \frac{h^2}{2}\ddot{q}_n - \frac{h^3}{6}\dddot{q}_n + \mathcal{O}(h^4). \quad (27)$$

The third derivative  $\dddot{q}_n$  of the position can be approximated by

$$\dddot{q}_n = \frac{\ddot{q}_{n+1} - \ddot{q}_n}{h} + \mathcal{O}(h). \quad (28)$$

By substituting this expression of  $\dddot{q}_n$  into (27) and using the relations in (16), we obtain

$$e = q_{n+1} - q_n - \frac{1}{h}\dot{q}_n - \frac{h^2}{3}\ddot{q}_n - \frac{h^2}{6}\ddot{q}_{n+1} + \mathcal{O}(h^4). \quad (29)$$

Finally, the optimal step size is computed with

$$h_{opt} = s h \left( \frac{tol}{\|e\|} \right)^{\left(\frac{1}{p}\right)}, \quad (30)$$

where  $s$  is a safety factor,  $tol$  is the user required tolerance and  $p$  corresponds to the order of consistency of the scheme. For the generalized- $\alpha$  method,  $p = 2$ .

### 3.2 Index-2 DAE integration scheme: HEM5 and PHEM56 with a projection technique.

HEM5 and PHEM56 belong to the family of half-explicit RK methods which have been deeply studied in [?]. Half-explicit methods use linear systems involving jacobian matrices that are evaluated at two different time instants  $t_n + c_i h$  and  $t_n + c_{i+1} h$  as in (31). These systems are to be solved at each stage to evaluate the estimations of the acceleration  $\dot{V}_i$  and the multiplier  $\Lambda_i$ ,

$$\begin{pmatrix} M(Q_i) & -G^T(Q_i, t_n + c_i h) \\ G(Q_{i+1}, t_n + c_{i+1} h) & 0 \end{pmatrix} \begin{pmatrix} \dot{V}_i \\ \Lambda_i \end{pmatrix} = \begin{pmatrix} F(Q_i, V_i, t_n + c_i h) \\ r_i \end{pmatrix}, \quad (31)$$

where

$$r_i = -\frac{G(Q_{i+1})}{ha_{i+1,i}}(v_n + h \sum_{j=1}^{i-1} a_{i+1,j} \dot{V}_j), \quad (32)$$

and  $c_i$  and  $a_{ij}$  are the coefficients of the method. It is worth noting that the matrix in (31) is not symmetric and that the solution of the system exists only under several considerations that are discussed in [?]. In addition, this non-symmetry may lead to numerical issues when  $G(Q_i)$  and  $G(Q_{i+1})$  do not have the same rank. This may happen on models with contact constraints.



The HEM5 solver is based on a half-explicit order 5 Runge-Kutta(RK) method with 8 stages, described in [?]. It is suited to the integration of DAEs, reduced to the index-2 formulation. V. Brasey and E. Hairer [?, ?, ?] defined an estimation of the integration error based on the 7<sup>th</sup> and 8<sup>th</sup> estimations  $Q_7$  and  $Q_8$  such that

$$\begin{aligned} e_1 &= \|q_{n+1} - Q_8\|_s = O(h^4) \\ e_2 &= \|q_{n+1} - q_n - h(\frac{5}{2}Q_7 - \frac{3}{2}Q_8)\|_s = O(h^3). \end{aligned} \tag{33}$$

Finally, the practical error estimation is computed by

$$e = \frac{e_1^2}{e_1 + 0.01 e_2} = O(h^5). \tag{34}$$

The PHEM56 is a Partitioned Half-Explicit Method as it is described in [?, ?].

Since these solvers **are suitable for** the index-2 formulation, violations of the constraints at the position level may be expected. A standard projection technique onto the admissible manifold can be used by solving the following non-linear problem for all  $\alpha \in I_2$ :

$$\begin{cases} N(q)(\tilde{q} - q) + G^T(q)\mu = 0 \\ g(\tilde{q}) = 0 \end{cases} \tag{35}$$

by means of Newton methods, where  $q$  is the position obtained at the end of the step by the numerical method. Usual choices are the identity matrix or the mass matrix  $M(q)$ .

### 3.3 An ODE integrator: DOPRI5 and its modified version with a projection technique MDOP5

One of the most popular Explicit Runge Kutta (ERK) schemes is the DOPRI5 scheme. It is suited for the resolution of ODEs and is based on six estimations of the derivatives. This method provides two approximations of the solution to evaluate the error. The 5<sup>th</sup> order approximation  $q^{(5)}$  corresponds to the solution at the end of the step while the 4<sup>th</sup> order estimation  $q^{(4)}$  is used for the computation of the error. The computational error is defined with  $e = \|q_{n+1}^{(5)} - q_{n+1}^{(4)}\|$  and the optimal time step is given by (30) with  $p = 4$ . Since the solver uses the index-1 formulation, violations of the constraints at the position and at the velocity level may be expected. After solving the problem (35) for correcting the position, we solve the following problem with the updated position  $\tilde{q}$  for all  $\alpha \in I_2$ :

$$\begin{cases} N(\tilde{q})(\tilde{v} - v) + G^T(\tilde{q})\eta = 0 \\ G(\tilde{q})\tilde{v} = 0 \end{cases} \tag{36}$$

where  $v$  the velocity after the application of the numerical scheme. This correction leads to a scheme similar to the MDOP5 solver [?].

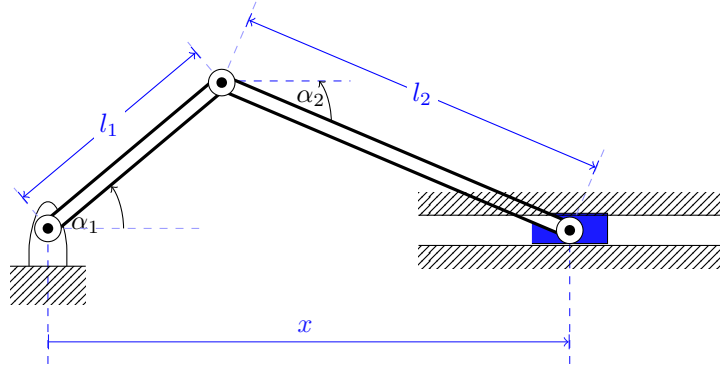


Figure 3: The slider-crank system

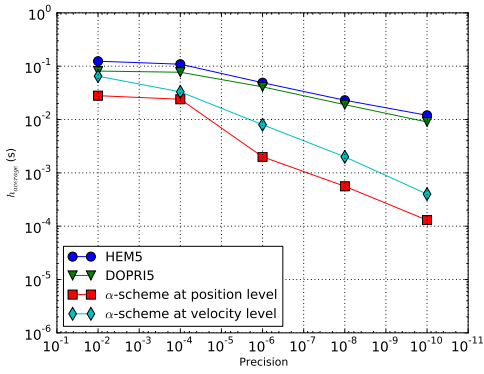
### 3.4 Comparison of the numerical schemes on one academic example

Before using the aforementioned methods on the industrial benchmarks from the Ansys library, we ran simulations on several academic examples that can be found in the [Ph.D thesis](#) [?], with constraints and flexible bodies. For these simulations, we used the HEM5, DOPRI5 and generalized- $\alpha$  scheme in its index-3 and index-2 DAE forms without stabilization. Let us analyze the behavior of these methods on one of these academic tests. The example consists of a slider-crank mechanism submitted to one bilateral constraint  $g_1(\alpha_1, \alpha_2) = l_1 \sin(\alpha_1) + l_2 \sin(\alpha_2)$ , depicted in Fig. 3.

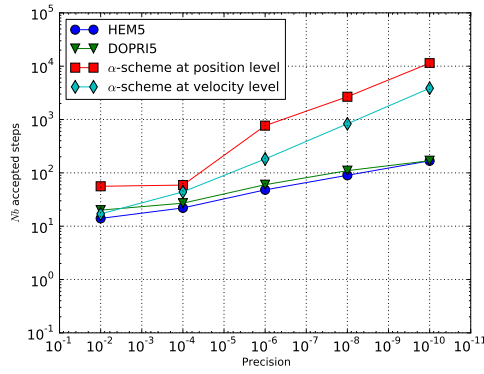
In Fig. 4, we present the average time step size and the maximum value of the violation of the constraints at both position and velocity levels. With the HEM5 solver, the violation of the constraints at the velocity level is close to the machine accuracy (Fig.4e). The violation at the position level is at very acceptable levels for all the values of tolerances on the integration error (Fig.4d). With the DOPRI5 method, the tolerance at the integration error is met with large time step sizes as shown in Fig.4a but the violation of the constraints is much higher than with HEM5. For the slider crank mechanism, the violation of  $g_1$  (Fig.4d) using the generalized- $\alpha$  scheme with position constraints is close to the machine accuracy. However, the accuracy at the velocity level is quite poor. Because of this violation, the detection of the closed contacts defined by the set  $I_2$  in (5) is affected if the violation is greater than the tolerance used to define  $I_2$ . When using the velocity-level generalized- $\alpha$  scheme, the violation is large for tolerances larger than  $10^{-6}$  (Fig.4d), and we can notice in Fig.5 that the numerical effort is smaller than for the position-level generalized- $\alpha$  scheme. From the computational effort point of view, HEM5 is less time-consuming than DOPRI5, and much less time-consuming than the generalized- $\alpha$  scheme with the position constraints and also cheaper than generalized- $\alpha$  with the velocity constraints.

Overall, the salient conclusions of the comparisons on the academic examples are [?]:

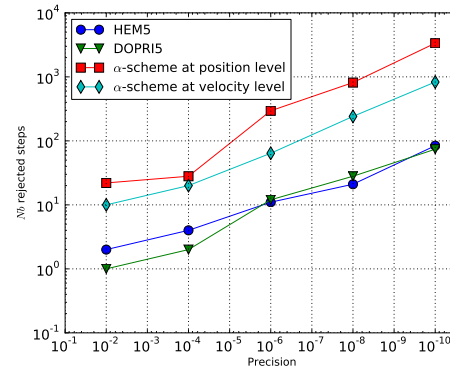
- For rigid systems with bilateral constraints, HEM5 is the most efficient to save computational effort while enforcing the velocity constraints at the machine accuracy and maintaining the violation at very low values for a wide range of [precisions](#) on the integration error. Given its high consistency order, the DOPRI5 scheme gives good results concerning



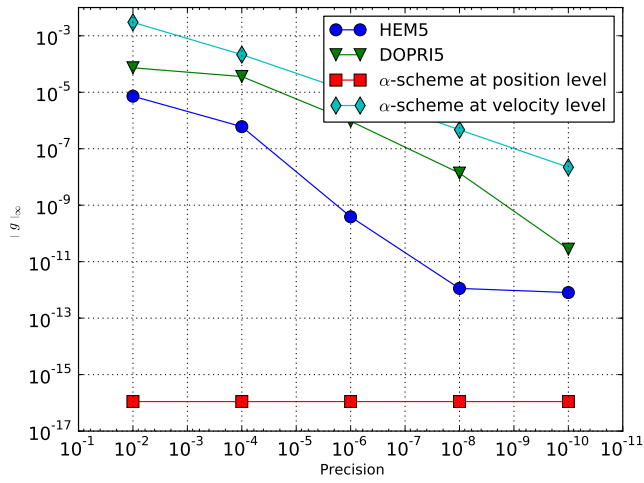
(a) Average time step



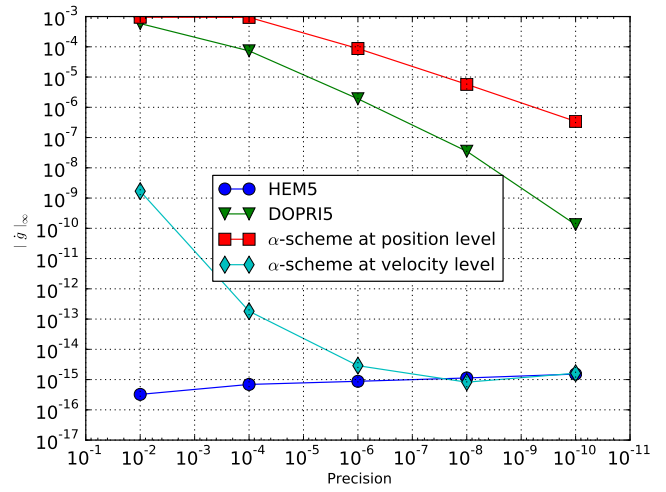
(b) # of accepted steps



(c) # of rejected steps



(d) maximum of violation of the position constraint



(e) maximum of violation of the velocity constraint

Figure 4: Slider crank: simulation characteristics.

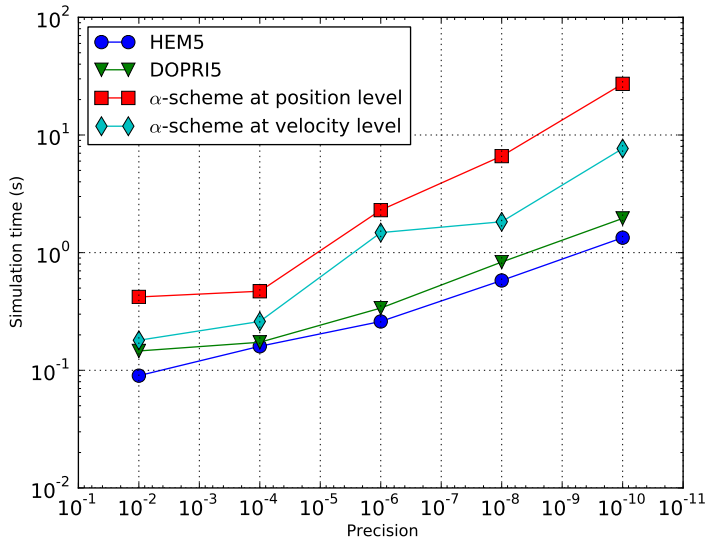


Figure 5: Work/Precision diagrams for the slider-crank mechanism.

the computation time and the enforcement of the constraints. Being initially designed for simulating the structural dynamics, the generalized- $\alpha$  scheme is less computationally efficient for this kind of systems. However, it is important to stress that when using the generalized- $\alpha$  method with velocity constraints instead of the position constraints, the numerical effort is drastically reduced.

- From the simulations using HEM5 and the generalized- $\alpha$  with velocity constraints, it follows that a better stabilization of the constraints is obtained with a discretization of the constraints at the velocity level. Indeed, these schemes enforce the velocity constraints at the machine accuracy while maintaining the violation at very low levels. HEM5 obviously outperforms the generalized- $\alpha$  scheme with velocity constraints due to its higher order (equal to 5).
- **The control of the violation at the acceleration level is also important to maintain at low level.** Some index-3 solvers show some instabilities [?, ?] in the evaluation of the multiplier that can be problematic for the stabilization of  $I_2$ . The DOPRI5 method, which exactly solves the acceleration constraints, outperforms HEM5 and position-level or velocity-level generalized- $\alpha$  methods in maintaining the acceleration constraints. Nevertheless, and as discussed in the literature [?], enforcing the velocity constraints remains the best option since it enables also to both control the violation of the position and the acceleration constraints.

## 4 Comparison of the numerical methods on industrial benchmarks

### 4.1 The framework of the study

The simulations on the industrial applications are performed using the Ansys software. The Newton-Euler formalism is used to describe the dynamics, with a nonsmooth formulation of the contact. The dynamics is then simulated using the event-driven scheme described in Section 2.3. For each test, we ran simulations using HEM5, MDOP5 and the index-2 stabilized generalized- $\alpha$  scheme. According to the results on the academic tests (see [?]), the index-3 generalized- $\alpha$  scheme has not been retained for the industrial tests. Concerning PHEM56, it is more efficient and accurate than HEM5. Some comparisons on academical mechanisms are reported in [?] that confirmed this expectation. As mentioned in Section 3.2 and detailed in Section 4.5, the half-explicit schemes for the index-2 formulation yield to asymmetric matrices that can be difficult to handle if the rank of the constraints changes during a step. We start by a first implementation of HEM5 in the industrial code. As discussed below, the HEM5 solver failed on several tests due to the rank variation. We decided to remove these problems from the test sets and to abandon the preliminary implementation of PHEM56 in the industrial code.

As we said before, the enforcement of the constraints is crucial for a simulation to be successful. If the acceleration constraints are violated, then the accelerations are not correct and neither are the computed forces. If the position or the velocity constraints are violated, this may lead to some non physical interpenetration which makes the geometry engine unable to solve the contact problem. Moreover, the precision of the enforcement of the constraints is directly related to that of the index sets  $I_1$  and  $I_2$ , which represent one of the **keystones** of an event-driven solver. Therefore, a tight and practical relative tolerance of  $10^{-4}$  is defined in the geometry engine which corresponds to the maximal allowed violation of the position constraints. If the position constraints violation exceeds  $10^{-4}$ , then it is corrected by solving the projection systems (35) and (36). Both projections are done for the MDOP5 solver and the projection for the **positions** for the HEM5 solver. The index-2 stabilized generalized- $\alpha$  scheme does not need these projection **steps** since it satisfies the constraints at every time-step. Also, when an impact has been detected, then if the corresponding interpolated position leads to an interpenetration configuration, the violation is corrected in a similar manner with the help of (35), and the impact problem is solved as in (4). Table 1 presents the parameters of the time-step control strategy.

Table 1: Parameters for time step control

Relative integration tolerance (tol)	$10^{-4}$
Minimum time step	$10^{-7}$
Tolerance on Newton's residual	$10^{-8}$
Maximum violation of $g$ and $G(q)v$	$10^{-4}$
Safety factor ( $s$ )	0.3

Table 2: Characteristics of the sets

Set	Max # of DOF	Min # of DOF	Max # of joints	Min # of joints	Max # of contacts	# of problems
1	19	1	38	2	-	69
2	11	2	15	2	7	32
3	31	5	15	4	1	14

## 4.2 Description of the mechanisms

The mechanisms studied in this section are provided from the [Ansys Rigid Body Dynamics, 16.0 test library](http://www.ansys.com/Products/Simulation+Technology/Structural+Analysis/ANSYS+Rigid+Body+Dynamics) (<http://www.ansys.com/Products/Simulation+Technology/Structural+Analysis/ANSYS+Rigid+Body+Dynamics>). They are split into three sets described in Table 2. The first test set comprises simple problems with a number of degrees of freedom between 1 and 19 and number of joints between 2 and 38. This set enables us to test the kinematics and the dynamics with a wide range of joints, without unilateral contacts and impact. The second set comprises mechanisms with impacts. It will help to validate the strategy of solving the shock at the impact and resuming the computation of the dynamics. The third set contains systems where flexible beam models with floating reference frame are used, as described in [?]; the dynamics of these bodies is simulated under different types of loads, and in presence of impacts in some cases. This set contains systems with a number of degrees of freedom that varies between 5 and 31 and a number of kinematic joints comprised between 4 and 14. Illustrations of some mechanisms from each set are depicted in Figures 6, 7, and 8.

## 4.3 Performance profile

The following metric is introduced in [?]. 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 criterion:

$$t_{p,s} = \text{computing time required for solver } s \text{ to solve problem } p. \quad (37)$$

We define the performance ratio as

$$r_{p,s} = \frac{t_{p,s}}{\min \{t_{p,s}, s \in S\}}. \quad (38)$$

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  $\tau \in \mathbb{R}$  :

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P, r_{p,s} \leq \tau\} \leq 1. \quad (39)$$

Thus,  $\rho_s$  is the distribution function for the performance ratio. The number  $\rho_s(\tau)$  represents the probability that the solver  $s$  has a performance ratio no larger than a factor  $\tau$  of the best solver. 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  $\rho_s$  is, the better

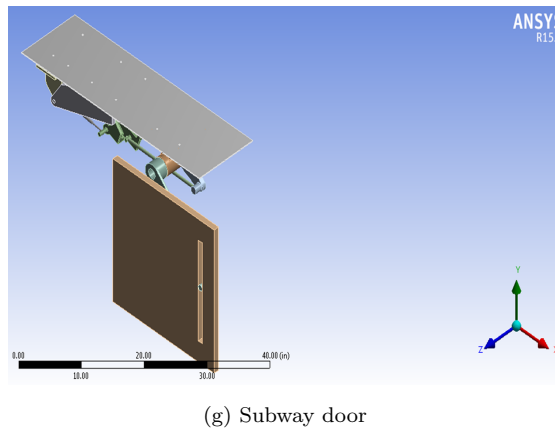
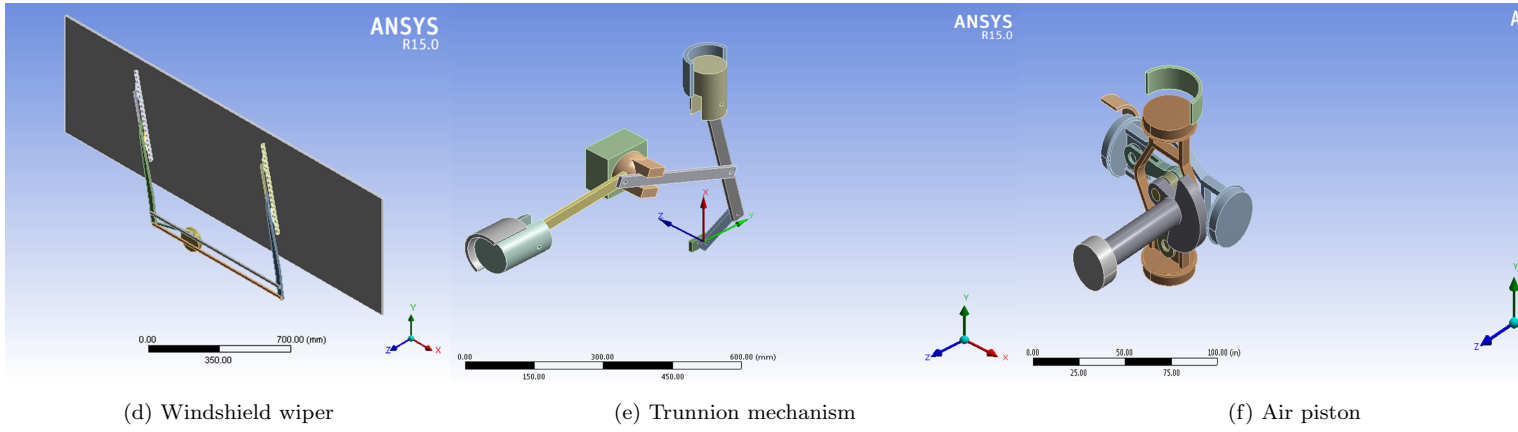
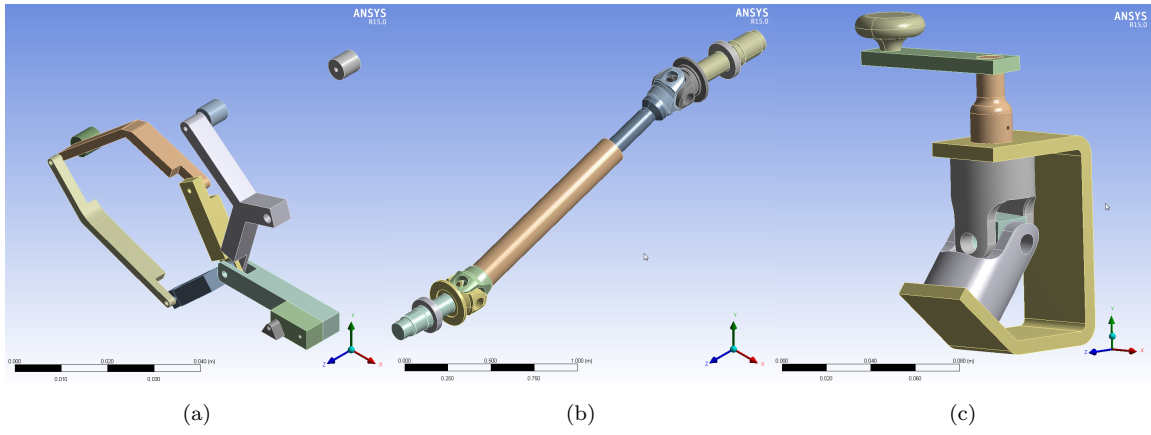
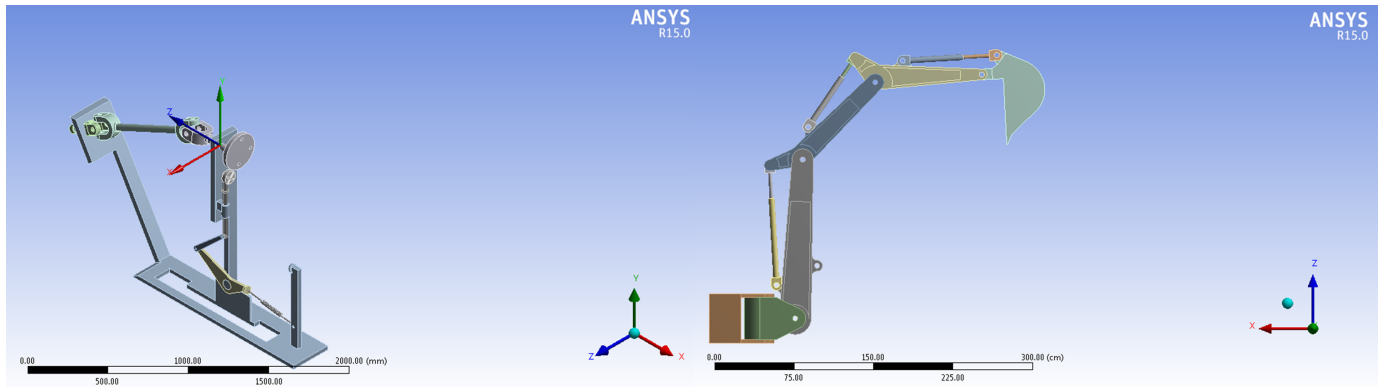
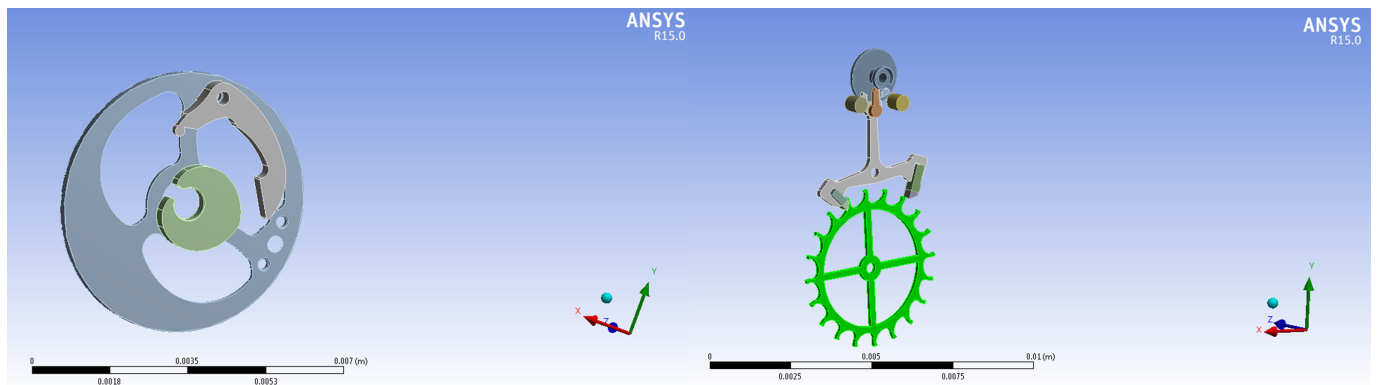


Figure 6: Examples from the first set.



(a) Press machine

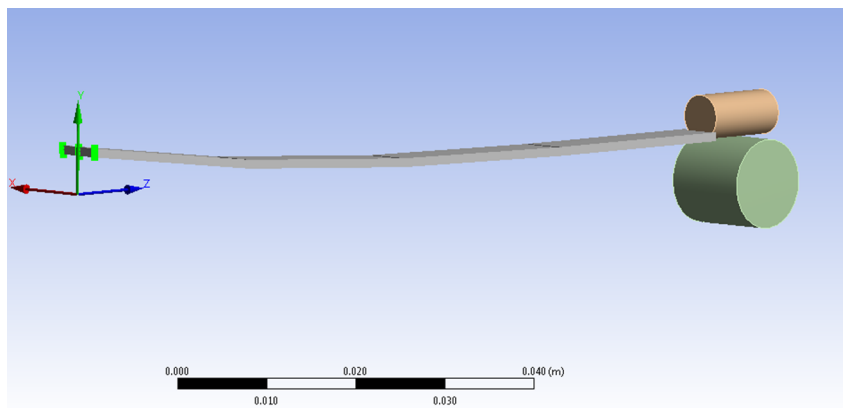
(b) An excavator model



(c) A watch mechanism sub-assembly

(d) Escapement of a mechanical watch

Figure 7: Examples from the second set.



(a) Beam under gravity in contact with a cylinder

Figure 8: Example from the third set.



the method is. The term *performance profile* will be used in the sequel for this concept, which will be clear when used to analyze the simulation results presented in section 4.4.

## 4.4 Simulations results

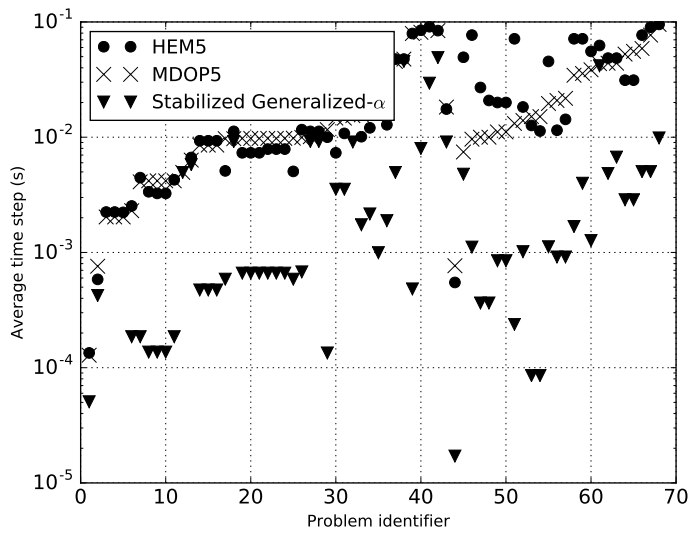
The goal of these simulations is to confirm the preliminary conclusions on real examples that come from industrial applications with non ideal geometries. In order to get some insight into the performance of schemes on large test sets, we use performance profiles. They allow us to draw some general conclusions that are valid in a statistical manner.

We collected some simulation data: the simulation time, the average time step size and the number of steps performed by each numerical scheme. Figures 9, 10, and 11 show the average time step and the number of steps for each problem and for each solver, as well as the distribution function  $\rho_s$  for each solver, with different values of  $\tau$ .

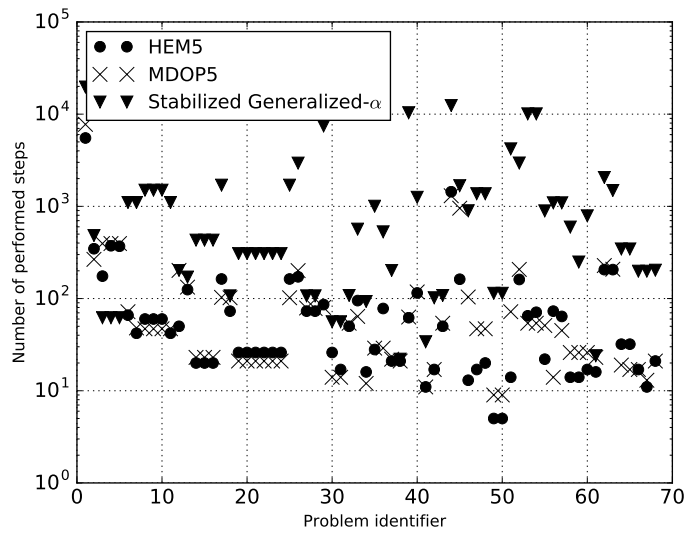
For the first set with very few if no bilateral constraints, we observe in Figure 9 that MDOP5 and HEM5, having the same consistency order ( $p = 5$ ), present the same performances in terms of the time step size used and the number of performed steps. The stabilized generalized- $\alpha$  method being of a lower order ( $p = 2$ ), it needs smaller time steps to reach the defined tolerance on the integration error, which also leads to making more integration steps than the two other methods.

For the second test, the performance profiles of Figure 10 show that when the dynamics is more challenging (large number of degrees of freedom, more non linear terms, large number of constraints to enforce and contact), the HEM5 scheme proves its robustness over the other methods. Indeed, in addition to its high accuracy order, HEM5 exactly solves the velocity constraints and maintains as well the violation of the position constraints at low levels. Therefore, most of the time it does not need any procedure of correction of the constraints, which is time-consuming. MDOP5 solves the constraints on the acceleration level and therefore may require an enforcement of the position and the velocity constraints. The generalized- $\alpha$  method based on the GGL method still requires smaller time steps than those used for HEM5 because it is of a lower accuracy order (equal to 2).

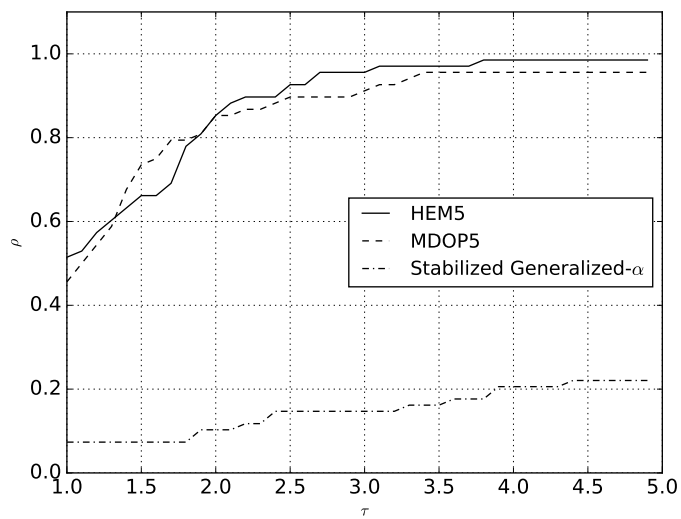
From the graphics of Figure 10, we can see that the stabilized generalized- $\alpha$  and MDOP5 need smaller time step sizes and a larger number of steps to solve the dynamics than HEM5. Furthermore, HEM5 stands out for solving more problems than MDOP5 or stabilized generalized- $\alpha$  for different values of the time ratio  $\tau$ , when the problems involve large number of degrees of freedom and constraints. Table 3 shows a comparison between the three schemes for  $\tau = 1.2$ . The value  $\rho_s(1.2)$  defines the percentage of problems solved by a solver  $s$  in a ratio of 1.2 with respect to the fastest solver.



(a) *haverage*

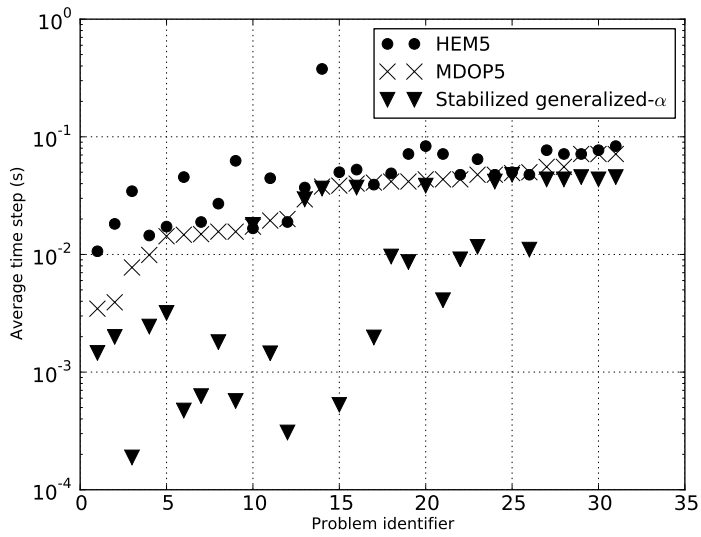


(b) # steps

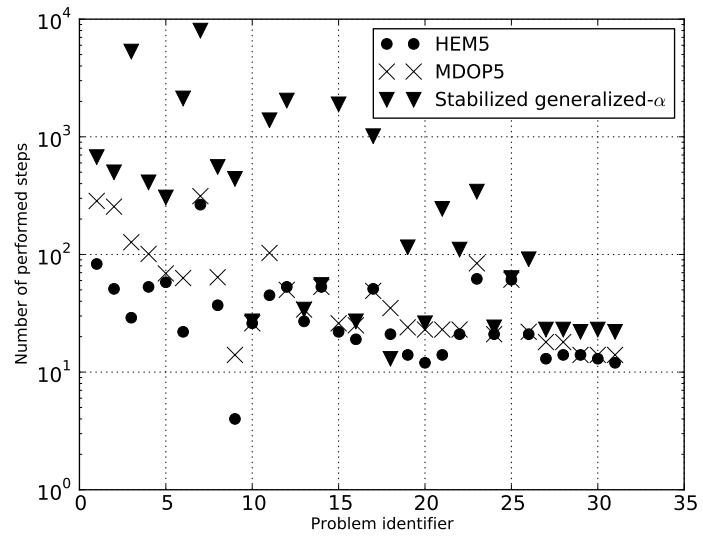


(c) performance profile

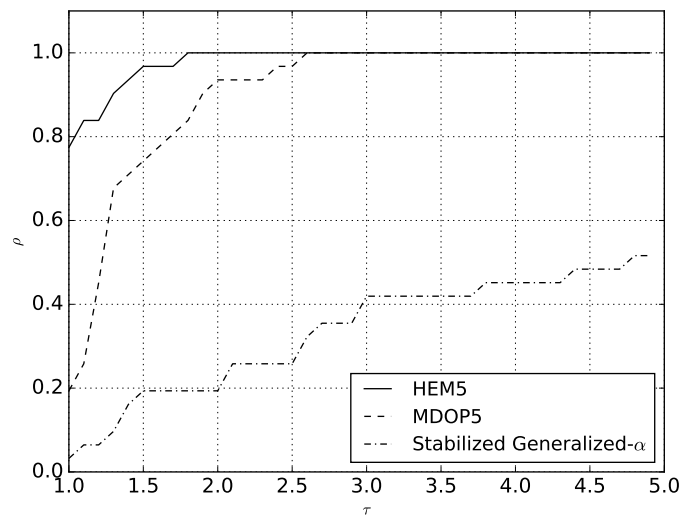
Figure 9: Average time step, number of iterations and performance profile of the first set.



(a) *haverage*



(b) # steps



(c) performance profile

Figure 10: Average time step, number of iterations and performance profile of the second set.

Table 3: Comparison of  $\rho$  between HEM5, MDOP5 and stabilized generalized- $\alpha$  for  $\tau = 1.2$ , for the various test sets

Set number	$\rho_{HEM5}(1.2)$	$\rho_{MDOP5}(1.2)$	$\rho_{StabilizedGeneralized-\alpha}(1.2)$
1	0.57	0.54	0.07
2	0.84	0.45	0.07
3	0.62	0.15	0.46

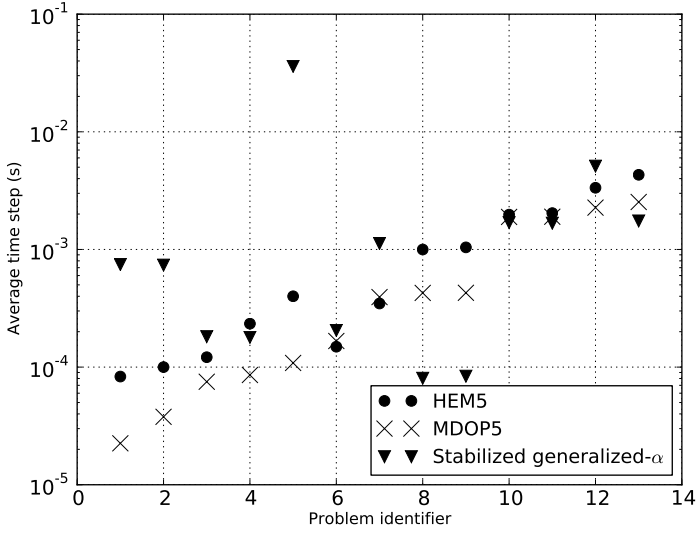
However, it is important to mention that the HEM5 scheme failed to solve the dynamics of a couple of mechanisms subject to contact constraints. Therefore, the results corresponding to the simulation of these mechanisms are not reported. The investigations have shown that the failure is due to an inconsistency in the matrix of (9) that becomes rectangular because  $G^\alpha(q)$  and  $G^{\alpha T}(q)$  do not have the same rank. This issue is discussed in Section 4.5.

In Fig. 11, it is worth noting that the stabilized generalized- $\alpha$  is the fastest solver for 40% of the problems of the third set. In Fig. 11d), we plot the performance profile for larger value of  $\tau$ . The value  $\rho(\tau)$  for larger value of  $\tau$  give a indication on the robustness of the method. Indeed, we leave more time to a method to succeed within a given ratio with the best solver. In terms of robustness, we note that the stabilized generalized- $\alpha$  is also the first to solve all the tests.

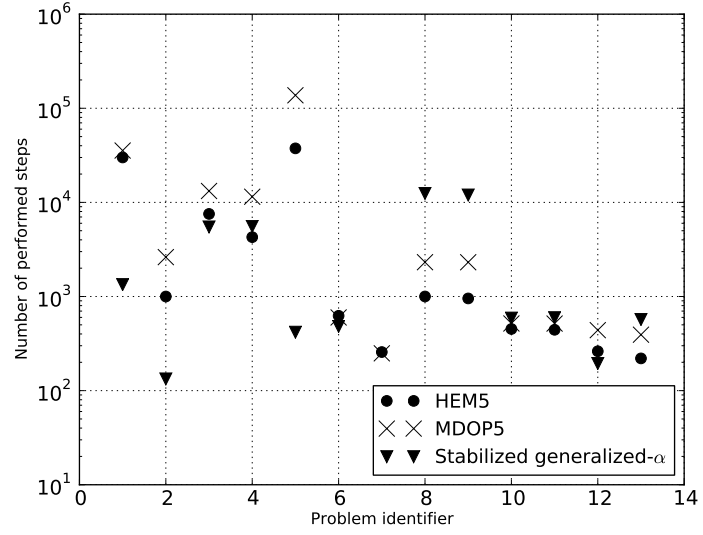
Most of the tests of this set involve systems with linear beams whose dynamics is not very stiff, which explains the fact that HEM5 still outperforms the implicit stabilized generalized- $\alpha$  method for some problems, as we see in Fig.11, for which HEM5 uses larger time step sizes. **Furthermore, HEM5 is faster than the fully explicit MDOP5.** Indeed, when considering systems with flexible bodies, implicit schemes are the best option. If explicit schemes are to be used, then low order schemes are to be selected preferably, to avoid stability issues leading to very small time steps. Moreover, we can note that the solvers failed to solve around 30% of the problems for the given values of  $\tau$ . However, the resolution of all the problems could be achieved for very large values of  $\tau$ .

Let us note that the problem whose identifier is 5, 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 and **which** 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 4. The increasing velocity of the beam leads to a geometric stiffening which makes the computation of the dynamics more challenging. We can see that the stabilized generalized- $\alpha$  method uses time step sizes that are 10 to 100 times larger than those used for HEM5 and MDOP5. It is also less time-consuming than HEM5, and much less time-consuming than MDOP5. As expected, fully implicit time stepping schemes are more relevant for stiff mechanical systems.

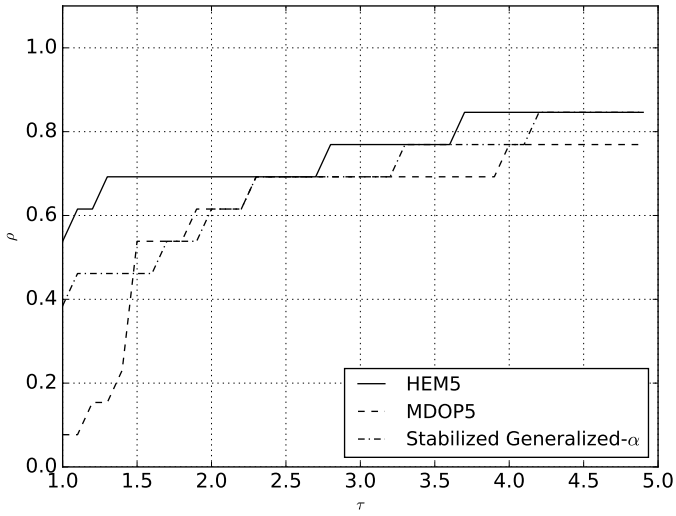
Let us consider another challenging mechanism consisting of a rotor mechanism that has been analyzed in [?], depicted in Fig.12. The system is composed of a flexible shaft, a symmetric disk, and a bearing. The axis of the rotor is along the  $y$  axis



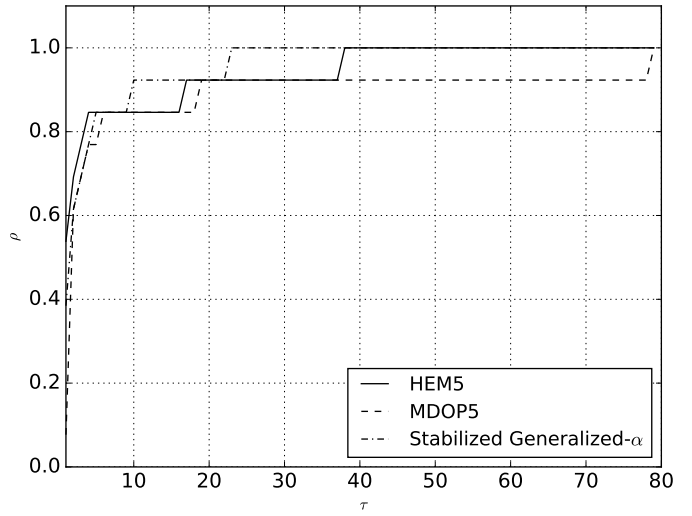
(a) *h* average



(b) # steps



(c) performance profile for  $\tau \in [1, 5]$



(d) performance profile for  $\tau \in [1, 80]$

Figure 11: Average time step, number of iterations and performance profile of the third set.

Table 4: 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	$10^{-2}$	$4.22 \cdot 10^{-4}$	548.98	34123	1381
	$10^{-4}$	$4.20 \cdot 10^{-4}$	549.70	34364	1393
	$10^{-6}$	$2.40 \cdot 10^{-4}$	985.79	62255	179
MDOP5	$10^{-2}$	$1.09 \cdot 10^{-4}$	622.453	137176	833
	$10^{-4}$	$1.08 \cdot 10^{-4}$	632.44	137386	899
	$10^{-6}$	$1.08 \cdot 10^{-4}$	634.94	137344	913
Stabilized generalized- $\alpha$	$10^{-2}$	$7.65 \cdot 10^{-2}$	3.59	196	0
	$10^{-4}$	$6.49 \cdot 10^{-3}$	33.42	2310	0
	$10^{-6}$	$5.40 \cdot 10^{-3}$	33.9218	2687	0

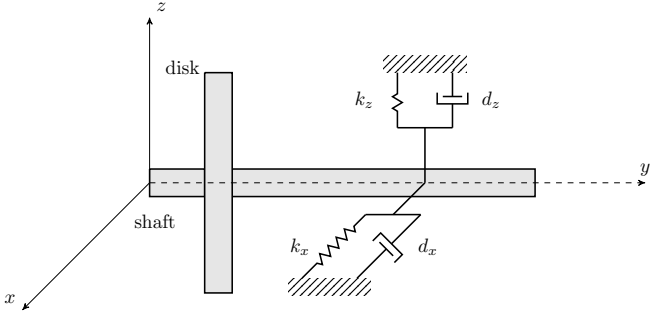


Figure 12: Rotor mechanism

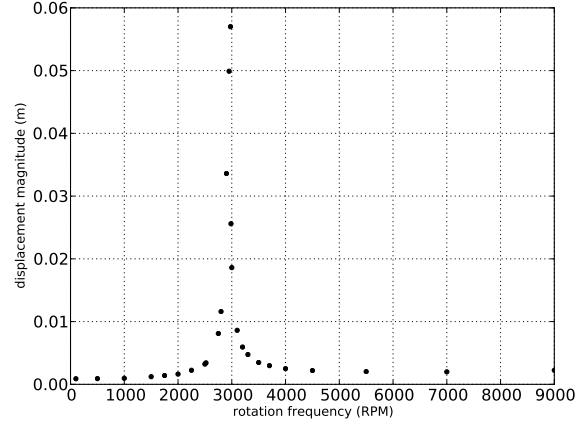


Figure 13: Amplitude of the displacement of the rotor

and displacements are allowed in the  $y$  and  $z$  directions. The system exhibits frequencies that are of the order of magnitude of  $10^5$ Hz. This makes it extremely hard for HEM5 and MDOP5 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 stabilized generalized- $\alpha$  scheme using a constant time step of 0.0004s. The amplitude of the displacement of the rotor with respect to the rotation velocity is shown in Fig.13. This is another case that proves the robustness and efficiency of the implicit schemes over explicit and half-explicit methods when the dynamics is stiff.

## 4.5 A Comment about half-explicit methods

Let us recall that half-explicit methods use linear systems with non-symmetric matrices (31). Therefore the solution exists under particular conditions addressed in [?].

For all the schemes, a QR decomposition is performed on the jacobian matrices in order to eliminate the redundant constraints. Redundant constraints can for example be due to a large number of contact points being generated by the geometry engine. It may happen in some industrial applications with several contact pairs, that the rank of the jacobian matrices involved are different between the stages  $i$  and  $i + 1$  and the QR decomposition leads to a rectangular matrix in (31). The resolution of the system (31) requires therefore different methods than those that are usually used in the solver, or reducing the time step size so that both jacobian matrices have the same rank. The heuristic required to address this issue proves to be complex and fragile, thus not suited for industrial software.

This problem has not been faced when simulating the dynamics of the systems belonging to the four sets described in Section 4.2, but rather when using the HEM5 method to simulate the dynamics of watch mechanisms for example. Indeed, such mechanisms may have several contacts defined, and the geometry engine may find out several contact pairs whose number may change from a stage to another. This leads to a failure in the resolution of (31). This problem might be solved for example by changing some tolerances on the detection of the active contacts or by introducing a "tacking" over time of the contact. But such strategies are quite complex and thus hard to use robustly in an industrial application. This problem turns out to be a considerable drawback for HEM5, and for the half-explicit schemes in general. From this point of view, an attempt of a full implementation of the PHEM56 [?, ?] solver has been stopped after first tests on the industrial problems. An efficient implementation of partitioned half explicit scheme has to be reconsidered in the context of redundant constraints.

## 5 Conclusion

This paper aims at performing a thorough comparison between index-3, index-2 and index-1 DAE solvers. To this purpose, we chose several methods: the generalized- $\alpha$  scheme with an enforcement of both position and velocity constraints, the HEM5 which solves the velocity constraints, and the MDOP5 methods which takes into account the constraints on the acceleration level. These schemes have been compared on a wide range of industrial benchmarks. The comparison covers three aspects: stabilization of the violation 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 shown in this paper and on many other industrial applications, 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 violation of the constraints and the behavior in case of stiff dynamics, the following salient conclusions may be drawn:

- *Computational effort.* Even though the HEM5 solver contains more stages (eight) than the MDOP5 scheme (five),

the numerical effort of the semi-explicit solver is lower than that of the explicit one when a strategy of time step control is used. Indeed, the ability of HEM5 to reduce the violation of the constraints, makes it save the computational effort because it does not need any external procedure of correction of the constraints, unlike MDOP5. For non-stiff mechanisms, both explicit and half-explicit methods give better results than the stabilized generalized- $\alpha$  scheme that needs very small time step sizes mainly due to its lower order (**two**).

- *Violation of the constraints.* As said several times before, to keep the violation of the constraints at low level is of utmost importance for event-driven schemes. A correct approximation of the constraints at the position and velocity levels helps us to evaluate and stabilize the index sets of active constraints. At the acceleration level, an accurate and stable evaluation of the constraints results in an accurate and stable Lagrange multiplier, which is also important for the evaluation of closed bilateral constraints. In particular, schemes that solves index-3 DAE may yield to instabilities in the Lagrange multiplier that can blur the evaluation of closed contact. HEM5 enforces the constraints at the velocity level. **This** enables to reduce drastically the violation at the position level. Often, **the integration can even proceed without** any procedure of projection of the constraints on the admissible manifold. This is not the case for the explicit scheme where a projection on the constraints is required in most cases. The stabilized generalized- $\alpha$  scheme used with an enforcement of the position and the velocity constraints gives better results than using it in its index-2 DAE form, and much better results when it is used in its index-3 DAE solver form. Discretizing the dynamics with a formulation of the constraint at the velocity level seems to be a good compromise to stabilize the violation at required 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 stabilized generalized- $\alpha$ . We even saw that when high frequencies are involved, as in the case of the rotor mechanism, HEM5 and MDOP5 were unable to integrate the dynamics with the chosen practical minimal time step size. In such cases, the stabilized generalized- $\alpha$  method proves its efficiency by handling this high frequency non-linear dynamics by the ability of introducing some numerical damping.
- *Industrial benchmarks.* The tests on industrial benchmarks reveal some results that do not confirm some of the conclusions drawn by the theoretical studies. In particular, the use of half-explicit schemes is difficult in practice mainly due to the use of non symmetric matrices that are not full rank if the constraints are redundant. Furthermore, the interaction with a CAD geometry module for the evaluation of the constraints and the Jacobian of the constraints renders difficult the use of numerical time integration schemes that are not robust with respect to the redundancy of constraints and the nonsmoothness of the contact surfaces. This is particularly the case for index-1 DAE solvers of higher order.

Future work will deal with the comparison between time-stepping and event-driven methods, and the development of a hybrid event-driven/time-stepping scheme. This strategy will help us to avoid the difficulties that event-driven schemes face in case



of accumulation of events, friction, and presence of geometrical discontinuities. Indeed, time-stepping schemes have proved to be robust with respect to such issues [?].