**CSMA 2013**
11e Colloque National en Calcul des Structures
13-17 Mai 2013

# Comparison of index-2 and index-1 DAE solvers for nonsmooth multibody systems with unilateral or bilateral constraints

Mounia Haddouni [1,2], Vincent Acary [1], Jean-Daniel Beley [2], Stéphane Garreau [2]

[1] INRIA,Bipop project team. mounia.haddouni@inria.fr, vincent.acary@inria.fr
[2] ANSYS, France. jean-daniel.beley@ansys.com, stephane.garreau@ansys.com

**Abstract** — The aim of this work is to study some numerical solvers for the integration of index-1 and index-2 differential algebraic equations that describe the dynamics of constrained mechanical systems, in the framework of event-driven schemes.
These solvers will be compared in terms of drift of the constraints, implementation and the way they handle cases when the constraints are not sufficiently differentiable.

**Keywords** — Nonsmooth mechanics, event-driven schemes, constraints, DAEs, ODEs, numerical solvers, LCP, NSCD

## 1 Introduction

Using the Lagrangian formalism, the dynamics of a rigid multibody system with $m$ frictionless contact points can be described by

$$\begin{cases} \dot{q}(t) = v(t) & \text{(1a)} \\ M(q(t))\dot{v}(t) = F(t,q(t),v(t)) + G^T(t,q(t))\lambda(t) & \text{(1b)} \\ 0 \leq \lambda(t) \perp g(q(t)) \geq 0, & \text{(1c)} \end{cases}$$

where $q(t) \in \mathbb{R}^n$ is the generalized coordinates vector, $v(t) \in \mathbb{R}^n$ is the generalized velocities vector, $M(q(t)) \in \mathbb{R}^{n\times n}$ is the symmetric definite positive matrix of inertia, $F(t,q(t),v(t))$ comprises the external applied loads, the non linear interactions between bodies and the non linear inertial terms, $g(q(t)) \in \mathbb{R}^m$ is the vector of constraints, $G = \frac{\partial}{\partial q} g(q(t))$ is the Jacobian of the constraints and $\lambda(t) \in \mathbb{R}^m$ is the Lagrange multiplier which is related to the force at the contact point. The complementarity condition (1c) models the unilateral contact, or the Signorini condition at the position level.

We suppose that impacts occur in an infinitely short period so that the displacements of the bodies during the contact period can be negligible and we use the Newton impact law with a coefficient of restitution $e$. If $g(q(t)) = 0$, we compute the impulse $p_i$ and the post-impact velocity $v^+(t_i)$ by solving the impact equations

$$\begin{cases} M(q(t_i))(v^+(t_i) - v^-(t_i)) = G^T(t_i,q(t_i))p_i \\ U_N^+(t_i) = G(t_i,q(t_i))v^+(t_i) \\ U_N^-(t_i) = G(t_i,q(t_i))v^-(t_i) \\ 0 \leq U_N^+(t_i) + eU_N^-(t_i) \perp p_i \geq 0, \end{cases} \tag{2}$$

where $U_N(t) = \frac{dg}{dt}(q(t)) = G(q(t))\dot{q}(t)$ is the relative normal velocity at contact. The complementarity condition in (2) describes the Signorini condition written at the velocity level, augmented of the impact law.

When an event-driven strategy is chosen for the time integration strategy of nonsmooth multibody systems with unilateral or bilateral constraints, any solver for differential algebraic equations (DAE) can be used between two events for the time integration of the smooth systems with equality constraints $g(q) = 0$. Three index sets are generally introduced in order to characterize the state of the contacts:

- The index set $I_0$ of all possible constraints in the system: $I_0 = \{1,...,m\}$
- The index set $I_1$ of all closed contacts: $I_1 = \{i \in I_0 | g_i(q(t)) = 0\}$
- The index set $I_2$ of all persistent contacts: $I_2 = \{i \in I_1 | \dot{g}_i(q(t)) = 0\}$

From the numerical point of view, the smooth dynamics is integrated with some DAE solver, during periods of time over which the index set $I_2$ is constant. An event detection is performed to find the time where a new contact occurs or a contact needs to be released. When an event is detected, the impact equations (2) are solved and the state of the dynamical system is updated. A complete algorithm for the event-driven schemes is detailed in [1]. As we said above, between two events, we have to solve the index-3 DAE (3) over the closed contact set $I_2$

$$\begin{cases} \dot{q}(t) = v(t) & \text{(3a)} \\ M(q(t))\dot{v}(t) = F(t,q(t),v(t)) + G^T(t,q(t))\lambda & \text{(3b)} \\ g^{\alpha}(q(t)) = 0, \lambda^{\alpha} \geq 0, \quad ,\alpha \in I_2. & \text{(3c)} \end{cases}$$

It is well known that index-3 DAEs are difficult to numerically handle. Therefore, the dynamics is usually integrated 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 the constraint 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 $g$ is differentiated once with respect to time, one obtains an index-2 DAE, we solve in this case (3a) and (3b) together with

$$G(t,q(t))v(t) = 0. \tag{4}$$

If $g$ is twice differentiated, one gets an index-1 DAE composed of (3a) and (3b) with

$$G(t,q(t))\dot{v}(t) + G_{qq}(v,v) = 0. \tag{5}$$

Numerically, the index reduction leads to the phenomenon of drift of the constraints, at different levels depending on the index. Indeed, as opposed to the continuous time case, enforcing (5) in discrete time does not imply that (3c) and (4) are satisfied.

# 2 Quick recall on the numerical time-integration schemes

## 2.1 An index-2 DAE solver. The HEM5 solver

The HEM5 solver is based on a half-explicit 5 order Runge-Kutta method with 8 stages, described in [2]. Half-Explicit Runge-Kutta methods have been introduced by Ernst Hairer in [3]. The HEM5 solver is suited to the integration of DAEs, reduced to the index 2 formulation. Let us denote $q_n$, $v_n$ and $t_n$ the values of position, velocity and time at the beginning of a time step, and let $h$ be the length of the time step. The numerical approximation is given by

$$\begin{cases} M(Q_i, t_n + c_i h)\dot{V}_i = F(t_n + c_i h, Q_i, V_i) + G^T(t_n + c_i h, Q_i)\Lambda_i \\ \dot{Q}_i = V_i \\ G(t_n + c_i h, Q_i)V_i = 0, \end{cases} \tag{6}$$

where the stages are evaluated as follows : $Q_i = q_n + h \sum\limits_{j<i} a_{ij}\dot{Q}_j$ and $V_i = v_n + h \sum\limits_{j<i} a_{ij}\dot{V}_j$.

At each stage, the estimations of position and velocity $Q_i$ and $V_i$ are explicitly computed, while $\Lambda_i$ and $\dot{V}_i$ are implicitly obtained by solving the following linear system

$$\begin{bmatrix} M_i & -G_i^T \\ G_{i+1} & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} F_i \\ r_i \end{bmatrix}, \tag{7}$$

where $r_i = -\dfrac{G(Q_{i+1})}{ha_{i+1,i}}(v_n + h \sum\limits_{j=1}^{i-1} a_{i+1,j}\dot{V}_j)$.

At the end of the time step, the numerical solution is given by $q_{n+1} = Q_9 = q_n + h\sum_{i=1}^{8} b_i\dot{Q}_i$ and $v_{n+1} =$

$V_9 = v_n + h\sum_{i=1}^{8} b_i \dot{V}_i$, with $b_i = a_{9i}$. The computation of coefficients $c_i$ and $a_{ij}$ is detailed in [2]. It is worthwhile to note that the HEM5 solver is constructed in a way that the constraint at the velocity level is satisfied at each stage: $G(Q_i)V_i = 0\,,\forall i = 1..8$. 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:

$$\begin{bmatrix} M & -G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \lambda \end{bmatrix} = \begin{bmatrix} F \\ r \end{bmatrix}, \tag{8}$$

where $r = -G_{qq}(v,v)$.

The embedded schemes technique is the most suited way to compute the optimal time step when we deal with Runge-Kutta schemes. Concerning the HEM5 solver, V.Brasey and E.Hairer [2] define an error based on the $7^{th}$ and $8^{th}$ estimations such that

$$err = \frac{{err_1}^2}{err_1 + 0.01 err_2} = O(h^5), \tag{9}$$

with

$$err_1 = \|q_{n+1} - Q_8\|_s = O(h^4) \text{ and } err_2 = \|q_{n+1} - q_n - h(\frac{5}{2}Q_7 - \frac{3}{2}Q_8)\|_s = O(h^3).$$

Finally, The optimal step size is computed with (10):

$$h_{opt} = s\,h(\frac{tol\ h}{err})^{\frac{1}{p}}, \tag{10}$$

with $p = 5$ is the local order of the method, $s$ a safety factor, and $tol$ the user defined tolerance.

### 2.2 A standard ODE solver. The Runge-Kutta-Fehlberg scheme

This method is suited for the integration of ODEs. It belongs to the family of Runge-Kutta-Embedded methods and provides 2 approximations of the solution to evaluate the error. The scheme is based on 6 estimations of the derivatives defined as: $Y_i = y_n + hF(\sum_{j<i} a_{ij}\dot{Y}_j)$.

In our case, we have: $Y = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$ and $F = \begin{bmatrix} v \\ -M^{-1}G^T(GM^{-1}G^T)^{-1}((Gv)_q v + GM^{-1}f) + M^{-1}f \end{bmatrix}$.

A $4^{th}$ and $5^{th}$ order approximations of the solution at the end of the step are computed by $y_{n+1} = y_n + \sum_{i\leq 6} b_i\dot{Y}_j$ and $\tilde{y}_{n+1} = y_n + \sum_{i\leq 6} \tilde{b}_i\dot{Y}_j$. The computational error is defined with:

$$err = \|\tilde{y}_{n+1} - y_{n+1}\|. \tag{11}$$

Finally, the computation of the optimal step size is given by (10) with $p = 4$.

## 3 Numerical study of the control of the drift of the constraints

### 3.1 The four-bar linkage

In this section we consider the four-bar linkage system, with 4 bars linked with revolute joints, and one of them being grounded, as described in Fig. 1:

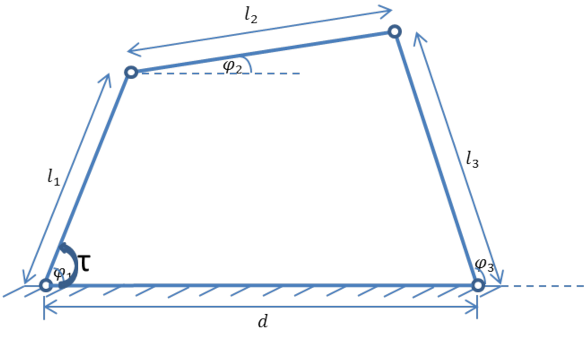
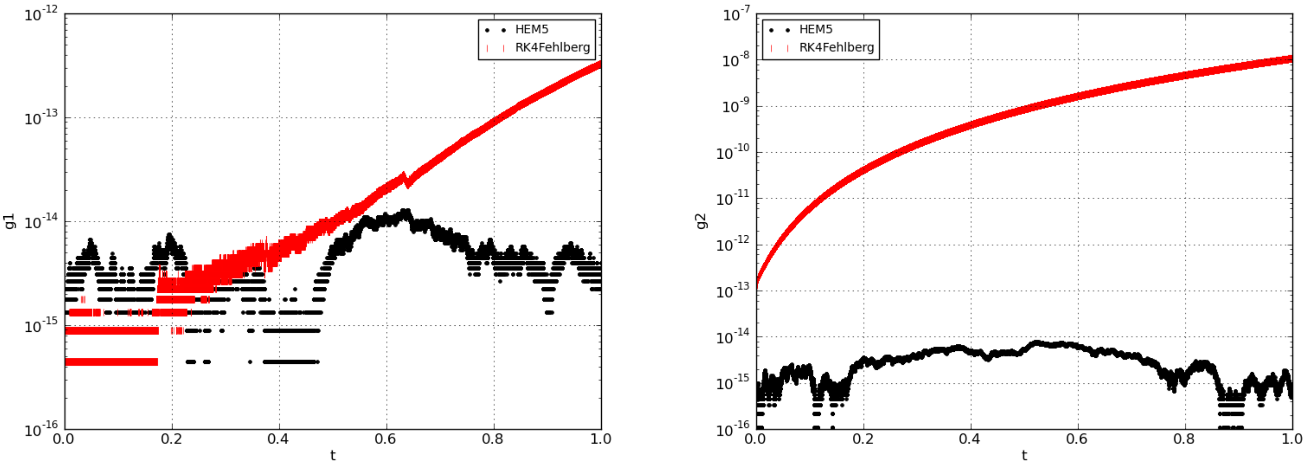


Fig. 1: Four-bar linkage

This system is driven by a torque of value $\tau = 6N.m$ and is described by the vector of generalized coordinates $q = [\varphi_1, \varphi_2, \varphi_3]$ as depicted on the figure. During the simulation, the lengths of the bars must be kept constant. To this aim, two constraints are defined:

$$\begin{cases} g_1(q) := l_1\cos(\varphi_1) + l_2\cos(\varphi_2) - l_3\cos(\varphi_3) - d_1 = 0 \\ g_2(q) := l_1\sin(\varphi_1) + l_2\sin(\varphi_2) - l_3\sin(\varphi_3) = 0. \end{cases} \tag{12}$$

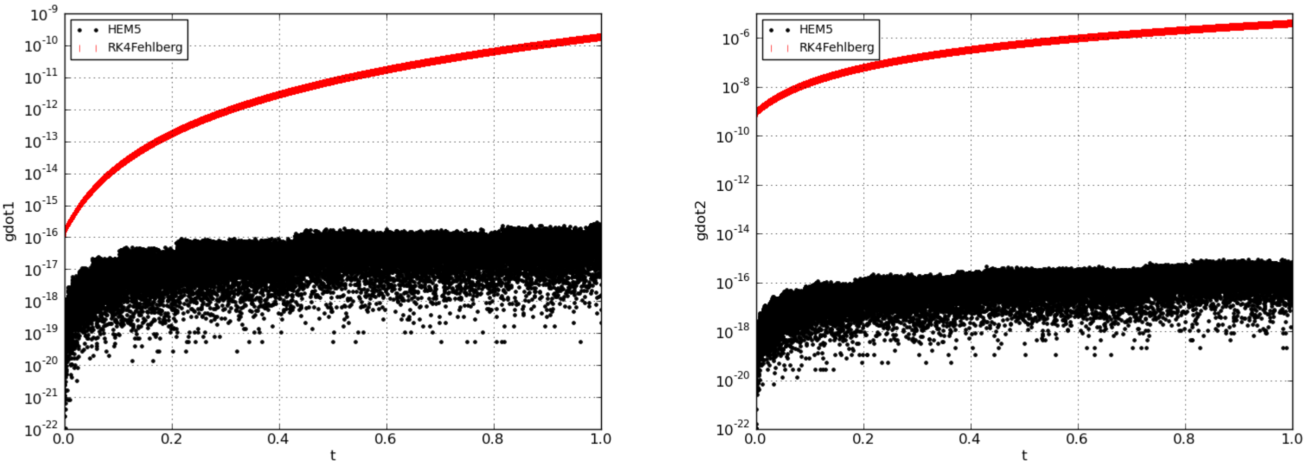
The initial conditions of position and velocity are given by $q_0 = [\frac{\pi}{2}, 0, \frac{\pi}{6}]^T$ and $v_0 = [0,0,0]^T$. They are compatible with the constraints at both position and velocity levels. The time step size is controlled in order to meet a tolerance of $10^{-10}$ on the integration error but also to maintain the drift of the constraints (at position and velocity levels) at an acceptable level ($10^{-6}$). We set the minimum step size at $10^{-6}s$ and the maximum step size at $10^{-2}s$. The drift of the constraints at both position and velocity levels, for both solvers, are plotted in Fig. 2 and Fig. 3.



(a) $g_1$ versus time.

(b) $g_2$ versus time.

Fig. 2: Four bar linkage example. Drift of the constraints at the position level



(a) $\dot{g}_1$ versus time.

(b) $\dot{g}_2$ versus time.

Fig. 3: Four bar linkage example. Drift of the constraints at the velocity level

For the levels of accuracy that we set, the Runge-Kutta-Fehlberg scheme uses the minimum step size ($10^{-6}s$) at every time step. This means that the prescribed tolerance is hardly achieved by the scheme in this range of time–steps. The time step sizes needed for the simulation with the HEM5 and the RKFehlberg schemes are presented in Fig. 4.

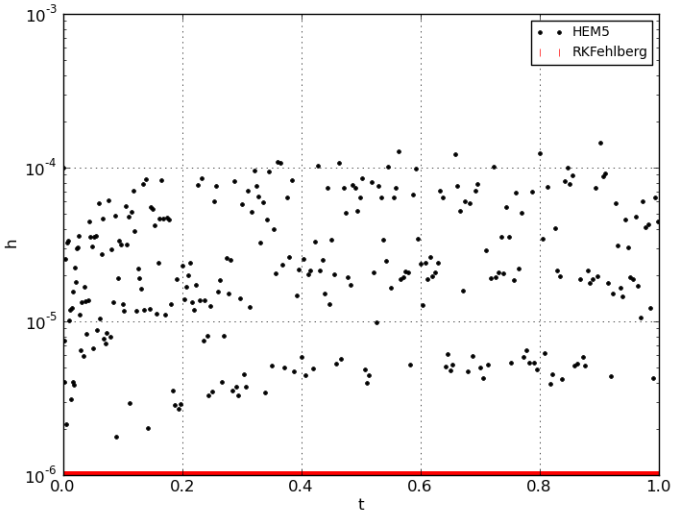


Fig. 4: Four-bar linkage. Evolution of the time step with the simulation time.

Thanks to its intrinsic characteristic of maintaining the constraint at the velocity level, the HEM5 solver reduces the drift of the constraint drastically at both position and velocity levels, and for a much lower computational effort compared to the Runge-Kutta-Fehlberg scheme.

### 3.1.1 The rocking block

The aim is to study the motion of the center of mass of the block depicted in Fig. 5.

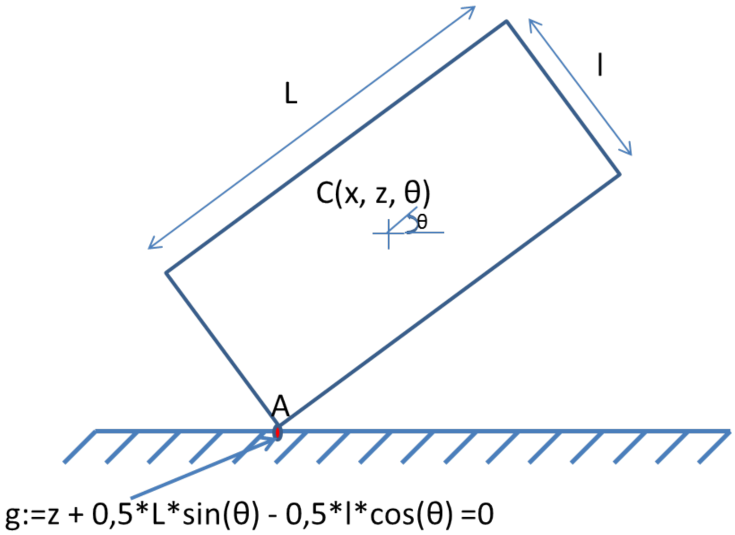
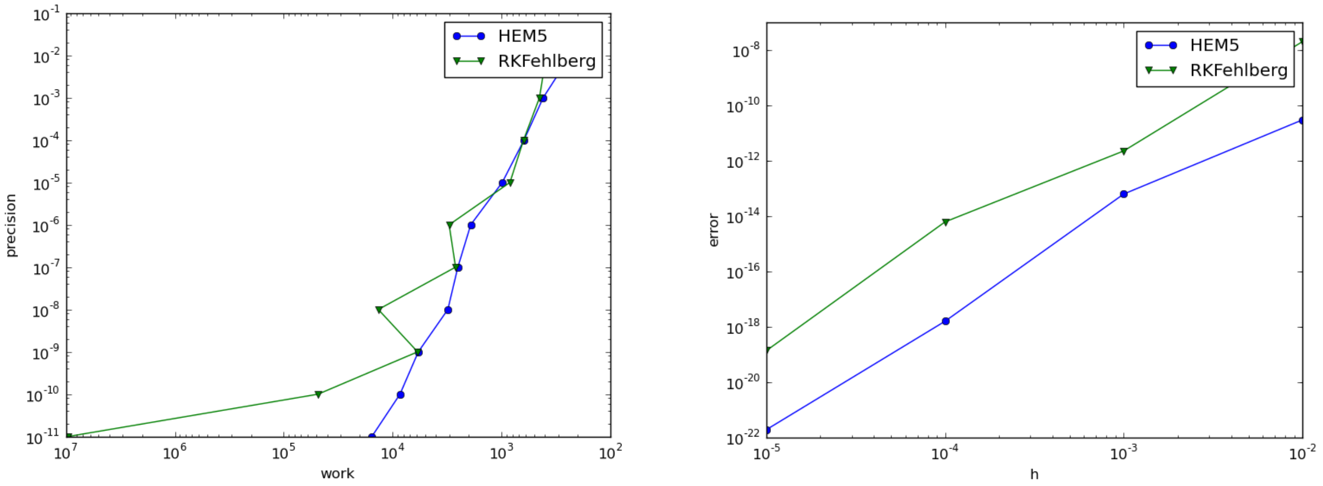


Fig. 5: Rocking block

The motion of the block can be described by the generalized coordinates $q = [x, z, \theta]$. During the simulation time ($0.5s$), the point $A$ of the block should be in contact with the ground. We varied the tolerances on the truncation error and on the drift of the constraints, and then we varied the time step size. The Work/Precision and the error/$h$ diagrams are given in Fig. 6a and Fig. 6b.



(a) work precision diagram of the rocking block, variable time step

(b) log(error)/log(h) diagram of the rocking block, fixed time step

Fig. 6: error and precision diagrams for the rocking block

We can see that holding the drift of constraints under very low thresholds enables the HEM5 solver to reduce the computational effort for high user-defined tolerances. The Runge-Kutta-Fehlberg needs lower step sizes to meet the tolerance on the drift of the constraints.
We varied the tolerance for both HEM5 and RKfehlberg schemes, and we computed the average time step size, the computational effort[1] and the maximum drift of $g$ and $\dot{g}$. Results are detailed in Table 1 and 2.

| tolerance | average $h$ | work | $g_{max}$ | $\dot{g}_{max}$ |
|---|---|---|---|---|
| $10^{-2}$ | 0.336 | 234 | $3,47.10^{-5}$ | $4,81.10^{-16}$ |
| $10^{-3}$ | 0.260 | 460 | $1,1310^{-7}$ | $3,3410^{-16}$ |
| $10^{-4}$ | 0.179 | 624 | $6,3410^{-8}$ | $6,7210^{-16}$ |
| $10^{-5}$ | 0.104 | 988 | $3,1610^{-9}$ | $6,9710^{-16}$ |
| $10^{-6}$ | 0.056 | 1924 | $4,7610^{-10}$ | $4,5810^{-16}$ |
| $10^{-7}$ | 0.054 | 2524 | $3,7810^{-10}$ | $6,4510^{-16}$ |
| $10^{-8}$ | 0.046 | 3120 | $610^{-11}$ | $6,1310^{-16}$ |
| $10^{-9}$ | 0.026 | 5824 | $8,0110^{-11}$ | $6,8410^{-16}$ |
| $10^{-10}$ | 0.018 | 8580 | $1,2910^{-11}$ | $7,1410^{-16}$ |
| $10^{-11}$ | 0.010 | 15548 | $4,1710^{-11}$ | $9,1710^{-16}$ |

Table 1: Rocking block: average $h$, work and maximum drift with HEM5

| tolerance | average $h$ | work | $g_{max}$ | $\dot{g}_{max}$ |
|---|---|---|---|---|
| $10^{-2}$ | $0,065$ | 390 | $1,710^{-4}$ | $9,510^{-5}$ |
| $10^{-3}$ | $0,063$ | 450 | $2,2110^{-5}$ | $8,0710^{-5}$ |
| $10^{-4}$ | $0,051$ | 630 | $1,2110^{-5}$ | $8,0710^{-5}$ |
| $10^{-5}$ | $0,039$ | 840 | $2,0210^{-6}$ | $2,3810^{-6}$ |
| $10^{-6}$ | $0,0009$ | 3036 | $5,6210^{-7}$ | $9,9810^{-7}$ |
| $10^{-7}$ | $0,017$ | 2700 | $4,9310^{-8}$ | $9,7510^{-8}$ |
| $10^{-8}$ | $0,0029$ | 13560 | $7,1310^{-9}$ | $10^{-8}$ |
| $10^{-9}$ | $0,0008$ | 5910 | $4,1710^{-10}$ | $9,9610^{-10}$ |
| $10^{-10}$ | $7,6^{-4}$ | 48510 | $5,310^{-11}$ | $10^{-10}$ |
| $10^{-11}$ | $1,9710^{-6}$ | 9585990 | $1.2310^{-11}$ | $9,7910^{-12}$ |

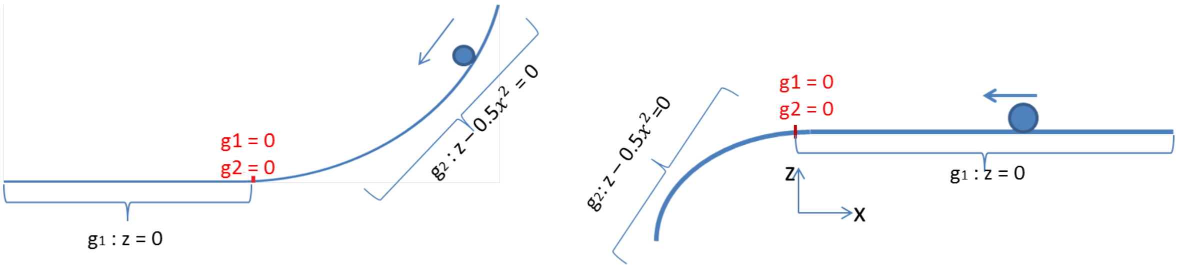
Table 2: Rocking block: average $h$, work and maximum drift with Runge-Kutta-Fehlberg

[1] The computational effort is computed by counting the number of function evaluations and the number of linear equations that are solved.

# 4 Problems with geometrical discontinuities

In this section, we discuss the problems that occur when in a given geometry, there are some edges or other discontinuities in the contact interface that yield nonsmoothness in the constraint. These discontinuities make the integration of the equations of motion more difficult, and could even lead to incoherent results if they are not correctly treated. In the case of problems with geometrical discontinuities (edges for example), the term $G_{qq}$ of Equation (5) is discontinuous, this leads to a jump in the acceleration and in the contact force.

Numerically, these discontinuities result in a difficulty when computing the estimations of acceleration or Lagrange multipliers because the system (7) is not well-posed. Let us illustrate this with two problems presented in Fig. 7a and Fig. 7b.



(a) (b)

Fig. 7: Systems with geometrical discontinuities, illustrative examples

For the system of Fig. 7a, when the sphere arrives at the edge, both constraints are active and with the HEM5 solver for example, the system (7) is not well-posed, here is an example of such a system:

$$\begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 6.65031e^{-5} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_1 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -9.81 \\ 0 \\ -3.62238e^{-2} \\ 4.41971e^{1} \end{bmatrix}$$

At $x = 0$, two constraints are active ($g1 : z = 0$ and $g2 : z - x^2 = 0$), but the Hessian of these constraints are not equal at the edge and this leads to a discontinuity in term $G_{qq}$. We propose two methods to solve this problem.

**$1^{st}$ method: Formulating the system as an LCP** The first solution consists in re-writing the systems for the computation for the estimations of accelerations and Lagrange multipliers (for example system (7) for the HEM5 solver) as a Linear Complementarity Problem:

$$\begin{cases} M(Q_i)\dot{V}_i - G^T(Q_i)\Lambda_i = F_i \\ G(Q_{i+1})\dot{V}_i - r_i \geq 0 \\ 0 \leq \dot{V}_i \perp \Lambda_i \geq 0. \end{cases} \tag{13}$$

This formulation transforms the unfeasible linear problem given by system (7) with two equality constraints into a feasible complementarity problem, which enables to find a solution to the problem. For our simulations, we used for the resolution of these LCPs Lemke's algorithm available in the open-source software SICONOS[2]. When applied to the examples of Fig. 7a and 7b, Lemke's algorithm provides some estimations of the acceleration and the contact forces which correspond to the expected theoretical ones.

**$2^{nd}$ method: Using the NSCD method** The other solution that we propose is to compute the dynamics in the zone that presents discontinuities with a time-stepping scheme, for instance the Non Smooth Contact Dynamics (NSCD) method (Moreau–Jean's time–stepping scheme [4, 5]). The NSCD method is used here to overcome the difficulty triggered by geometries with non-continuous curvatures. More precisely, it is used as a way to sort the constraints to be activated, that is decide the 'direction' to

[2] `http://siconos.gforge.inria.fr`

go through when crossing an edge for example. The following pseudo-algorithm explains the way the NSCD method is used in the whole integration process:

1. Detect time $t_d$ when the discontinuity (edge, undetermined normal) occurs
2. Compute the dynamics with the NSCD method
3. Compute the set $I_d$ of active constraints at the end of the step
4. Go back at time $t_d$ and solve the dynamics with a DAE solver with taking into account only the constraints belonging to $I_d$

We favor the NSCD method as a solution to the presented problems rather than the LCP solution (previous subsection) since it is a physical solution, so the results are only depending on the dynamics of the given problem.

# 5 Conclusion

Even though the HEM5 solver contains more stages (8) than the Rune-Kutta-Fehlberg scheme (6), the numerical effort of the first solver is lower than that of the second one when they are used with an adaptive time step. Indeed, the order (5) of the HEM5 solver in addition to its characteristics of reducing the violation of the constraints, enables using smaller time step sizes than those used for the RKFehlberg scheme, and thus reduces the computational effort.

The HEM5 solver enforces the constraints at the velocity level and it 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 RKFehlberg scheme where a projection on the constraints is mandatory. Discretizing the dynamics with a formulation of the constraint at the velocity level seems to be a good compromise.

None of the solvers enable to avoid integration issues (systems not well-posed when constraints are not compatible) that occur at some geometrical discontinuities. We presented two solutions to this problem: using Lemke's algorithm which gives correct results when the problem is feasible, or using the NSCD method. The last solution is promising since it is only driven by the dynamics and does not depend on any feasibility condition.

Further works will deal with index-3 DAE resolution as in the work of O.Bruls and M.Arnold in [6], and the use of hybrid mixed time stepping integrators in [7] and projected integration schemes in [8].

## References

[1] V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems*. Springer, 2008.

[2] V. Brasey. Hem5 user's guide. 1994.

[3] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer, 2002.

[4] J.J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In J.J. Moreau and Panagiotopoulos P.D., editors, *Nonsmooth Mechanics and Applications*, pages 1–82. CISM 302, Spinger Verlag, 1988.

[5] M. Jean. The non smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177:235–257, 1999.

[6] O. Brüls and M. Arnold. Convergence of the generalized-α scheme for constrained mechanical systems. 2007.

[7] 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.

[8] 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*, in press, 2013.