This version of the article has been accepted for publication, after peer review and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/s11044-023-09896-7

Noname manuscript No.
(will be inserted by the editor)

A benchmark problem with singularities for multibody system dynamics formulations with constraints

Maurizio Ruggiu · **Francisco González**

Received: date / Accepted: date

Abstract When performing the numerical integration of multibody systems (MBS) dynamics, it is possible to choose from a wide variety of methods and implementations. Selecting the most appropriate option for a particular application is not a straightforward task; as a consequence, several benchmark examples have been formulated by the MBS research community with the intent to assess the accuracy and performance of different solution methods when applied to certain kinds of mechanical problems. This paper introduces a variation of the slider-crank mechanism, already employed as a benchmark problem in the MBS literature, intended to evaluate the performance of formulations that feature kinematic constraints. Three cases, featuring singular configurations and external actions, were defined. The example is used to illustrate some necessary elements in the definition of a benchmark problem and in the process of comparing different solution methods, as well as difficulties that can arise during this task. The use of the proposed example was demonstrated in the evaluation of the behaviour of different solution methods, which employed both fixed- and variable-step integration formulas.

Keywords Benchmark problems · Numerical integration · Efficient Methods · Singular configurations

Mathematics Subject Classification (2020) MSC 34 · MSC 65

M. Ruggiu
Dipartimento di Ingegneria Meccanica, Chimica e dei Materiali, Università di Cagliari, Italy
Tel.: +39-70-6755716
E-mail: maurizio.ruggiu@unica.it

F. González
Laboratorio de Ingeniería Mecánica, Industrial Campus, CITENI, University of A Coruña, Spain.

1 Introduction

A considerable number of methods and algorithms for the simulation and analysis of Multibody System (MBS) Dynamics have been proposed since the early developments in this area were first published [1, 16, 21, 22]. The performance of each approach depends on the characteristics of the problems to which it is applied, and so methods that are effective in the simulation of a certain type of mechanical system may be inefficient when applied to mechanisms with a different topology or subjected to other kinds of physical phenomena. Fully recursive methods [9], for instance, may become ineffective in the solution of heavily constrained multibody systems; mechanisms that feature redundant constraints or singular configurations pose a problem for solution algorithms that expect the Jacobian matrix of the constraints to have a full row rank throughout the motion [11, 17]. Moreover, implementation techniques, third-party software libraries, e.g., for the linear algebra routines required by most MBS codes, and the hardware platform used to execute the code, as well as the interaction between them, have a critical impact on the time elapsed in computations [13, 24]. For these reasons, selecting an appropriate MBS formalism for its application to a particular problem may prove challenging in some cases, particularly when efficiency constraints are imposed as a requirement.

Benchmark problems represent a useful tool to evaluate the accuracy and efficiency of MBS codes, as well as their ability to handle particular kinds of problems. Ideally, benchmarks should be simple enough to enable their exact reproduction by any researcher or team interested in using them. At the same time, they must be nontrivial problems that provide interesting information about some aspect of the behaviour of the solution method [14]. In recent years, several initiatives have been put forward by MBS researchers to propose meaningful test problems that can be generally accepted as benchmarks by the community. The IFToMM Library of Computational Benchmark Problems [15] is a well-known collection of such examples, which includes test problems for forward- and inverse-dynamics, as well as linearization. These examples illustrate the performance of MBS formulations and implementations when dealing with complex issues such as redundant constraints, singular configurations, stiff problems, and contacts, to mention just a few. Another instance of collection of benchmark problems can be found in [2], which puts forward a series of cases for the validation of flexible multibody dynamics algorithms. Benchmark problems for contact dynamics were introduced in [19]. In the case of MBS dynamics, benchmarking is not limited to the algorithms for the integration of the equations of motion, but has also been extended to applications in which the multibody part is a necessary component, like estimators based on Kalman filters [20]. Benchmark problems for particular applications of MBS dynamics can be found in the areas of railway vehicles [5] and co-simulation [25].

The benchmark problem discussed in this paper, a variation of the well-known slider-crank linkage, is geared towards the evaluation of the ability of MBS methods to successfully perform the forward-dynamics simulation of mechanical systems with singular configurations and dead centres. This can be challenging, in particular, when the system dynamics is expressed in terms of a set of non-minimal generalized coordinates, thus requiring the use of kinematic constraints to formulate the equations

of motion [11]. Two versions of the linkage were defined, with and without singular configurations, and both were subjected to the action of a force that varied over time, to prevent a periodic system motion.

The paper puts forward a systematic approach for the formulation of benchmark problems, including the definition of relevant metrics and criteria for the comparison of different solution approaches. These are presented in Section 2, together with the description of the mechanical model of the slider crank used here. The use of the benchmark problem is demonstrated in Section 3 in the evaluation of different simulation codes in which the system is defined using kinematic constraints. These were solved using both fixed- and variable-step integration algorithms. Results showed that the combination of singular configurations and externally applied forces with constant frequency made the proposed example a challenging problem for most solution methods. Section 4 summarizes the conclusions of the study.

2 Methods

There exist three components that should always be present in the definition of a benchmark problem for MBS dynamics algorithms:

- A definition of the problem to be solved. This must include the specification of the properties and initial state of the mechanical system, as well as information about the manoeuvre to be simulated, such as duration in time and input forces and torques.
- A reference solution which, for the purposes of benchmarking, can be considered correct. This reference solution may be obtained from experimental results, an analytical solution of the problem at hand, or upon the convergence of several simulation processes.
- Appropriate error metrics and comparison criteria. These enable the assessment of the solutions obtained with different simulation methods [12].

Moreover, other optional components can be added too, such as reference implementations of the simulation code or data structures to enable the efficient collection and processing of simulation results.

In order to be useful, benchmark examples should be clearly defined problems that are easy to replicate. They should also represent nontrivial scenarios, which are meaningful or challenging in at least one respect. Ideally, they should also be representative of a wider class of systems. For instance, a slider-crank that undergoes singular configurations can be defined in a straightforward way with a reduced set of kinematic and kinetic parameters. In spite of being a relatively simple mechanical system, it poses a problem for simulation algorithms that cannot deal with rank-deficient Jacobian matrices; even some solution methods that can handle them need to be carefully adjusted to deliver correct results [11], as is the case of augmented Lagrangian algorithms. Results obtained with this linkage can then be used to assess the general ability of solution methods to carry out the simulation of systems with singularities.

In some cases, it is possible to find an analytical solution for the motion of simple mechanical systems. Generally, this is not the case. In benchmark problems that do

not represent a physical system, for which experimental results are not available, physical magnitudes can be used as indicators instead. For instance, the variation of the mechanical energy can be used to quantify the accuracy of a given simulation method if the benchmark problem represents a conservative system. These indicators should be used with precaution, as a precise energy conservation does not necessarily guarantee the correctness of the obtained results. Arriving at a reference solution through the convergence of several simulation methods is advisable when analytical and experimental solutions are not available.

Error metrics and criteria are also an important component of a benchmark problem. First, it is necessary to select the variables that will be selected to evaluate the accuracy of a solution. These may include kinematic variables, such as positions, velocities, and accelerations, or kinetic magnitudes like forces or energies. Usually n variables of interest can be selected and will suffice to measure the precision of the results. It is also necessary to specify at what points in time these variables will be evaluated. When using fixed-step integrators this is relatively simple, as data can be gathered regularly during the simulation. With variable-step simulators usually interpolation methods have to be used. Second, a metric to quantify the deviation of the results obtained with a particular method with respect to the reference solution is necessary. The local error at time point t_i for variable y_j can be evaluated as

$$\varepsilon_j(t_i) = y_j(t_i) - y_j^{\mathrm{ref}}(t_i) \tag{1}$$

where y_j^{ref} denotes the value that corresponds to the reference solution. Relative definitions of the error can be used too [12]. Absolute errors, however, show a better behaviour when the variables of interest approach zero. The total error of a simulation can be calculated as

$$\varepsilon_{\mathrm{T}} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\frac{w_j}{m}\sum_{i=1}^{m}\left(\varepsilon_j\left(t_i\right)\right)^2} \tag{2}$$

where m is the total number of time points collected during the simulation and w_j is a weight factor that represents the contribution of variable j to the total error. Factor w_j can also be used to make errors dimensionless, so that variables with different units can be added together in a single error indicator. Besides the total error in Eq. (2), it is also possible to select other indicators, such as the maximum or minimum absolute error for a single variable or group of variables.

The existence of a metric like the one in Eq. (2) makes it possible to establish a validity criterion that determines whether a simulation is accurate enough or not. A criterion like this is particularly important if the benchmark problem is to be used to compare several solution approaches in terms of efficiency, because that comparison should be carried out requesting the same accuracy level from every method.

Finally, comparison criteria can be defined to quantify the differences between the different solution methods. The elapsed time in computations is a commonly used criterion to rank algorithms and implementations, but other metrics, such as energy balances and satisfaction of kinematic constraints can be used too [18].

2.1 Problem description

The benchmark problem used in this paper is a variation of the well-known slider-crank linkage, already included in the IFToMM benchmark library [15] and shown in Fig. 1. This planar mechanical system is composed of two rods, links 1 and 2, with uniformly distributed mass m_1 and m_2 and length L_1 and L_2, respectively. The slider, link 3, has mass m_3 and moves without friction along the x axis. The mechanism moves under gravity effects with $g = 9.81$ m/s^2 acting along the negative y axis. The system has one degree of freedom; at time $t = 0$, rod 1 is at an angle $\theta_{1,0}$ with respect to the x axis and the velocity of point Q is $\dot{x}_{\mathsf{Q},0}$. A horizontal external force f acts on point Q during motion.

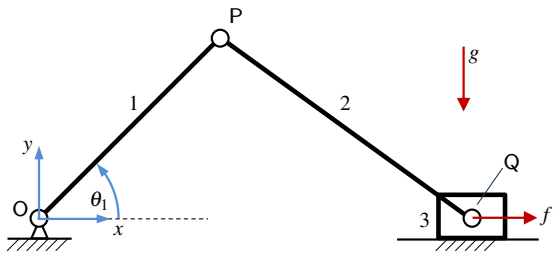

Fig. 1 Slider-crank mechanism used as a benchmark problem.

In spite of its simplicity, the slider-crank linkage is a challenging problem for MBS dynamics algorithms, due to the existence of singularities in the system motion, under certain conditions. A slider-crank with $L_1 = L_2$ will feature a singular configuration when $\theta_1 = \pm\pi/2$, i.e., when both rods are aligned on the y axis. This configuration represents a bifurcation point, from which the linkage can continue its motion either as a slider-crank mechanism or as a pendulum with point Q motionless at the location of point O, as shown in Fig. 2. In theory, switching between these two branches of motion is only possible when the velocity of the slider is zero. However, some MBS dynamics formulations with explicit modelling of the kinematic constraints may predict a change of branch also with a nonzero velocity. From a mathematical point of view, the Jacobian matrix of the kinematic constraints suddenly loses rank at these points. Dynamics formulations unable to deal with rank-deficient Jacobian matrices cannot carry out the simulation of such systems; moreover, some methods which can handle this situation experience numerical difficulties in the proximity of these points as well. This is usually caused by poor conditioning of the algorithm leading matrices, and by the accumulation of constraint violations. The resulting changes of branch introduce large impact forces in the simulation, and instantaneous drops in the mechanical energy of the system [3, 11].

A different kind of numerical difficulty is present when both links, 1 and 2, regardless of their relative lengths, are aligned on the horizontal x axis, i.e., $\theta_1 = 0$ or $\theta_1 = \pi$. In this situation, a horizontal force exerted on point Q of the slider should not have any effect on the system motion and will only increase the reaction force at point

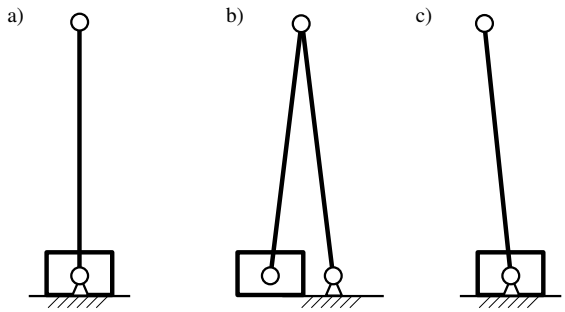

Fig. 2 From a singular configuration (a), a slider-crank linkage with $L_1 = L_2$ can either keep its slider-crank motion (b) or continue as a single pendulum (c).

O. Poor matrix conditioning and the accumulation of position errors in the proximity of this point may result into the motion of the mechanism towards a position with either a positive or a negative angle θ_1 (Fig. 3). It must be noted that, here, a change of branch does not occur, as the system continues moving as a slider-crank linkage regardless of the sign of θ_1.

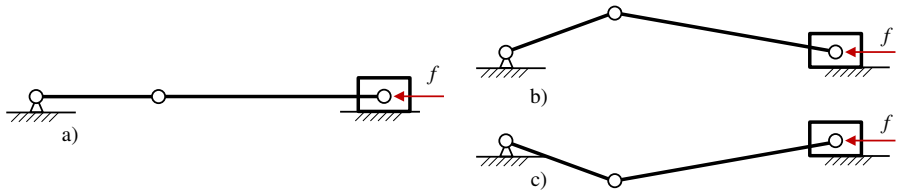

Fig. 3 The application of a horizontal force f in configuration (a) should not cause the motion of the linkage. A misalignment of the links in the predicted system position will move the mechanism towards configurations (b) or (c).

To illustrate these difficulties, we consider now a model of the slider-crank linkage in which the generalized coordinates $\mathbf{q} = [x_{\mathsf{P}}, y_{\mathsf{P}}, x_{\mathsf{Q}}]^{\mathrm{T}}$ are the x and y coordinates of point P and the distance x from point Q to origin O. Because links 1 and 2 are rigid, two kinematic constraints enforcing a constant distance between points O and P, and P and Q have to be introduced to describe the motion correctly:

$$\mathbf{\Phi} = \begin{bmatrix} x_{\mathsf{P}}^2 + y_{\mathsf{P}}^2 - L_1^2 \\ (x_{\mathsf{P}} - x_{\mathsf{Q}})^2 + y_{\mathsf{P}}^2 - L_2^2 \end{bmatrix} = \mathbf{0} \tag{3}$$

The velocity- and acceleration-level expressions of Eq. (3) are

$$\dot{\mathbf{\Phi}} = \mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0}, \quad \text{and} \quad \ddot{\mathbf{\Phi}} = \mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0} \tag{4}$$

where $\mathbf{\Phi_q}$ is the Jacobian matrix of the constraints,

$$\mathbf{\Phi_q} = \begin{bmatrix} 2x_\mathsf{P} & 2y_\mathsf{P} & 0 \\ 2\left(x_\mathsf{P} - x_\mathsf{Q}\right) & 2y_\mathsf{P} & 2\left(x_\mathsf{Q} - x_\mathsf{P}\right) \end{bmatrix} \tag{5}$$

whose expression does not explicitly depend on the lengths of links 1 and 2.

For a slider-crank mechanism with $L_1 = L_2 = L$, when the system reaches a singular configuration, e.g., $\theta_1 = \pi/2$, the Jacobian matrix becomes

$$\mathbf{\Phi_q} = \begin{bmatrix} 0 & 2L & 0 \\ 0 & 2L & 0 \end{bmatrix} \tag{6}$$

The Jacobian matrix loses rank instantaneously, and the constraints are compatible with both a slider-crank and a simple-pendulum motion simultaneously. This gives rise to the failure of multibody formulations that are unable to handle rank-deficient Jacobian matrices, and may also cause numerical problems to others that can negotiate this situation.

The extreme-point configuration shown in Fig. 3 takes place when $\theta_1 = 0$, and so $y_\mathsf{P} = 0$. In this case the Jacobian matrix becomes

$$\mathbf{\Phi_q} = \begin{bmatrix} 2x_\mathsf{P} & 0 & 0 \\ 2\left(x_\mathsf{P} - x_\mathsf{Q}\right) & 0 & 2\left(x_\mathsf{Q} - x_\mathsf{P}\right) \end{bmatrix} \tag{7}$$

Together with the system dynamics equations, the acceleration-level constraint equation in (4) shows that, if the mechanism is at rest at $\theta_1 = 0$, and the only applied force is a horizontal action f acting on point Q, the system accelerations are zero. However, small perturbations in the system position will give rise to finite acceleration values. During a forward-dynamics simulation, a lack of accuracy in the solution methods might result in the prediction of an incorrect system motion at these extreme points.

The simulation cases selected for the present benchmark problem intend to test the ability of MBS formulations to correctly handle problems with these difficulties.

2.1.1 Simulation cases

Three simulation cases are considered as shown in Table 1, which details the values of the physical parameters of the system, its initial state, and the externally applied actions.

Table 1 Simulation cases

Case	L_1 (m)	L_2 (m)	m_1 (kg)	m_2 (kg)	m_3 (kg)	$I_{G,1}$ (kgm^2)	$I_{G,2}$ (kgm^2)	$\theta_{1,0}$ (rad)	$\dot{x}_{Q,0}$ (m/s)	f (N)
1	1	1	1	1	0	1/12	1/12	π/4	-4	0
2	3	6	1.5	3	0.25	9/8	9	0	0	$100\sin(\pi t)$
3	1	1	1	1	0	1/12	1/12	π/4	0	$100\sin(\pi t)$

Case 1 corresponds to the slider-crank benchmark problem in [15]. In this case, rods 1 and 2 have the same length ($L_1 = L_2$) and this causes the linkage to pass through a singular configuration when $\theta_1 = \pm\pi/2$. The slider is massless and the externally applied force f in this case is zero during motion.

In case 2, rods 1 and 2 have different lengths and so the linkage motion is not affected by singular configurations. The externally applied force follows now a sinusoidal expression, $f = 100\sin(\pi t)$. The introduction of this force gives rise to numerical difficulties in the solution of the problem. The system ceases to be conservative and its motion is no longer periodic. Moreover, the extreme positions of the motion, reached when both rods are aligned on the x axis and, thus, with the external force f, become challenging from the point of view of numerical simulation. There, depending on the value of the force f and the accuracy of the integration process, the motion can continue following one of two possible branches, namely those that correspond to $\dot{\theta}_1 > 0$ and to $\dot{\theta}_1 < 0$.

Case 3 uses the same physical parameters as case 1, but the externally applied force f follows the sinusoidal expression from case 2. The resulting problem is subjected to both singular configurations and numerical difficulties at the extreme points of the motion.

2.1.2 Variables of interest and metrics

The mechanical system under study has one degree of freedom and a single variable should be enough to keep track of its motion. However, in cases 1 and 3 singular configurations exist and these give rise to the existence of two branches of motion. For this reason, two variables will be monitored and used to evaluate the error in Eq. (2): angle θ_1 between link 1 and the x axis, and the x coordinate of point Q on the slider.

The sampling interval for error evaluation was set to 10 ms; a total simulation length of 10 s was used in all cases. The total errors associated with variables θ_1 and x_Q were aggregated into a single error indicator using Eq. (2) with weights $w_\theta = 1\ \mathrm{rad}^{-2}$ and $w_{x_Q} = 1\ \mathrm{m}^{-2}$, intended to make the final metric dimensionless. Case 1 is a conservative system, and so the total mechanical energy was used there as additional metric.

When using variable step-size integrators, the variables of interest are usually not evaluated exactly at the sampling points. This also happens when constant integration steps are used, but they are not exact dividers of the sampling interval. In these cases, data need to be interpolated; a linear polynomial interpolation has been used here to determine the necessary values.

Two levels of accuracy have been established for the defined cases. In case 1, a high-precision simulation corresponds to a maximum admissible total error $\varepsilon_T = 2 \cdot 10^{-4}$. This is roughly equivalent to the energy criterion set in the IFToMM benchmark, which accepted simulations with maximum energy errors below 0.001 J. The admissibility threshold for low-precision simulation in this case was increased up to $\varepsilon_T = 2 \cdot 10^{-3}$. The same values were used for case 2. Case 3 is more challenging from a computational point of view, and its thresholds were relaxed. The selected thresholds are shown in Table 2.

Table 2 Maximum admissible errors for each simulation case

Case	High-precision ε_Γ	Low-precision ε_Γ
1	$2 \cdot 10^{-4}$	$2 \cdot 10^{-3}$
2	$2 \cdot 10^{-4}$	$2 \cdot 10^{-3}$
3	$2 \cdot 10^{-3}$	$1 \cdot 10^{-2}$

In general, the selection of the threshold depends on factors like the time scale of the dynamics and the difficulty of the problem.

2.2 Solution methods

Several solution methods were used to carry out the forward-dynamics simulation of the benchmark example. Their main characteristics are summarized on Table 3. Unless otherwise specified, MATLAB implementations were used to perform the simulation.

Table 3 Summary of the methods employed to solve the slider-crank benchmark problem

Method	**Coordinates**	**Constraints**	**Integrator**	**Step-size**	**Accuracy**	**Nature**
AL	12 natural, 3D	12 redundant	TR	fixed	O(2)	implicit
ALi3p	12 natural, 3D	12 redundant	TR	fixed	O(2)	implicit
ALi3pvs	12 natural, 3D	12 redundant	TR	variable	O(2)	implicit
mAL	12 Cartesian, 2D	11 independent	ode45	variable	O(4)	explicit
mNS	12 Cartesian, 2D	11 independent	ode45	variable	O(4)	explicit
mAL-ode4	12 Cartesian, 2D	11 independent	ode4	fixed	O(4)	explicit
mNS-ode4	12 Cartesian, 2D	11 independent	ode4	fixed	O(4)	explicit
mAL-ode15s	12 Cartesian, 2D	11 independent	ode15s	variable	variable	implicit

The first method (AL) uses the index-1 augmented Lagrangian algorithm with position and velocity projections presented in [4], integrated with the trapezoidal rule (TR) in fixed-point iteration form. Method ALi3p stands for the index-3 augmented Lagrangian algorithm with projections of velocities and accelerations [6, 7]. Algorithm ALi3pvs is the variable-step version of ALi3p introduced in [8]. Different sets of coordinates could be selected to model the mechanism with these formulations; here we use three-dimensional natural coordinates [16], consisting in the the x, y, and z coordinates of points P and Q and two unit vectors perpendicular to links 1 and 2 respectively. The computational model was built using a 3D multibody software

library developed by the authors; kinematic constraints describing the rigid-body nature of the links and the joints that connect them were subsequently added. The resulting model consists of 12 variables and 12 redundant kinematic constraints. Besides, methods mAL (MATLAB augmented Lagrange) and mNS (MATLAB null-space) were also assessed, in which MATLAB ode45 integration formula is used. The first one uses a stabilized augmented Lagrangian algorithm, while the second adopts a null-space formulation similar to the one in [23]. These methods describe the system with reference point (Cartesian) coordinates, namely the x and y coordinates of the centre of mass of each body (the ground is treated as another body in the assembly) and its orientation angle with respect to the x axis, and impose on them 11 independent constraint equations. In cases 1 and 3, the Jacobian matrix of these constraints loses rank and, as expected, the mNS method was unable to successfully complete the simulation. For the purpose of comparing fixed- and variable-step integrators, methods mAL and mNS were also combined with a fourth order, fixed-step Runge-Kutta integration formula in its classical form, denoted in the text as ode4. The ode15s integration formula, designed for stiff systems, was also used in the evaluation in combination with the mAL formulation.

2.3 Reference solutions

Reference solutions for each case were obtained by convergence of the different methods summarized in Section 2.2. Figures 4 – 6 show the slider displacement x_Q that corresponds to the reference solutions of cases 1 – 3. The way to refine the results depended on the numerical integrator used by each method. For those that used constant integration step-sizes, the step was reduced to increase the precision. In the case of methods with variable-step integration, the tolerances were made more stringent. The convergence towards a reference solution in all cases was additionally verified integrating the system dynamics expressed as a single unconstrained degree-of-freedom equation of motion with MATLAB's ode15i method. This solution, upon convergence, was used to provide the values of $y_j^{\mathrm{ref}}(t_i)$ required by Eq. (1).

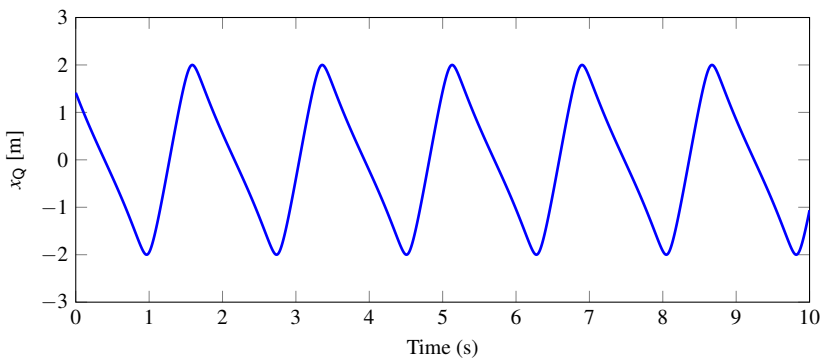

Fig. 4 Reference solution: displacement x_Q of the slider in case 1.

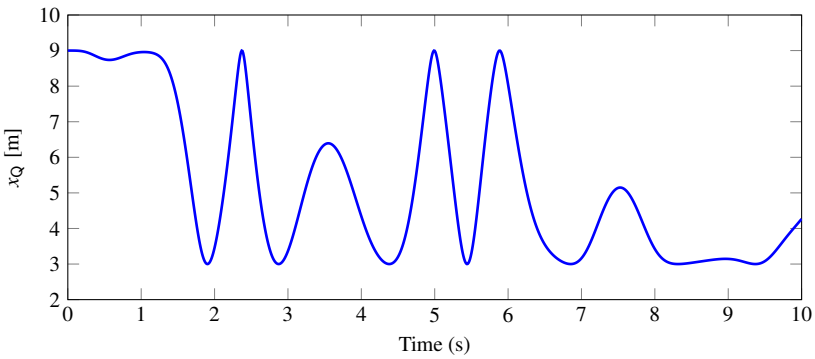

Fig. 5 Reference solution: displacement x_Q of the slider in case 2.

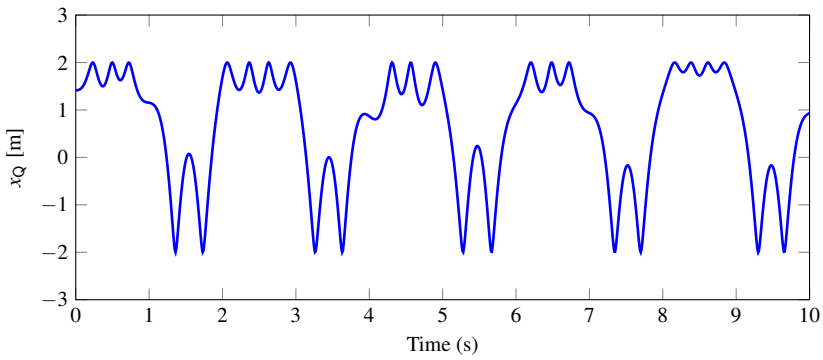

Fig. 6 Reference solution: displacement x_Q of the slider in case 3.

Upon convergence, the differences in the monitored variables across the solutions delivered by the methods remained below $3 \cdot 10^{-5}$ m for x_Q and $2.5 \cdot 10^{-5}$ rad for θ_1 in case 1, $7 \cdot 10^{-4}$ m for x_Q and $8 \cdot 10^{-4}$ rad for θ_1 in case 2, and $4 \cdot 10^{-3}$ m for x_Q and $6 \cdot 10^{-3}$ rad for θ_1 in case 3, at every sampling point. Additionally, for case 1, the error in the mechanical energy of the reference solution was lower than 10^{-6} J.

3 RESULTS

The benchmark scenarios defined in Section 2 were used to evaluate and compare the performance of the simulation solutions in Table 3. The computations were performed on an Intel Core i7-7700HQ at 2.80 GHz with 16 GB of RAM, running Windows 64-bit and MATLAB R2020b. For the purposes of this study, and considering the particular characteristics of the simulation environment, besides the precision thresholds in Table 2, a maximum admissible elapsed time in computations of 2000 s was established for all simulations.

In the case of the constant-step AL, ALi3p, mAL-ode4, and mNS-ode4 methods, the integration step-size h was found to be the parameter that had the greatest impact on the efficiency and accuracy of the simulation. For the trapezoidal rule in ALi3pvs, this role corresponded to the maximum admissible increment of the variables between iterations in each step, $\varphi_{\max}$, selected as stopping criterion for the convergence of the integration formula. In the case of the mAL and mNS methods, integrated with ode45, the most critical factors were the absolute and relative tolerances of the integrator, φ_{abs} and φ_{rel}. This was also the case with ode15s. It is worth mentioning that tuning the ALi3pvs method to each particular problem introduced an additional complexity, as its performance depended as well on parameters like the maximum number of iterations γ allowed during the Newton-Raphson iteration and the upper and lower limits $h_{\max}$ and $h_{\min}$ for the integration step-size. Tuning these parameters was not a straightforward process, because the impact of a given selection on the simulation performance is highly nonlinear. Moreover, because the algorithm adjusts the step-size based on its previously used value, the initially used step-size h_0 must also be considered a parameter of the simulation. It is difficult to provide general recommendations on the selection of the these parameters. In general, it seems advisable to decrease the limit of admissible iterations per step γ and regulate the error of the simulation by tuning the convergence criterion $\varphi_{\max}$. When using ode45-methods, φ_{abs} and φ_{rel} can be selected independently from each other introducing, also in this case, an additional complexity in the tuning process. Regarding the mAL-ode15s method, adjusting its parameters and tolerances was even more complicated, as its convergence trends were not as clear as with ode45.

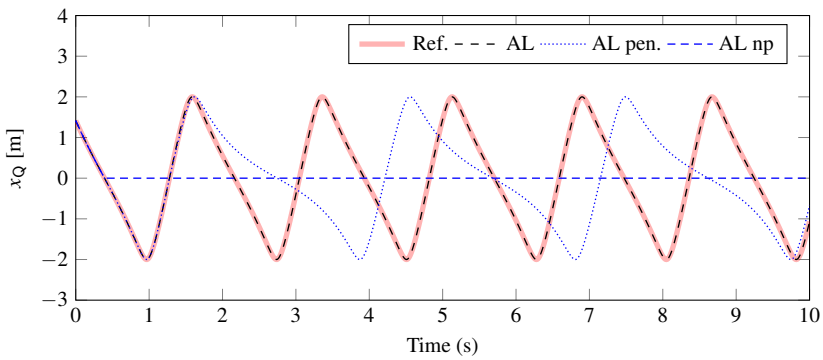

Fig. 7 Case 1: effect on displacement x_Q of the slider of the parameter selection of augmented Lagrangian formulations.

All the augmented Lagrangian formulations are, in principle, able to deal with singular configurations, which is required by cases 1 and 3. Figure 7, however, confirms that the parameters of the augmented Lagrangian formulations also need to be tuned appropriately to arrive at correct results. The penalty factor α required by these methods has an admissible range of validity. For a constant step-size $h = 1$ ms, $10^7 \leq \alpha \leq 10^{15}$ for the ALi3p method, and $1 \leq \alpha \leq 10^6$ for the AL formulation.

Within this range, both methods correctly predict the system motion, delivering plot 'AL' in Fig. 7. An incorrect tuning of α leads to an invalid simulation, as is the case of the 'AL pen.' plot, in which the penalty of the AL method was set to $\alpha = 10^7$. Moreover, it is noteworthy that, even though these formulations are able to handle rank-deficient Jacobian matrices, branch changes can affect the predicted motion in some cases, even when the system velocity is not zero. Plot 'AL np' in Fig. 7 corresponds to the case in which position and velocity projections have been disabled in the AL formulation; the resulting method is theoretically able to perform the simulation of the system motion as a slider-crank, but it switches to a pendulum motion around $t = 0.45$ s.

Table 4 Best results in the solution of case 1 (low precision)

Method	Elapsed (s)	ε_T	Parameters
AL	2.70	$1.60 \cdot 10^{-3}$	$h = 2.5$ ms, $\alpha = 10^6$
ALi3p	3.13	$1.67 \cdot 10^{-3}$	$h = 3.5$ ms, $\alpha = 10^{12}$
ALi3pvs	2.07	$1.51 \cdot 10^{-3}$	$h_{\min} = 0.5$ ms, $h_{\max} = 4.5$ ms, $\alpha = 10^{12}$, $\gamma = 1$, $\varphi_{\max} = 10^{-5}$
mAL	2.90	$1.95 \cdot 10^{-3}$	$\alpha = 10^{10}$, $\varphi_{\text{rel}} = 5 \cdot 10^{-2}$, $\varphi_{\text{abs}} = 10^{-1}$
mNS	Failed		
mAL-ode4	2.66	$1.69 \cdot 10^{-3}$	$h = 2$ ms, $\alpha = 10^{10}$
mNS-ode4	Failed		
mAL-ode15s	9.96	$1.06 \cdot 10^{-3}$	$\alpha = 10^7$, $\varphi_{\text{rel}} = 1 \cdot 10^{-7}$, $\varphi_{\text{abs}} = 10^{-11}$

Table 5 Best results in the solution of case 1 (high precision)

Method	Elapsed (s)	ε_T	Parameters
AL	6.99	$1.38 \cdot 10^{-4}$	$h = 1$ ms, $\alpha = 10^6$
ALi3p	6.39	$1.81 \cdot 10^{-4}$	$h = 1$ ms, $\alpha = 10^{12}$
ALi3pvs	4.69	$1.20 \cdot 10^{-4}$	$h_{\min} = 0.5$ ms, $h_{\max} = 3$ ms, $\alpha = 10^{12}$, $\gamma = 1$, $\varphi_{\max} = 5 \cdot 10^{-7}$
mAL	3.45	$1.35 \cdot 10^{-4}$	$\alpha = 10^{10}$, $\varphi_{\text{rel}} = 5 \cdot 10^{-3}$, $\varphi_{\text{abs}} = 2 \cdot 10^{-2}$
mNS	Failed		
mAL-ode4	3.20	$1.02 \cdot 10^{-4}$	$h = 1.75$ ms, $\alpha = 10^{10}$
mNS-ode4	Failed		
mAL-ode15s	20.17	$1.05 \cdot 10^{-4}$	$\alpha = 10^7$, $\varphi_{\text{rel}} = 1 \cdot 10^{-8}$, $\varphi_{\text{abs}} = 10^{-11}$

The benchmarking results obtained in the simulation of case 1 are shown in Tables 4 and 5, for the low and high precision thresholds specified in Table 2, respec-

tively. Numerical experiments confirmed that methods mNS and mNS-ode4 were unable to deal with the singular configurations of the mechanism, which resulted in the failure of the simulation. The augmented Lagrangian methods, on the other hand, completed the simulation successfully. The elapsed times reported correspond to the average of five different simulation runs. For the low-precision case, all the successful methods delivered comparable results in terms of efficiency. As expected, times increased in the high-precision scenario, particularly for the methods that modelled the system with natural coordinates and used the trapezoidal rule as integration routine. The use of a variable step-size in ALi3pvs resulted in an improvement of the efficiency with respect to its fixed-step counterpart, especially in the high-precision case.

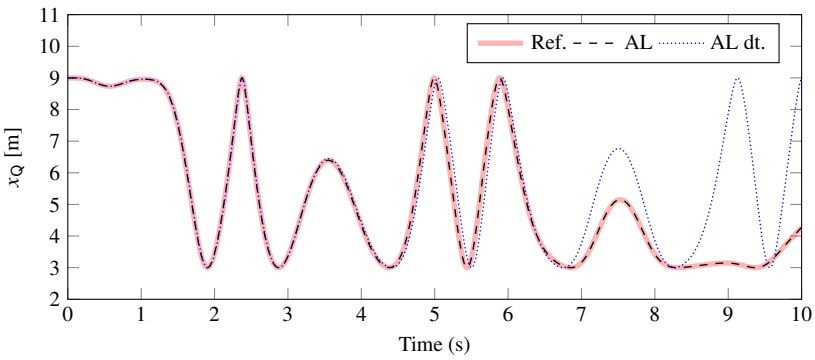

Fig. 8 Case 2: effect on displacement x_Q of the slider of the parameter selection of augmented Lagrangian methods.

Case 2 is not subjected to singular configurations, but its simulation is challenging for a different reason. Some methods, like the AL and ALi3p used in this study, introduce numerical dissipation in the solution of the dynamics equations [10]. This dissipation may slow down slightly the predicted motion of the system. The external, time-dependent force f that acts on the slider is then applied to the mechanism at a configuration that differs from that of the reference solution at each particular time. In some cases, this eventually leads to the departure of the system motion from theoretically correct values, as shown in Fig. 8. The augmented Lagrangian formulations deliver results that match the reference solution provided that small enough integration step-sizes are used ('AL'), but might follow a different path otherwise, as can be seen for plot 'AL dt', which corresponds to the AL formulation integrated with a step-size $h = 5$ ms. This issue imposes an important limitation on the efficiency of these formulations.

Tables 6 and 7 summarize the results obtained with the different methods in the solution of case 2, for low and high precision, respectively. All the methods were able to meet the low-precision threshold in the simulation of this case, although augmented Lagrangian algorithms with projections required more time to complete the integration, as a consequence of the energy dissipation mentioned in the previous paragraph.

Table 6 Best results in the solution of case 2 (low precision)

Method	Elapsed (s)	ε_T	Parameters
AL	1345.03	$1.79 \cdot 10^{-3}$	$h = 0.004$ ms, $\alpha = 10^6$
ALi3p	12.18	$1.19 \cdot 10^{-3}$	$h = 0.5$ ms, $\alpha = 10^{12}$
ALi3pvs	7.85	$1.77 \cdot 10^{-3}$	$h_{\min} = 0.1$ ms, $h_{\max} = 2$ ms, $\alpha = 10^{12}$, $\gamma = 1$, $\varphi_{\max} = 10^{-7}$
mAL	3.44	$1.93 \cdot 10^{-3}$	$\alpha = 10^7$, $\varphi_{\text{rel}} = 10^{-3}$, $\varphi_{\text{abs}} = 5 \cdot 10^{-2}$
mNS	0.24	$1.69 \cdot 10^{-3}$	$\varphi_{\text{rel}} = 10^{-5}$, $\varphi_{\text{abs}} = 10^{-9}$
mAL-ode4	3.65	$1.61 \cdot 10^{-3}$	$h = 1.5$ ms, $\alpha = 10^7$
mNS-ode4	0.58	$1.64 \cdot 10^{-3}$	$h = 7.5$ ms
mAL-ode15s	26.77	$1.28 \cdot 10^{-3}$	$\alpha = 10^7$, $\varphi_{\text{rel}} = 10^{-10}$, $\varphi_{\text{abs}} = 10^{-7}$

Table 7 Best results in the solution of case 2 (high precision)

Method	Elapsed (s)	ε_T	Parameters
AL	Failed		
ALi3p	53.28	$6.61 \cdot 10^{-5}$	$h = 0.1$ ms, $\alpha = 10^{12}$
ALi3pvs	15.87	$1.09 \cdot 10^{-4}$	$h_{\min} = 0.1$ ms, $h_{\max} = 1$ ms, $\alpha = 10^{12}$, $\gamma = 1$, $\varphi_{\max} = 10^{-8}$
mAL	5.98	$1.18 \cdot 10^{-4}$	$\alpha = 10^7$, $\varphi_{\text{rel}} = 10^{-4}$, $\varphi_{\text{abs}} = 10^{-5}$
mNS	0.31	$1.97 \cdot 10^{-4}$	$\varphi_{\text{rel}} = 10^{-7}$, $\varphi_{\text{abs}} = 10^{-6}$
mAL-ode4	7.07	$1.07 \cdot 10^{-4}$	$h = 0.5$ ms, $\alpha = 10^7$
mNS-ode4	0.97	$1.64 \cdot 10^{-4}$	$h = 4.5$ ms
mAL-ode15s	34.32	$2.06 \cdot 10^{-4}$	$\alpha = 10^7$, $\varphi_{\text{rel}} = 10^{-11}$, $\varphi_{\text{abs}} = 10^{-7}$

This effect was particularly noticeable for the AL algorithm, which required a step size of $h = 0.004$ s to achieve correct results. The issue was still more critical in the high-precision run, which the AL method was unable to complete below the 2000 s limit.

Two interesting facts can be remarked. The first is that the null-space formulations delivered the most efficient simulation execution. Case 2 does not feature singular configurations, and so these methods can be used to effectively solve the system dynamics. The second is that the use of variable-step integrators resulted in significant time savings with respect to their fixed-step counterparts. The computational improvement became more relevant as the precision required from the execution increased. This was clear in the case of the ALi3p and ALi3pvs methods, but could also be observed for mAL and mNS formulations. Figure 9 shows the times elapsed by these methods in the solution of Case 2 for different precision levels, and confirms the computational advantage of variable-step integration for higher precision requirements.

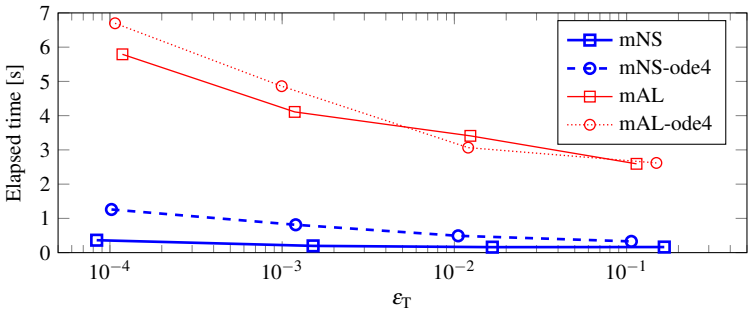

Fig. 9 Elapsed time in the solution of case 2 as a function of the precision ε_T delivered by mAL and mNS methods.

Table 8 Best results in the solution of case 3 (low precision)

Method	Elapsed (s)	ε_T	Parameters
AL	Failed		
ALi3p	92.74	$9.36 \cdot 10^{-3}$	$h = 0.1$ ms, $\alpha = 10^{15}$
ALi3pvs	48.84	$1.75 \cdot 10^{-2}$	$h_{\min} = 0.05$ ms, $h_{\max} = 0.1$ ms, $\alpha = 10^{15}$, $\gamma = 1$, $\varphi_{\max} = 10^{-8}$
mAL	5.83	$6.33 \cdot 10^{-3}$	$\alpha = 10^{10}$, $\varphi_{\text{rel}} = 10^{-3}$, $\varphi_{\text{abs}} = 10^{-5}$
mNS	Failed		
mAL-ode4	6.76	$5.26 \cdot 10^{-3}$	$h = 0.8$ ms, $\alpha = 10^{10}$
mNS-ode4	Failed		
mAL-ode15s	11.04	$4.82 \cdot 10^{-3}$	$\alpha = 10^{7}$, $\varphi_{\text{rel}} = 10^{-7}$, $\varphi_{\text{abs}} = 10^{-8}$

Finally, the simulation results that correspond to case 3 are summarized in Tables 8 and 9. The mNS formulations failed again in this case, due to the existence of singular configurations. The AL algorithm was unable to meet the precision thresholds in less than 2000 s. Regarding the ALi3p method, it required an increased value of the penalty factor ($\alpha = 10^{15}$) to deliver correct results in the low-precision run. The ALi3pvs delivered again a significant reduction of the elapsed time in computations by means of time-step adjustments, even though its step-size was set to be always smaller than that of the fixed-step ALi3p. This is explained by a reduction in the number of iterations required by the method to attain convergence at each integration step. Decreasing the step size with ALi3p, conversely, did not result in performance improvements. The mAL and mAL-ode4 delivered the most efficient results. Regarding the high-precision threshold, we did not find any parameter combination by which ALi3p and ALi3pvs were able to meet the simulation requirements. It must be pointed out that these formulations are known to suffer from numerical issues for very small integration steps [6]. Moreover, tuning the parameters of these methods, especially the variable-step ALi3pvs, is not straightforward, and it was not possible to determine a standardized protocol to adjust them.

Table 9 Best results in the solution of case 3 (high precision)

Method	Elapsed (s)	ε_{T}	Parameters
AL	Failed		
ALi3p	Failed		
ALi3pvs	Failed		
mAL	12.88	$1.42 \cdot 10^{-3}$	$\alpha = 10^{10}$, $\varphi_{\mathrm{rel}} = 5 \cdot 10^{-6}$, $\varphi_{\mathrm{abs}} = 10^{-5}$
mNS	Failed		
mAL-ode4	11.23	$9.78 \cdot 10^{-4}$	$h = 0.5$ ms, $\alpha = 10^{10}$
mNS-ode4	Failed		
mAL-ode15s	26.30	$1.53 \cdot 10^{-3}$	$\alpha = 10^{7}$, $\varphi_{\mathrm{rel}} = 10^{-8}$, $\varphi_{\mathrm{abs}} = 5 \cdot 10^{-9}$

The presented simulation results confirm the ability of the proposed benchmark problem to help discriminate between several solution methods for MBS dynamics under different precision requirements and execution conditions. This can be useful to assess the capabilities of newly proposed formalisms and software implementations in the simulation of mechanical systems with singular configurations and time-dependent inputs.

4 Conclusions

In this work, a benchmark example for the evaluation of multibody dynamics formulations with kinematic constraints has been introduced. The selected example consists of three variations of a planar slider-crank linkage, with different physical parameters and applied external forces. The combination of these forces with the existence of singular configurations in the mechanism resulted in a challenging problem for the majority of the tested solution methods. Indeed, the simulation software needs to be able to accurately describe the system motion both in the proximity of the singularities and at the extreme points of the motion, which becomes more complicated when an external force is acting on the mechanism.

The slider-crank linkage example also served to highlight relevant aspects of the definition of benchmark problems, such as the necessity of an available reference solution and suitable metrics to compare the performance of different solution methods.

The three proposed cases of this benchmark problem were used to assess the ability of different formulations in combination with fixed- and variable-step integration methods for multibody system dynamics. Results showed that the performance of each approach depends on the precision requirements and the characteristics of the problem being solved and confirmed the usefulness of the proposed example for benchmarking MBS dynamics solution methods.

Acknowledgements F. González acknowledges the support of the Ministry of Economy of Spain through the Ramón y Cajal programme, contract RYC-2016-20222.

References

1. Bauchau, O.A.: Flexible Multibody Dynamics. Springer, Dordrecht, The Netherlands (2011). DOI 10.1007/978-94-007-0335-3. ISBN:978-94-007-0334-6
2. Bauchau, O.A., Betsch, P., Cardona, A., Gerstmayr, J., Jonker, B., Masarati, P., Sonneville, V.: Validation of flexible multibody dynamics beam formulations using benchmark problems. Multibody System Dynamics **37**(1), 29–48 (2016). DOI 10.1007/s11044-016-9514-y
3. Bayo, E., Avello, A.: Singularity-free augmented Lagrangian algorithms for constrained multibody dynamics. Nonlinear Dynamics **5**(2), 209–231 (1994). DOI 10.1007/bf00045677
4. Bayo, E., Ledesma, R.: Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. Nonlinear Dynamics **9**(1-2), 113–130 (1996). DOI 10.1007/bf01833296
5. Bezin, Y., Pålsson, B.A.: Multibody simulation benchmark for dynamic vehicle-track interaction in switches and crossings: modelling description and simulation tasks. Vehicle System Dynamics pp. 1–16 (2021). DOI 10.1080/00423114.2021.1942079
6. Cuadrado, J., Cardenal, J., Morer, P., Bayo, E.: Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments. Multibody System Dynamics **4**(1), 55–73 (2000). DOI 10.1023/a:1009824327480
7. Dopico, D., González, F., Cuadrado, J., Kövecses, J.: Determination of holonomic and nonholonomic constraint reactions in an index-3 augmented Lagrangian formulation with velocity and acceleration projections. Journal of Computational and Nonlinear Dynamics **9**(4) (2014). DOI 10.1115/1.4027671
8. Dopico, D., Sanjurjo, E., Cuadrado, J., Luaces., A.: A variable time-step and variable penalty method for the index-3 augmented Lagrangian formulation with velocity and acceleration projections. In: Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics. Prague, Czech Republic (2017)
9. Featherstone, R.: Robot dynamics algorithms. Springer, New York (NY), USA (1987). DOI 10.1007/978-0-387-74315-8. ISBN:978-1-4757-6437-6
10. García Orden, J.C., Dopico, D.: On the stabilizing properties of energy-momentum integrators and coordinate projections for constrained mechanical systems. In: J.C. García Orden, J.M. Goicolea, J. Cuadrado (eds.) Computational Methods in Applied Sciences, vol. 4, pp. 49–67. Springer, Dordrecht (2007). DOI 10.1007/978-1-4020-5684-0_3
11. González, F., Dopico, D., Pastorino, R., Cuadrado, J.: Behaviour of augmented Lagrangian and Hamiltonian methods for multibody dynamics in the proximity of singular configurations. Nonlinear Dynamics **85**(3), 1491–1508 (2016). DOI 10.1007/s11071-016-2774-5
12. González, M., Dopico, D., Lugrís, U., Cuadrado, J.: A benchmarking system for MBS simulation software: Problem standardization and performance measurement. Multibody System Dynamics **16**(2), 179–190 (2006). DOI 10.1007/s11044-006-9020-8
13. González, M., González, F., Dopico, D., Luaces, A.: On the effect of linear algebra implementations in real-time multibody system dynamics. Computational Mechanics **41**(4), 607–615 (2008). DOI 10.1007/s00466-007-0218-2
14. González, M., González, F., Luaces, A., Cuadrado, J.: A collaborative benchmarking framework for multibody system dynamics. Engineering with Computers **26**(1), 1–9 (2010). DOI 10.1007/s00366-009-0139-0
15. IFToMM Technical Committee for Multibody Dynamics: Library of computational benchmark problems (2021). URL https://www.iftomm-multibody.org/benchmark/. URL:https://www.iftomm-multibody.org/benchmark/
16. García de Jalón, J., Bayo, E.: Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge. Springer-Verlag, New York (NY), USA (1994). ISBN:3-540-94096-0
17. García de Jalón, J., Gutiérrez-López, M.D.: Multibody dynamics with redundant constraints and singular mass matrix: existence, uniqueness, and determination of solutions for accelerations and constraint forces. Multibody System Dynamics **30**(3), 311–341 (2013). DOI 10.1007/s11044-013-9358-7
18. Marques, F., Souto, A.P., Flores, P.: On the constraints violation in forward dynamics of multibody systems. Multibody System Dynamics **39**(4), 385–419 (2016). DOI 10.1007/s11044-016-9530-y
19. Masoudi, R., Flores, P., McPhee, J.: Benchmark problems for contact dynamics in multibody systems. In: The 3rd International Conference on Multibody System Dynamics (IMSD). Busan, South Korea (2014)
20. Sanjurjo, E., Naya, M.Á., Blanco-Claraco, J.L., Torres-Moreno, J.L., Giménez-Fernández, A.: Accuracy and efficiency comparison of various nonlinear Kalman filters applied to multibody models. Nonlinear Dynamics **88**(3), 1935–1951 (2017). DOI 10.1007/s11071-017-3354-z

21. Schiehlen, W. (ed.): Advanced Multibody System Dynamics. Springer Netherlands (1993). DOI 10.1007/978-94-017-0625-4
22. Seabra Pereira, M.F.O., Ambrósio, J.A.C. (eds.): Computer-Aided Analysis of Rigid and Flexible Mechanical Systems. Springer Netherlands (1994). DOI 10.1007/978-94-011-1166-9
23. Shabana, A.A.: Computational Dynamics, second edn. Wiley, Hoboken, NJ, USA (2001). ISBN:978-0-471-05326-2
24. Torres-Moreno, J.L., Blanco-Claraco, J.L., López-Martínez, J., Giménez-Fernández, A.: A comparison of algorithms for sparse matrix factoring and variable reordering aimed at real-time multibody dynamic simulation. In: Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics. Zagreb, Croatia (2013)
25. Zar, A., González, F., Rodríguez, B., Luaces, A., Naya, M.A., Cuadrado, J.: Benchmark problems for co-simulation methods. In: COSIM 2021: International Symposium on Co-Simulation and Solver Coupling in Dynamics. Ferrol, Spain (2021)