Semiexplicit numerical integration by splitting with application to dynamic multibody problems with contacts

Klas Modin and Dag Fritzson

SKF Engineering Research Centre
MDC, RKs–2
SE–415 50 Göteborg, Sweden
E–mail: Klas.Modin@na.lu.se, Dag.Fritzson@skf.com

Claus Führer

Centre for Mathematical Sciences
Box 118, SE–221 00 Lund, Sweden
E–mail: claus@maths.lth.se

December 4, 2007

Numerical integration is considered for second order differential equations on the form

$$\ddot{q} = A(q, \dot{q}, t) + B(q, \dot{q}, t)\,,$$

where A is significantly more expensive to evaluate than B, and B is stiff (highly oscillatory) in comparison with A. Examples of such problem are multibody problem with contact forces acting between bodies, and constraints formulated as penalty forces.

Based on the splitting $A + B$ of the acceleration field, a numerical integration algorithm, which is explicit in the A–part and implicit in the B–part, is suggested. Consistency and linear stability analysis of the proposed method is carried out.

Numerical examples with the proposed method is carried out for two simple test problems, and for a complex multibody model of a rotating ball bearing. Comparison with conventional implicit methods is given for each example. The results indicate that the proposed method is more efficient, in terms of number of evaluations of A, at the same accuracy level.

Contents

1. Introduction

We are interested in the numerical time integration of dynamic multibody systems. In particular, problems where bodies frequently come in contact with each other, which is modelled by complex force laws. The application we have in mind is simulation of rolling bearings, where contacts between bodies are present. Our aim is to design a numerical integrator that is particularly efficient for such problems.

In Section 2, a characterization of the governing equations is given. The objective is twofold: (i) to pin-point where the computational cost is high; (ii) to estimate typical frequencies in the solution. The specific character of the governing equations is then used as a basis in Section 3 in order to design a more efficient integrator. Further, in Section 4 the proposed integrator is analyzed in terms of consistency and linear stability. In Section 5 we discuss issues concerned with adaptive time-stepping. Lastly, in Section 6, we give numerical test examples, both for simple test problems and for a fully complex rolling bearing problem.

2. Characterization of the governing equations

In this section we give an overview of the formulation, and a characterization, of the governing differential equations that are to be numerically integrated. There are, of course, a number of choices on how to formulate the equations of motion for multibody systems. Primarily, the so called "floating frame of reference" is what we have in mind. In particular, the formulation used in the multibody simulation software BEAST, which is a tool for detailed transient analysis of rolling bearings and other machine

elements, developed and maintained by SKF (www.skf.com). For a full specification of the governing equations, see [Nak06]. In this paper we consider only issues that are essential for the proposed integrator.

Let $\mathcal{Q} = \mathbb{R}^d$ be the configuration space of the multibody system. Further, let $\boldsymbol{q} \in \mathcal{Q}$ denote the position coordinates. For rigid systems these are the centre of mass and the orientation of each body, relative to a fixed global coordinate system in Euclidean 3–space. In the case of elasiticity, generalized coordinates describing the deflection field of each body are also included in $\boldsymbol{q}$. The governing equations are on the form

$$M(\boldsymbol{q},\dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} = F(\boldsymbol{q},\dot{\boldsymbol{q}},t)\,, \qquad \boldsymbol{q}(0) = \boldsymbol{q}_0\,, \qquad \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0\,, \tag{1}$$

where the mass matrix $M(\boldsymbol{q},\dot{\boldsymbol{q}})$ is a symmetric positive definite $d \times d$–matrix, and the force field $F(\boldsymbol{q},\dot{\boldsymbol{q}},t)$ is a vector valued map corresponding to the forces acting on the system. Time is denoted t. Notice that (1) is a second order ordinary differential equation (ODE).

Constraints are taken into account by penalty force laws. Thus, we do not utilize the standard Lagrangian multiplier formulation, which is typically used for constrained mechanical systems. Notice that the penalty formulation implies that (1) is highly stiff. Thus, an implicit numerical integrator must be used.

Remark. The integrator suggested in this paper could easily be extended to governing equations formulated as a differential algebraic system using Lagrangian multipliers.

2.1. Computational costs

Solving (1) numerically with a standard ODE solver basically involves the following computations in order to evaluate the vector field:

- Compute the inverse of the mass matrix. This is a cheap operation, as the mass matrix is block diagonal (one block per body).
- Compute the constraint forces, which is also cheap.
- Compute "simple" non-stiff forces such as gravity and Coriolis forces.
- Compute contact forces. This task is heavily dominating the computational cost. Evaluation of each contact involves searching for intersecting surfaces. In the case of rolling bearings, the surface geometries of the bodies are highly complex. Further, for very detailed contact models (as in BEAST), tribological issues, such as oil film thickness in the contact, are also taken into account. See [SF01] for details on advanced contact modelling.

In order to be able to separate the computationally intensive force evaluations from less costly evaluations, we rewrite (1) as

$$M(\boldsymbol{q},\dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} = F^A(\boldsymbol{q},\dot{\boldsymbol{q}},t) + F^B(\boldsymbol{q},\dot{\boldsymbol{q}},t)\,, \tag{2}$$

where F^A are the contact forces plus "simple" non-stiff forces (from now on we refer to F^A as contact forces), and F^B are the constraint penalty forces. Multiplying from the left with the inverse of the mass matrix we get a second order ODE written on standard form

$$\ddot{\boldsymbol{q}} = A(\boldsymbol{q},\dot{\boldsymbol{q}},t) + B(\boldsymbol{q},\dot{\boldsymbol{q}},t) \quad \text{or shorter} \quad \ddot{\boldsymbol{q}} = C(\boldsymbol{q},\dot{\boldsymbol{q}},t)\,. \tag{3}$$

The maps A and B correspond to accelerations due to contact forces and constraint forces respectively. $C = A + B$ is the total acceleration field. The field A is much more expensive to compute than B. This "splitting formulation" of the governing equations will be utilized by the integrator algorithm described in Section 3.

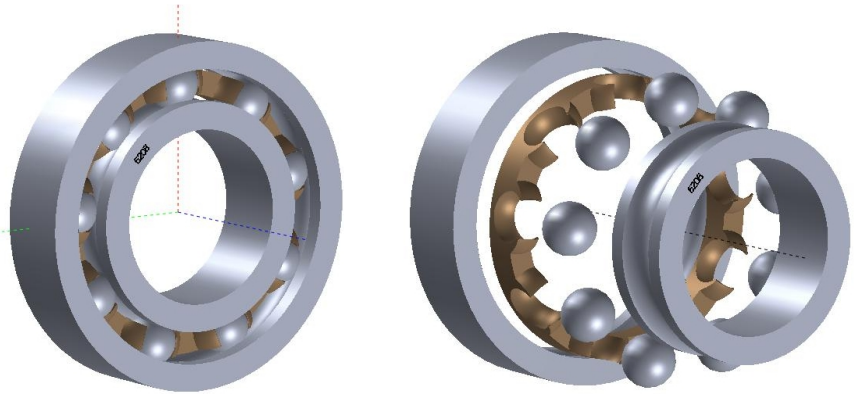

Figure 1: A common ball bearing. To the right in exploded view. The bodies in the model are: (1) an outer ring; (2) a cage; (3) nine balls; (4) an inner ring. All elements are steel, except the cage which is plastic.

Notice that (3) is a non-autonomous differential equation, i.e., the right hand side depends explicitly on time t. From now on we rewrite the governing equations as an autonomous first order system, evolving on the extended phase space $\mathscr{P} = T\mathscr{Q} \times \mathbb{R} = \mathbb{R}^{2d+1}$, equiped with coordinates $\boldsymbol{z} = (\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$. That is, we write (3) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ t \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{q}} \\ A(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) + B(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \\ 1 \end{pmatrix} \quad \text{or shorter} \quad \dot{\boldsymbol{z}} = X(\boldsymbol{z})\,. \tag{4}$$

The phase flow corresponding to (4) is denoted φ^h. That is, φ^h is a map $\mathscr{P} \to \mathscr{P}$, depending on the time step length h, such that $\varphi^h(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t) = (\boldsymbol{q}(t+h), \dot{\boldsymbol{q}}(t+h), t+h)$.

2.2. Estimated frequencies

Our next objective is to estimate the frequencies, or time scales, in the system due to A and B. In order to do so we consider a simplified model of a ball bearing, see Figure 1.

We begin with typical frequencies due to contact forces, i.e., due to A. The stiffness in a steel–steel contact (e.g. between a ball and the outer ring) is typically about $k = 10^8\,\mathrm{N/m}$. The damping and friction forces are small, so we neglect them in this simple analysis. The mass of a ball is about $m = 0.01\,\mathrm{kg}$. Thus, a typical translational frequency is about $(\sqrt{k/m})/(2\pi) \approx 1.6 \cdot 10^4\,\mathrm{Hz}$. The smallest moment of inertia of the outer ring is about $J = 3 \cdot 10^{-5}\,\mathrm{kg \cdot m^2}$ and its radius about $3 \cdot 10^{-2}\,\mathrm{m}$. This gives a typical rotational frequency of about $5 \cdot 10^4\,\mathrm{Hz}$. From experience with the BEAST software, using a standard implicit ODE solver, it is known that the time step length, for bearings like in Figure 1, typically is about $10^{-6}\,\mathrm{s}$. This time step is small enough to fully capture the dynamics of A.

In order to get an accurate result, the stiffness (and damping) of the constraint forces must be set much higher than the stiffness of the "real physical" forces. From our point of view, it means that the time scales of B must be much smaller than those of A. For the bearing case described it is suitable to choose the stiffness and damping of the penalty forces so that the frequencies of B are about $10^7\,\mathrm{Hz}$. As the time step $10^{-6}\,\mathrm{s}$ is not small enough to resolve such frequencies, it is essential to use an implicit integrator.

3. Integrator algorithm

From the previous section it is clear that the governing equations are stiff due to constraint forces, so for the time step lengths we have in mind an implicit integrator is needed in order to achieve stability. From an efficiency point of view, an implicit integrator is, per time step, more expensive than an explicit dito, because at each step a non–linear root finding problem needs to be solved iteratively. In particular, this implies that A and B are evaluated several times per time step. Furthermore, the Jacobian needs to be computed for an implicit method (which is severely expensive for A in our case), as it is used by the root finding algorithm (typically some variant of Newton's method).

The frequency analysis of the previous section indicates that the dynamics of A can be resolved accuratelly with an explicit integrator for the time step lengths we have in mind. However, it is not so for B. One possibility is of course to choose a much smaller time step, but that would be inefficient, as to many time steps are needed.

Our approach is to treat A explicitly and B implicitly. The notion is that B is "responsible" for the stiff character of the system, so in order to have stability for long time steps it is enough if only this part is handled implicitly. The idea is that the number of evaluations of A should be on par with that of explicit methods, whereas B can be evaluted more frequently without any significant increase in the computational cost.

An obvious possibility is to use so called splitting methods (see [MQ02]). That is, to consider an explicit method Φ_h^A for the equation $\ddot{\boldsymbol{q}} = A(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ and an implicit method Φ_h^B for $\ddot{\boldsymbol{q}} = B(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$, and then utilized a composition of the two methods, e.g. $\Phi_h^A \circ \Phi_h^B$ or $\Phi_{h/2}^B \circ \Phi_h^A \circ \Phi_{h/2}^B$. However, this approach is not so good in our case. Indeed, as Φ_h^A does not take constraint forces into account, the solution would at each step drift away a little from the constraint manifold, and then be "forced back" towards it by Φ_h^B. As Φ_h^B does not exactly project onto the constraint manifold, high frequency $\mathcal{O}(h)$ oscillations would thus appear in the solution.

We suggest the following discretization of the governing equations (4)

$$\begin{aligned} \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \frac{h^2}{2}\Big(A_{n+\alpha} + B_{n+\beta}\Big) \\ \dot{\boldsymbol{q}}_{n+1} &= \dot{\boldsymbol{q}}_n + h\Big(A_{n+\alpha} + B_{n+\beta}\Big) \\ t_{n+1} &= t_n + h\,, \end{aligned} \tag{5}$$

where $\alpha, \beta \in [0, 1]$ are method parameters and

$$\begin{aligned} A_{n+\alpha} &= A(\boldsymbol{q}_n + h\alpha\dot{\boldsymbol{q}}_n, \dot{\boldsymbol{q}}_n, t_n + \alpha h) \\ B_{n+\beta} &= B\big((1-\beta)\boldsymbol{z}_n + \beta\boldsymbol{z}_{n+1}\big)\,, \quad \boldsymbol{z}_n = (\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n)\,. \end{aligned}$$

Notice that $A_{n+\alpha}$ is independent of $\boldsymbol{q}_{n+1}$ and $\dot{\boldsymbol{q}}_{n+1}$ for all α. Thus, it only needs to be evaluated once per time step, and the Jacobian of A is not needed. This is what we mean by "treating A explicitly".

The method (5) is a blend of the Störmer–Verlet method, the explicit Euler method, and the implicit midpoint rule. The correspondence is as follows.

Condition	**Method**
$B = 0$, $\alpha = 1/2$, and A independent of $\dot{\boldsymbol{q}}$	Störmer–Verlet
$B = 0$ and A independent of $\boldsymbol{q}$	explicit Euler (for the $\dot{\boldsymbol{q}}$ part)
$A = 0$ and $\beta = 1/2$	implicit midpoint rule

4. Consistency and linear stability analysis

In this section we analyse the order of consistency and the linear stability of the proposed method (5). We begin with the consistency analysis. Next, the stability analysis, which is split into two parts. First the explicit component of the algorithm is analysed separately, i.e. the case $B = 0$. Based on these results, stability of the full algorithm, with constraint forces present, is thereafter analysed.

4.1. Order of consistency

Let $\Phi_h : \mathscr{P} \to \mathscr{P}$ be the numerical flow of (5), i.e., the map defined by

$$\Phi_h(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) = (\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, t_{n+1}) .$$

The local error map is given by $E_h = \Phi_h - \varphi^h$. Consistency of Φ_h with respect to φ^h means that $E_h(\boldsymbol{z}) = \mathcal{O}(h^2)$ as $h \to 0$ for all $\boldsymbol{z} \in \mathscr{P}$. Concering the principle term of E_h we have the following result:

Lemma 1. *The $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and t components of the local error map E_h fulfills*

$$E_h(\boldsymbol{z}) = \begin{pmatrix} \frac{1}{6}h^3\Big((3\alpha-1)\big(\partial_{\boldsymbol{q}}A(\boldsymbol{z})\dot{\boldsymbol{q}} + \partial_t A(\boldsymbol{z})\big) - \partial_{\dot{\boldsymbol{q}}}A(\boldsymbol{z})C(\boldsymbol{z}) + (3\beta-1)\partial_{\boldsymbol{z}}B(\boldsymbol{z})X(\boldsymbol{z})\Big) + \mathcal{O}(h^4) \\ \frac{1}{2}h^2\Big((2\alpha-1)\big(\partial_{\boldsymbol{q}}A(\boldsymbol{z})\dot{\boldsymbol{q}} + \partial_t A(\boldsymbol{z})\big) - \partial_{\dot{\boldsymbol{q}}}A(\boldsymbol{z})C(\boldsymbol{z}) + (2\beta-1)\partial_{\boldsymbol{z}}B(\boldsymbol{z})X(\boldsymbol{z})\Big) + \mathcal{O}(h^3) \\ 0 \end{pmatrix} .$$

Proof. Compare Taylor expansions in h of $\Phi_h(\boldsymbol{z})$ and $\varphi^h(\boldsymbol{z})$. □

Using the lemma we immediatelly obtain the following result:

Theorem 2. *The method Φ_h, defined by* (5)*, has the following order of consistency properties.*

- *It is consistent for all α, β.*
- *It is second order accurate in position variables $\boldsymbol{q}$ for all α, β.*
- *If $\alpha = \beta = 1/2$ and A is independent of $\dot{\boldsymbol{q}}$, then it is second order accurate in all variables.*

Remark. In our application, the friction and damping in the contacts are very small in comparison to the stiffness. That is, $\|\partial_{\dot{\boldsymbol{q}}}A(\boldsymbol{z})\|$ is small in comparison to $\|\partial_{\boldsymbol{q}}A(\boldsymbol{z})\|$. Thus, we have "almost" second order accuracy for $\alpha = \beta = 1/2$.

4.2. Stability analysis when $B = 0$

In absence of constraint forces, i.e., when $B = 0$, the method Φ_h is fully explicit. Thus, it has a bounded stability region in terms of the step size h. We carry out a linear stability analysis for the scalar test equation given by

$$\ddot{q} = A(q, \dot{q}) = -kq - c\dot{q} , \qquad k, c \in \mathbb{R} . \tag{6}$$

Recall that stability of the method Φ_h means that $\lim_{n\to\infty} \Phi_h^n(\boldsymbol{z})$ is bounded. The result is as follows.

Theorem 3. *The stability region of the method* (5) *applied to* (6) *(with $B = 0$) is given by*

$$\Omega = \Big\{(hc, h^2k) \in \mathbb{R}^2 \setminus \{(0,0),(2-4\alpha,4)\} \; ; \; h^2k \geq 0, hc \leq 2 - \alpha h^2k, hc \geq (1/2-\alpha)h^2k\Big\} . \tag{7}$$

See Figure 2 for an illustration.

Proof. The numerical flow map Φ_h is linear for the test equation (6), i.e., it can be written

$$\Phi_h(q,\dot{q}) = R\begin{pmatrix} q \\ \dot{q} \end{pmatrix}, \qquad \text{with} \quad R = R(h,k,c) \in \mathbb{R}^{2\times 2}. \tag{8}$$

Thus, $\lim_{n\to\infty} \Phi_h^n(q,\dot{q}) = \lim_{n\to\infty} R^n \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$. This expression is bounded for all $(q,\dot{q})$ if and only if the eigenvalues of R satisfy the root condition, i.e., they lie on or within the unit circle and if on the unit circle they are simple. Written out, the characteristic equation $\det(R - \mathrm{Id}\lambda) = 0$ is

$$1 - hc + \frac{1}{2}h^2k(1-2\alpha) - \left(2 - hc - \frac{1}{2}h^2k(1+2\alpha)\right)\lambda + \lambda^2 = 0. \tag{9}$$

The solutions, i.e., the two eigenvalues, are

$$\lambda_\pm = 1 - \frac{hc}{2} - \frac{h^2k(1+2\alpha)}{4} \pm \frac{1}{4}\sqrt{\left(2hc + h^2k(1+2\alpha)\right)^2 - 16h^2k}. \tag{10}$$

The eigenvalues are equal (non-simple) and lie on the unit circle for $(hc, h^2k) \subset \{(0,0),(2-4\alpha,4)\}$. They are unequal (simple) and and one of them lie on the unit circle at $\partial\overline{\Omega} \setminus \{(0,0),(2-4\alpha,4)\}$. Furthermore, inside Ω the are both strictly inside the unit circle, and outside Ω at least one of them is strictly outside the unit circle. □

From (10) in the proof of Theorem 3 we get the following result, which is a discrete analog to what is known as critical damping.

Corollary 4. *The numerical solution is oscillatory in*

$$\Omega^{osc} = \left\{(hc, h^2k) \in \Omega \,;\, hc < 2h\sqrt{k} - \frac{1}{2}h^2k(1+2\alpha)\right\}$$

and non-oscillatory in $\Omega \setminus \Omega^{osc}$. *See Figure 2 for an illustration.*

Remark. The oscillation condition (critical damping) for the exact flow φ^h of (6) is given by

$$\omega^{osc} = \left\{(hc, h^2k) \,;\, hc < 2h\sqrt{k}\right\}.$$

Notice that for $\alpha > -1/2$ we always have $\Omega^{osc} \subset \omega^{osc}$. This means that critical damping in the numerical flow is "reached too fast" as c is increased from zero. Curiously, the choice $\alpha = -1/2$ gives exactly the correct critical damping.

4.3. Full stability analysis

We now extend the analysis to the linear test equation

$$\ddot{q} = A(q,\dot{q}) + B(q,\dot{q}) = -k_A q - c_A\dot{q} - k_B q - c_B\dot{q}, \tag{11}$$

with $A(q,\dot{q}) = -k_A q - c_A\dot{q}$ and $B(q,\dot{q}) = -k_B q - c_B\dot{q}$. This test equation is then discretized by the proposed scheme (5). Again, the numerical flow map is linear, i.e.,

$$\Phi_h(q,\dot{q}) = S\begin{pmatrix} q \\ \dot{q} \end{pmatrix}, \qquad \text{with} \quad S = S(h,k_A,c_A,k_B,c_B) \in \mathbb{R}^{2\times 2}. \tag{12}$$

Hence, our objective is to study the magnitude of the eigenvalues of the matrix S. The following result connects the stability analysis carried out in the previous section (the case $B = 0$) to the current case.

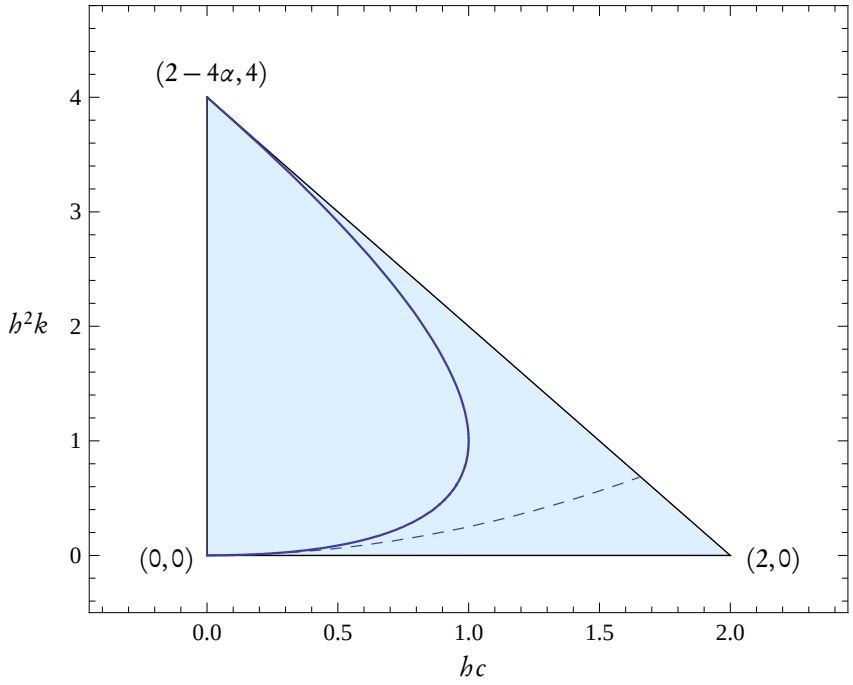

Figure 2: Stability region of the method (5) applied to (6) illustrated in the (hc, h^2k)–plane. The region is triangular, with corners marked. The full drawn curve within the region gives the "critical damping" condition, i.e., to the left of this curve the numerical solution is oscillatory and to the right it is non–oscillatory. The dashed curve gives the critical damping condition of the exact flow.

Lemma 5. *The transformation*

$$hc \quad \longleftrightarrow \quad \frac{2h(c_A + c_B)}{2 + 2\beta hc_B + \beta h^2 k_B} \tag{13a}$$

$$h^2k \quad \longleftrightarrow \quad \frac{2h^2(k_A + k_B)}{2 + 2\beta hc_B + \beta h^2 k_B} \tag{13b}$$

$$\alpha \quad \longleftrightarrow \quad \frac{\alpha h^2 k_A + \beta h^2 k_B}{h^2 k_A + h^2 k_B} \tag{13c}$$

takes the characteristic equation (9) *into the characteristic equation* $\det(S - \mathrm{Id}\lambda) = 0$.

Proof. Substitute (hc, h^2k, α) in (9) by the transformation (13). The resulting equation coincides with the characteristic equation $\det(S - \mathrm{Id}\lambda) = 0$. □

From Theorem 3 we know for which (hc, h^2k, α) the eigenvalues of R in (8) fulfill the root condition. Thus, using Lemma 5, we can investigate the root condition for the eigenvalues of S in (12). The following result, which asserts that the stability of the "explicit part" is not affected by the "implicit part", is then obtained:

Theorem 6. *For $\beta \geq 1/2$ the method* (5) *applied to* (11) *is stable for all $k_B, c_B \geq 0$ if $(hc_A, h^2k_A) \in \Omega$.*

Proof. From Theorem 3 it follows that the root condition is fulfilled for the roots of (9) if

(i) $h^2k \geq 0$,

(ii) $hc + \alpha h^2 k \leq 2$,

(iii) $(1/2 - \alpha)h^2k - hc \leq 0$.

Using Lemma 5, the root condition for the roots of $\det(S - \mathrm{Id}\lambda) = 0$ are fulfilled if the conditions obtained by substituting (13) in (i)–(iii) are fulfilled. Thus, our objective is to investigate (i)–(iii) after the substitution (13).

(i) Trivial because (13b) is always non-negative, so the condition is always true.

(ii) After substitution the condition becomes

$$\frac{2hc_A + 2\alpha h^2 k_A + 2hc_B + 2\beta h^2 k_B}{2 + 2\beta hc_A + \beta h^2 k_B} \leq 2 .$$

For $\beta \geq 1/2$ the left hand side is a decreasing function of both c_B and k_B. Thus, the left hand side is maximal when $c_B = k_B = 0$, which corresponds to the case $B = 0$.

(iii) After substitution the condition becomes

$$\frac{h^2 k_A(1 - 2\alpha) - 2hc_A + h^2 k_B(1 - 2\beta) - 2hc_B}{2 + 2\beta hc_A + \beta h^2 k_B} \leq 0 .$$

Again, for $\beta \geq 1/2$ the left hand side is a decreasing function of both c_B and k_B, and the maximum at $c_B = k_B = 0$ corresponds to the case $B = 0$. □

Remark. The stability result in Theorem 6 is the best possible, because for $B = 0$ it replicates Theorem 3, and for $A = 0$ it gives unconditional stability (cooresponding exactly to the classical θ–method with $\theta = \beta$).

5. Adaptivity

In order to increase the efficiency of the integration process it is important to introduce adaptive time stepping. There are various ways of doing this. It is out of scope of this paper to discuss any of them in full detail, but we mention two techniques.

The classical approach is to estimate the local error l^{err} at each time step, and then to consider the control objective $l^{err} \approx tol$ for some user specified tolerance level tol, see [Söd02, Söd03] for details.

Another approach, which is typically used in conjunction with geometric integration, is to introduce a Sundman transformation of the governing equations, which is a dynamic time transformation. Let s be a strictly positive real valued function on the phase space $\mathscr{P}$, called a scaling function, and introduce a new independent variable τ defined dynamically by $\mathrm{d}/\mathrm{d}\tau = s(\boldsymbol{z})\mathrm{d}/\mathrm{d}t$. The governing equations (4) then transform into

$$\frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}\tau} = s(\boldsymbol{z})X(\boldsymbol{z})\,. \tag{14}$$

Solutions to (14) correspond to time stretched solutions of (4). Thus, equidistant steps ε in the τ–domain correspond to variable steps $h = h(\boldsymbol{z}) = s(\boldsymbol{z})\varepsilon$ in the physical time domain. The easiest way to use this approach in conjunction with the proposed method (5) is to set $h = s(\boldsymbol{z}_n)\varepsilon$ at each step $n \to n+1$. For other, more intricate, techniques that also conserve the geometric properties of the flow (e.g. energy for conservative systems), see [MF06, HS05, HLW06].

5.1. Choice of control objective

As mentioned above, in classical ODE solvers the local error is estimated at each time step and is used as step size control objective. We suggest another choice based instead on the stability condition.

In Section 4 we found that for A only dependent on $\boldsymbol{q}$ and t (i.e. no damping), the linear stability condition is $h^2\sigma(\partial A(\boldsymbol{z})/\partial \boldsymbol{q}) \leq 4$. Thus, if an estimate $\sigma^{est} \approx \sigma(\partial_{\boldsymbol{q}} A(\boldsymbol{z}))$ is available, then a feasible step size control objective is $h^2\sigma^{est} \approx tol \leq 4$.

For the proposed algorithm (5), an estimate of $h\alpha\partial_{\boldsymbol{q}} A(\boldsymbol{z}_n)\dot{\boldsymbol{q}}$ is given by $A_{n+\alpha} - A_n$. Thus, the quantity

$$\sigma^{est} = \frac{||A_{n+\alpha} - A_n||}{h\alpha||\dot{\boldsymbol{q}}||} \tag{15}$$

gives an estimate of the stiffness in the direction of the flow. Notice that (15) is a function of $\boldsymbol{z}_n$, i.e., $\sigma^{est} = \sigma^{est}(\boldsymbol{z})$ is a function on the phase space $\mathscr{P}$. Thus, the corresponding scaling function is given by $s(\boldsymbol{z}) = 1/\sqrt{\sigma^{est}(\boldsymbol{z})}$. Hence, the Sundman transformation technique (14) may be used in conjunction with this control objective. Furthermore, from Lemma 1 it is evident that for $\alpha \neq 1/2$ this choice corresponds to keeping the principle relative local error term constant for velocity variables.

6. Numerical examples

In this section we present numerical examples of the proposed algorithm (5) applied to: (1) a simple linear problem consisting of two harmonic oscillators; (2) a non-linear pendulum problem in Cartesian coordinates; and (3) a complex multibody ball bearing problem. The last example is carried out in the multibody environment BEAST, where the method has been implemented.

6.1. Harmonic oscillators

The problem describes two particles, both with mass 1, moving on the real line. Between the two particles there is a linear spring with stiffness 10^3 and damping 10^2. One of the particles is attached to

a spring with stiffness 1 and no damping. The governing equations are

$$\ddot{q} = -\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} q - 10^3 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} q - 10^2 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \dot{q} \,.$$

In terms of (3), we split the acceleration field so that the first term corresponds to $A(z)$, and the two last terms to $B(z)$. Thus, A corresponds to the weak spring, and B to the stiff spring "constraining" the two particles to stick together. We are interested in resolving the dynamics in A, but not that in B. The frequencies in the system are $1/(4\pi)$ Hz (due to the weak spring) and $(1/\pi) \cdot 10^3$ Hz (due to the stiff spring). As initial data we choose $q_0 = (1.0, 1.1)$ and $\dot{q}_0 = (0, 0)$.

Numerical simulation with the method (5) is carried out for the constant step size $h = 1$, and method parameters $\alpha = 1/2$ and $\beta = 0.8$. A comparison is given with the classical θ–method, with $\theta = 1/2$ and $\theta = 0.8$. This method is fully implicit in both A and B, and thus requires many more evaluations of A (which we "pretend" to be expensive).

The results in Figure 3 show that, although the θ–method is more expensive per time step, it is less accurate. This is due to the fact that $\alpha = 1/2$ corresponds to a symplectic integrator for the A–part (the explicit part), which is known to have superior accuracy for conservative systems. The θ–method with $\theta = 1/2$ (i.e. the implicit midpoint rule) is also symplectic, but with this choice the highly oscillatory dynamics is not damped out correctly.

6.2. Non-linear pendulum

The problem is a pendulum expressed in Cartesian coordinates $q = (q^x, q^y)$. The length and mass of the pendulum is 1. Thus, a constraint is given by $||q||^2 - 1 = 0$. This constraint is modeled by as stiff spring. The governing equations are

$$\ddot{q} = g - 10^4(||q||^2 - 1)q \,,$$

where the gravity is given by $g = (0, -1)$. We choose A as the first term and B as the second term.

Initial conditions are given by $q_0 = (1.01, 0)$ and $\dot{q}_0 = (0, 0)$. Numerical simulation is carried out for the constant step size $h = 0.01$, and method parameters $\alpha = 1/2$ and $\beta = 0.6$. A comparison is given with the θ–method, with $\theta = 1/2$ and $\theta = 0.6$.

The plots in Figure 4 show the error in the variable q^x. It is small for the $\theta = 1/2$ method, but the solution there contains high oscillations due to the constraint forces, which are not damped out correctly (as in the previous example). Figure 5 shows the constraint error, i.e., the quantity $||q||^2 - 1$. Furthermore, these small oscillations cause the Newton solver to require significantly more iterations, which means more evaluations of A (which we pretend to be expensive).

6.3. Complex ball bearing

This example consists of the ball bearing model illustrated in Figure 1. The outer ring is held fixed, and the inner ring is rotated with 10^4 revolutions per minute. Further, the inner ring is loaded axially with a constant force of 10^3 N.

Simulations of the system is carried out within the software package BEAST with: (i) the proposed integrator with $\alpha = 1/2$, $\beta = 0.8$ and constant step size $h = 10^{-6}$ s; (ii) a standard implicit BDF–solver with adaptive time steps (CVODE, see [Hin]). The plots in Figure 6 show: the contact forces between the first ball and the outer ring; between the first ball and the cage; and the angular velocity of the cage. The results are nearly identical. Since these variables are highly sensitive (especially contact forces on the cage) high similarity between the two simulations indicate that the accuracy is about the same.

Statistics from the two simulations are given in the table below.

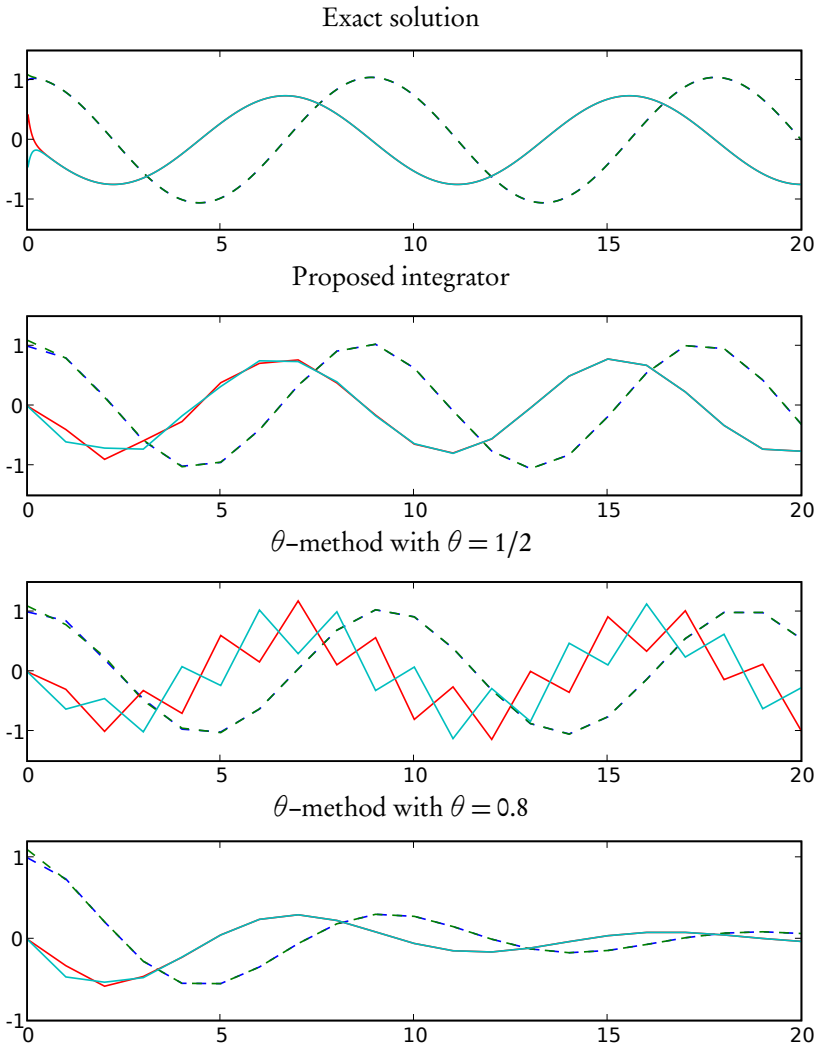

Figure 3: Numerical results for the test problem in Section 6.1. The full drawn curves are the position variables $\boldsymbol{q}$ and the dashed curves are the velocity variables $\dot{\boldsymbol{q}}$. The upper graph shows the exact solution. Notice that the θ method gives a less accurate result than the proposed method (both for $\theta = 1/2$ and $\theta = 0.8$), even though it is more expensive in terms of evaluations of A.

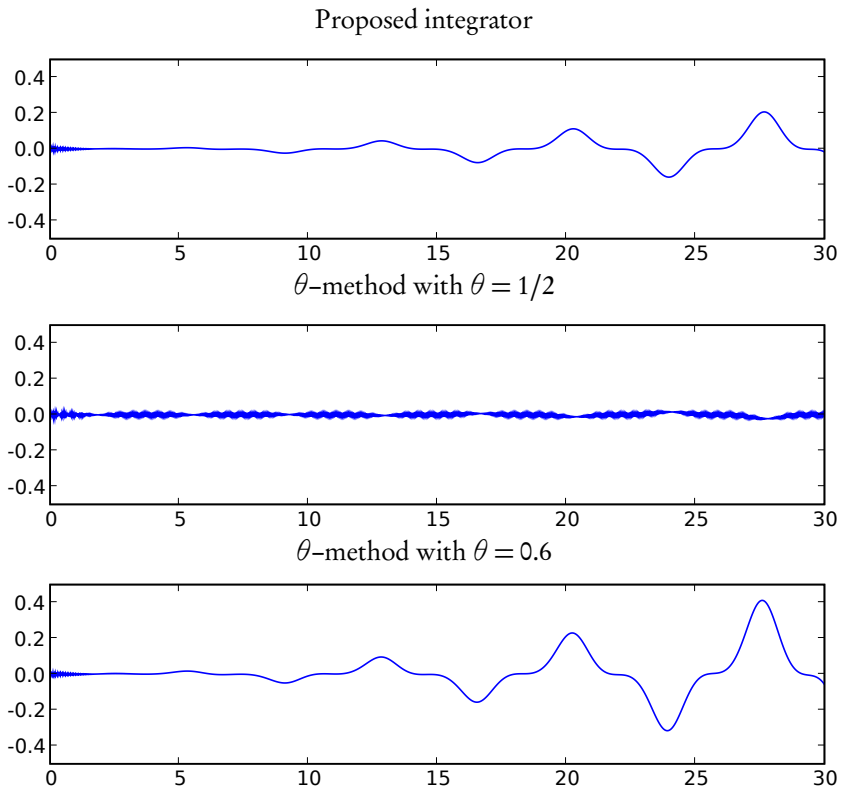

Figure 4: Numerical results for the test problem in Section 6.2. Global error in the q^x variable. Notice that, although the error is small for the $\theta = 1/2$ method, the solution is highly oscillatory.

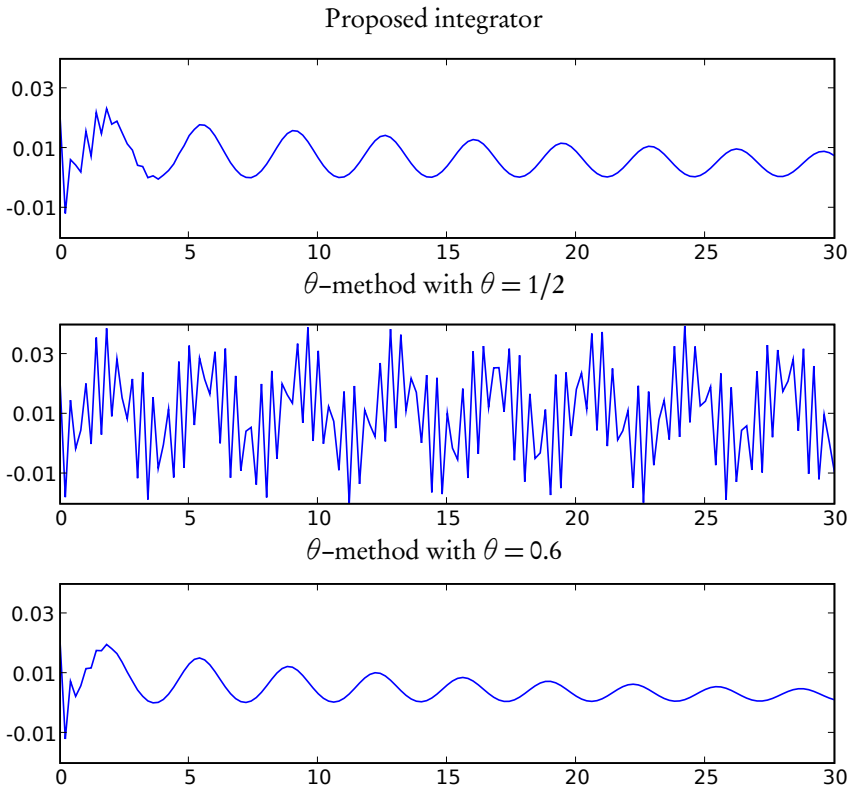

Figure 5: Numerical results for the test problem in Section 6.2. Error in the constraint $||q|| - 1$.

	mean h	force evaluations
BDF–solver	$6.8 \cdot 10^{-7}$ s	1588[1]
proposed method	$1 \cdot 10^{-6}$ s	600

Thus, we gain a factor of about $1588/600 \approx 2.6$ in efficiency. We estimate that the gain will increase if the proposed solver is implemented with adaptive time steps. Further, for rigid models a Jacobian evaluation is relatively cheap (12 force evaluations are needed). With flexible bodies it is much more expensive ($12+2n_f$ force evaluations with n_f number of flexible states), so the potential efficiency gain for models with flexible bodies is promising.

7. Conclusions

A new numerical integrator specifically designed for problems of the type described in Section 2 (e.g. multibody problems with contact forces between bodies) have been proposed. Contrary to standard methods for such problems, the proposed integrator requires only one evaluation of the contact forces per time step, and no contact Jacobians.

Consistency and stability analysis of the proposed integrator have been carried out, and a control objective for adaptive step size implementations has been proposed, based on the stability condition.

Numerical examples show that the proposed integrator is more efficient (in terms of number of contact force evaluations) in comparison with standard implicit integrators.

[1] 1288 force evaluations from iterations plus 25 Jacobian evaluations. Each Jacobian require 12 force evaluation, so in total $1288 + 25 \cdot 12 = 1588$.

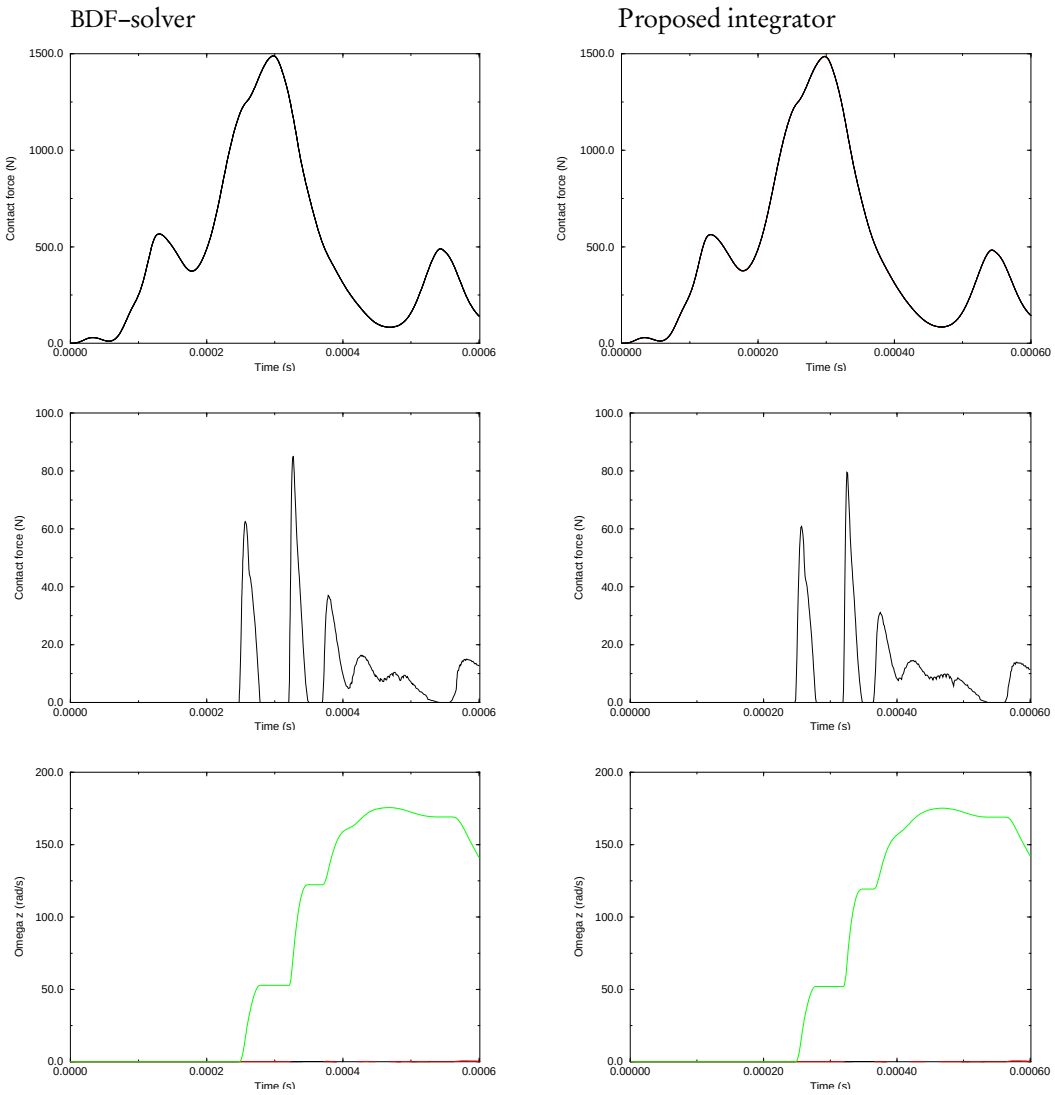

Figure 6: Numerical results for the test problem in Section 6.3. In the left column a standard adaptive BDF–solver is used. In the right column the proposed method is used. The first row is the contact force between one of the balls and the outer ring. The second row is the contact force between one of the balls and the cage. The third row is the angular velocity of the cage.

References

[Hin] A. Hindmarsh. *CVODE open-source software*. `www.llnl.gov/CASC/sundials/`.

[HLW06] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[HS05] Ernst Hairer and Gustaf Söderlind. Explicit, time reversible, adaptive step size control. *SIAM J. Sci. Comput.*, 26(6):1838–1851 (electronic), 2005.

[MF06] Klas Modin and Claus Führer. Time-step adaptivity in variational integrators with application to contact problems. *ZAMM Z. Angew. Math. Mech.*, 86(10):785–794, 2006.

[MQ02] Robert I. McLachlan and G. Reinout W. Quispel. Splitting methods. *Acta Numer.*, 11:341–434, 2002.

[Nak06] I. Nakhimovski. *Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analyses*. PhD thesis, Linköpings Universitet, Linköping, 2006.

[SF01] L-E Stacke and D. Fritzson. Dynamical behaviour of rolling bearings: simulations and experiments. *Proc. Instn. Mech. Engrs.*, 215:499–508, 2001.

[Söd02] Gustaf Söderlind. Automatic control and adaptive time-stepping. *Numer. Algorithms*, 31(1-4):281–310, 2002. Numerical methods for ordinary differential equations (Auckland, 2001).

[Söd03] Gustaf Söderlind. Digital filters in adaptive time-stepping. *ACM Trans. Math. Software*, 29(1):1–26, 2003.