ELSEVIER Journal of Computational and Applied Mathematics 66 (1996) 443–456

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

On the numerical solution of a wheel suspension benchmark problem

B. Simeon*

Fachbereich Mathematik, Technische Hochschule Darmstadt, Schlossgartenstrasse 7, D-64289 Darmstadt, Germany

Received 27 July 1994; revised 19 July 1995

Abstract

For the simulation of a five-link wheel suspension, an alternative formulation with built-in projection and corresponding numerical algorithms are presented. The wheel suspension, a benchmark problem in vehicle dynamics, can only be modeled as a system of differential-algebraic equations (DAEs). Besides the numerical integration, the computation of consistent initial values and the treatment of discontinuities play an important role. Simulation results illustrate the performance of the introduced algorithms.

Keywords: Differential-algebraic equations; Projecting descriptor form; Numerical integration; Wheel suspension problem

AMS classification: 65L05, 70-08

1. Introduction

Advanced simulation software gains more and more importance in the design and analysis of vehicle prototypes. To model such mechanical systems, the *multibody system approach* provides the basic methodology [15, 22, 23]. The vehicle is described as a collection of interconnected rigid bodies which can move relative to one another. Joints constrain the relative motion of pairs of bodies while springs and dampers act as compliant elements representing the elasticity. In general, the governing equations of motion form a system of *differential-algebraic equations* (*DAEs*) in redundant coordinates.

This article presents both an alternative formulation with built-in projection and numerical algorithms for the computational tasks associated with these DAEs. The algorithms are applied to the model of a *five-link wheel suspension*, a road vehicle benchmark problem due to Hiller and Frik [16]. The wheel suspension features the so-called closed kinematic loops in the multibody topology

* E-mail: simeon@mathematik.th-darmstadt.de.

SSDI 0377-0427(96)00169-7

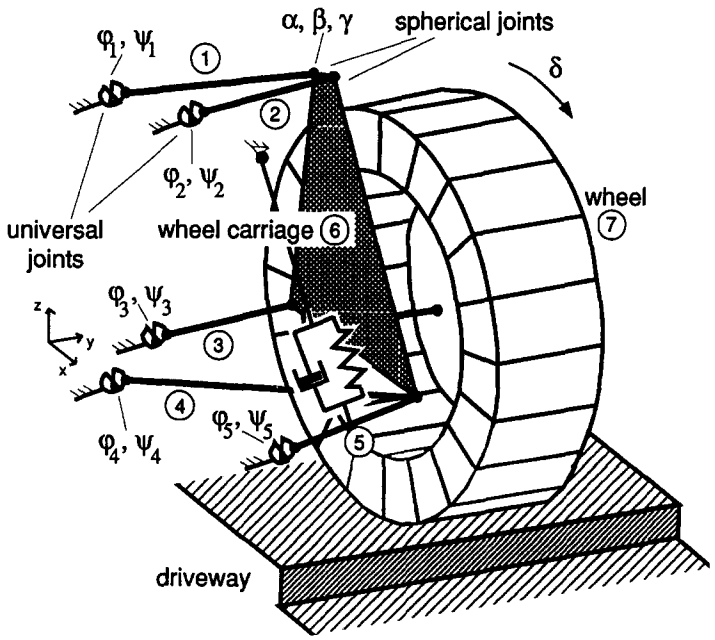

Fig. 1. Five-link wheel suspension.

and thus serves as an example of a constrained mechanical system which cannot be modeled as a system of ordinary differential equations in minimum coordinates.

The problem description and the governing equations of motion are introduced in Section 2, which also specifies the computational tasks. Section 3 presents the alternative formulation, the *projecting descriptor form*, while Section 4 outlines the numerical methods and investigates their properties. Finally, Section 5 gives various simulation results. Throughout the text, the reader is not assumed to be familiar with the numerical analysis of DAEs, and emphasis is placed on an algorithmic way of presentation.

2. Problem description

Consider the model of a passenger car wheel suspension in Fig. 1. Hiller and Frik [16] proposed this spatial multibody system as a benchmark problem in vehicle system dynamics since it enables a thorough ride quality analysis and features several properties which make it a challenging example for the computational treatment. Seven rigid bodies (rods ①–⑤, wheel carrier ⑥, wheel ⑦) and various interconnection elements like spherical and universal joints are used to model the real mechanical system. Because of the five rods connecting car body and wheel carrier, the suspension is called a *five-link wheel suspension*. It is assumed that the car body is fixed to the inertial reference frame and that the driveway is represented by an excitation acting on the wheel. Moreover, a spring damper element between body ⑤ and car body stands for the shock absorber and a coil spring, and another spring damper models the tire.

2.1. Equations of motion

Depending on time t, $n_p = 14$ position coordinates

$$p = [\varphi_1, \psi_1, \alpha, \beta, \gamma, \varphi_2, \psi_2, \varphi_3, \psi_3, \varphi_4, \psi_4, \varphi_5, \psi_5, \delta]^T$$

describing the relative angles in the joints and the rotation of the wheel uniquely specify the position and orientation of all bodies. The dynamic behavior of such a multibody model is given by the Euler–Lagrange equations or Lagrange equations of type one [15]:

$$\dot{p} = v, \tag{1a}$$

$$M(p)\dot{v} = f(p, v, t) - G(p)^T\lambda, \tag{1b}$$

$$0 = g(p), \tag{1c}$$

where $\dot{p}$ denotes the time derivative of $p, v(t) \in \mathbb{R}^{n_p}$ the velocity coordinates, $\lambda(t) \in \mathbb{R}^{n_\lambda}$ the Lagrange multipliers, and $M(p) \in \mathbb{R}^{n_p, n_p}$ the symmetric, positive-definite mass matrix. The mappings $f: \mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \times \mathbb{R} \to \mathbb{R}^{n_p}$ and $g: \mathbb{R}^{n_p} \to \mathbb{R}^{n_\lambda}$ define the forces and the kinematic constraints with $G(p) = \partial g(p)/\partial p$ as Jacobian. M, f, and g are assumed to be sufficiently smooth.

The complexity of the governing equations of motion (1) requires a computer formulation. Here, the equations were generated using the symbolic multibody program NEWEUL [23], forming a FORTRAN77 subroutine of about 7000 lines of source code. This subroutine, equipped with standard interface [6] and detailed description, is available to interested readers, see the e-mail address.

The holonomic constraint equations (1c) result from the joints which constrain the relative motion of pairs of bodies. For this reason, the wheel suspension has only $n_p - n_\lambda = 14 - 12 = 2$ degrees of freedom. However, the multibody topology of this model is characterized by 4 *closed kinematic loops*, which make it impossible to apply the implicit function theorem to (1c) globally. Thus, a reduction of the differential-algebraic system or *descriptor form* (1) to a system of ordinary differential equations in minimum coordinates, the *state-space form* or Lagrange equations of type two, cannot be taken into consideration.

2.2. Computational tasks

In order to specify the computational tasks associated with the descriptor form (1), the index is determined first. Intuitively, the index provides a measure of the singularity of a DAE and is defined by the number of differentiation steps necessary for the transformation to an ODE (see [6, 7, 14] for definitions). As a rule of thumb, the higher the index of a DAE, the more complicated is its numerical analysis.

Differentiation of the constraints (1c) on position level with respect to time t leads to

$$0 = \frac{d}{dt} g(p) = G(p)\dot{p} = G(p)v, \tag{2}$$

$$0 = \frac{d^2}{dt^2} g(p) = G(p)\dot{v} + z(p, v), \qquad z(p, v) := \left(\frac{d}{dt} G(p)\right) v = \dot{G}(p, v)v. \tag{3}$$

Obviously, any solution of (1) must satisfy the constraints (2) on velocity level and the constraints (3) on acceleration level. Consequently, (1b) and (3) imply

$$\begin{bmatrix} \dot{v} \\ \lambda \end{bmatrix} = Q(p)^{-1} \begin{bmatrix} f(p,v,t) \\ -z(p,v) \end{bmatrix}, \qquad Q(p) := \begin{bmatrix} M(p) & G(p)^{\mathrm{T}} \\ G(p) & 0 \end{bmatrix}, \tag{4}$$

where the matrix $Q(p)$ is regular for a well-specified multibody system. Differentiating the right-hand side of λ one more time results in a system of ODEs in terms of p, v, λ. To summarize, three differentiation steps transform the DAE (1) into an ODE, which means that the index of (1) is 3. Furthermore, the differentiation steps show that (1) is characterized by additional *hidden constraints*, which impose *consistency conditions* on initial values $p_0 = p(t_0)$, $v_0 = v(t_0)$, $\lambda_0 = \lambda(t_0)$ and provoke severe difficulties for the numerical integration [6, 14, 27].

On the other hand, consider again the ODE for p and v given by (1a) and (4). This formulation is derived by differentiating the constraints twice and thus lowering the index from 3 to 1. Standard ODE methods can be applied to it by solving the linear system (4) for $\dot{v}$ and λ. Due to differentiation, however, this approach lacks information of the original constraints on position level (1c) and on velocity level (2) and may turn unstable — the numerical solution *drifts off from the constraints* [6, 9, 26].

Two computational tasks arise accordingly for any numerical method.

Task 1: Compute consistent initial values p_0, v_0, λ_0 satisfying the constraints (1c) and (2) and the linear system (4).

Task 2: Integrate the DAE (1) such that all constraints (1c), (2) and (3) are satisfied by the numerical solution.

A third task must be performed during the numerical integration of the wheel suspension. Due to the driveway specification and a piecewise defined force law of the tire model, discontinuities may appear in the forces f, which should be treated by the *switching function technique* [3, 9].

Task 3: Check signs of switching functions and interrupt the integration process if a zero was found.

3. Projecting descriptor form

Constructing numerical schemes for the descriptor form (1) is a focus of research in numerical analysis and mechanical engineering. In contrast to the ODE case, not only discretization schemes are analyzed but also *stabilized reformulations* of (1), see [4, 11, 12, 20]. Here, an index-reduced formulation is presented which features a built-in projection for inconsistent data, is equivalent to (1) but of index 1 instead of 3. The derivation of the numerical algorithms in Section 4 will be based on this formulation.

The projecting descriptor form realizes the following idea: With the Lagrange multipliers λ in (1) being generalized constraint forces corresponding to the constraints (1c), is it possible to use additional multipliers, say η and τ, which correspond to the hidden constraints (2) and (3)? In [12], Gear et al. gave one method how to introduce additional constraints and multipliers. Their approach, however, ends up with an index 2 system. Instead, the following formulation is derived

directly from the Euler–Lagrange formalism by imposing constraints (1c) and hidden constraints (2), (3) *simultaneously* on the motion of the multibody system, yielding the *projecting descriptor form* [25]

$$M(p)\dot{q} = M(p)v + \dot{M}(p,\dot{p})(p-q) + \dot{G}(p,\dot{p})^{\mathrm{T}}\tau, \tag{5a}$$

$$M(p)\dot{u} = M(p)w + \dot{M}(p,v)(v-u) + \dot{G}(p,v)^{\mathrm{T}}\eta, \tag{5b}$$

$$0 = \begin{bmatrix} M(p)w - f(p,v,t) + G(p)^{\mathrm{T}}\lambda \\ G(p)w + \dot{G}(p,v)v \end{bmatrix}, \tag{5c}$$

$$0 = \begin{bmatrix} M(p)(v-u) + G(p)^{\mathrm{T}}\eta \\ G(p)v \end{bmatrix}, \tag{5d}$$

$$0 = \begin{bmatrix} M(p)(p-q) + G(p)^{\mathrm{T}}\tau \\ g(p) \end{bmatrix}. \tag{5e}$$

Here additional position variables $q \in \mathbb{R}^{n_p}$, velocity variables $u \in \mathbb{R}^{n_p}$, acceleration variables $w \in \mathbb{R}^{n_p}$, and multipliers $\eta, \tau \in \mathbb{R}^{n_\lambda}$ have been introduced.

Theorem 3.1. *The projecting descriptor form* (5) *is of index* 1. *For arbitrary initial values* u_0 *and initial values* q_0 *sufficiently close to the constraint manifold* $\{\tilde{p}: 0 = g(\tilde{p})\}$, *the initial values* $p_0, v_0, w_0, \lambda_0, \eta_0\, \tau_0$ *are uniquely determined, the solution in terms of* p, v, λ *satisfies* (1), *and it holds* $\dot{v} = w$, $\dot{\eta} = \dot{\tau} = 0$.

Proof. First, the number of differentiation steps necessary to transform (5) to a system of ODEs is determined. Eqs. (5c) and (5d) are rewritten as

$$Q(p)\begin{bmatrix} w \\ \lambda \end{bmatrix} = \begin{bmatrix} f(p,v,t) \\ -\dot{G}(p,v)v \end{bmatrix}, \qquad Q(p)\begin{bmatrix} v \\ \eta \end{bmatrix} = \begin{bmatrix} M(p)u \\ 0 \end{bmatrix}. \tag{6}$$

Due to the regularity of matrix Q from (4), (6) can be solved for the algebraic variables $w = W(p,v,t)$, $\lambda = \Lambda(p,v,t)$, $v = V(p,u,t)$, and $\eta = E(p,u,t)$. Thus, one differentiation step suffices to obtain an ODE for these variables. For the remaining variables p and τ, (5e) is differentiated, yielding $0 = G(p)\dot{p}$ and

$$\begin{aligned} M(p)\dot{p} + \dot{M}(p,\dot{p})p &= M(p)\dot{q} + \dot{M}(p,\dot{p})q - G(p)^{\mathrm{T}}\dot{\tau} - \dot{G}(p,\dot{p})^{\mathrm{T}}\tau \\ &\overset{(5a)}{\Rightarrow} M(p)\dot{p} = M(p)v - G(p)^{\mathrm{T}}\dot{\tau} \\ &\Rightarrow Q(p)\begin{bmatrix} \dot{p} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} M(p)v \\ 0 \end{bmatrix}. \end{aligned} \tag{7}$$

Again, the regularity of Q implies that (7) is an ODE for p and τ whence the index of (5) is 1. Next, concerning the initial values for given q_0 and u_0, the linear systems (6) define $v_0, w_0, \lambda_0, \eta_0$ uniquely. However, the nonlinear constraints (5e) enable only a local result for p_0 and τ_0. Due to the implicit function theorem, (5e) can be solved for p_0 and τ_0 depending on q_0 if

$$Q(p_0) + \begin{bmatrix} (\partial/\partial p M(p_0))(p_0 - q_0, \,) + (\partial/\partial p G(p_0)^{\mathrm{T}})(\tau_0, \,) & 0 \\ 0 & 0 \end{bmatrix}$$

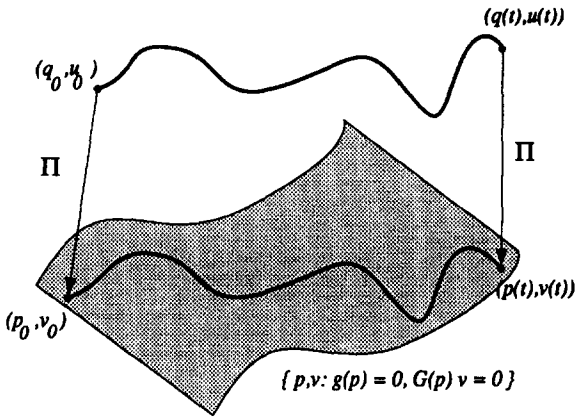

Fig. 2. Interpretation of (5).

is regular, which holds if q_0 is sufficiently close to the manifold $\{\tilde{p}\colon 0 = g(\tilde{p})\}$. Finally, it remains to show that the solution in terms of p, v, λ satisfies (1). Using one more time the regularity of Q, it follows that $GM^{-1}G^{\mathrm{T}}$ is regular. Hence, (7) can be solved explicitly for $\dot{\tau}$, and from $0 = G(p)\dot{p} = G(p)v$ one concludes

$$G(p)M(p)^{-1}G(p)^{\mathrm{T}}\dot{\tau} = G(p)v = 0 \;\Rightarrow\; \dot{\tau} = 0 \text{ and } \dot{p} = v.$$

By analogous arguments, starting at (5d) and inserting (5b), one shows $\dot{\eta} = 0$ and $\dot{v} = w$. Inserting the latter relation in (5c), (1b) follows immediately. This completes the proof since the constraints (1c) are part of (5e). □

Theorem 3.1 gives the basic properties of the projecting descriptor form (5): The index is reduced to 1, and the solution in terms of p, v, and λ still satisfies the equations of motion — there is no loss of information. Fig. 2 illustrates the built-in projection mechanism. The integration starts with potentially inconsistent data q_0, u_0 and then proceeds. Though the solution in terms of q, u may not satisfy the constraints, (5e) and (5d) define implicitly a projection Π which maps q and u to p and v on the constraint manifold.

Moreover, (5) contains additional information in terms of the multipliers η and τ, which are a measure for the deviation of q and u from the constraint manifold,

$$\eta = (G(p)M(p)^{-1}G(p)^{\mathrm{T}})^{-1}G(p)u, \tag{8}$$

$$\tau = (G(p)M(p)^{-1}G(p)^{\mathrm{T}})^{-1}g(q) + \mathrm{O}(\|p - q\|^2). \tag{9}$$

η and τ vanish if q and u satisfy the constraints. In this case, the *curvature expressions* in terms of $\dot{M} = (\mathrm{d}/\mathrm{d}t)M$ and $\dot{G} = (\mathrm{d}/\mathrm{d}t)G$ in (5a) and (5b) also vanish. Otherwise, however, these expressions are necessary in order to guarantee the properties stated in Theorem 3.1, in particular $\dot{\eta} = \dot{\tau} = 0$.

Remarks.

- For constant mass matrix M, (5e) can be interpreted as a *minimum distance criterion* for the projection of inconsistent data q. This criterion utilizes a quadratic form in the mass matrix M and reads

$$(p-q)^{\mathrm{T}}M(p-q) \stackrel{!}{=} \min \quad \text{subject to } g(p)=0. \tag{10}$$

  Similarly, (5d) is related to the criterion (with p fixed)

$$(v-u)^{\mathrm{T}}M(p)(v-u) \stackrel{!}{=} \min \quad \text{subject to } G(p)v=0. \tag{11}$$

- The curvature expressions in terms of $\dot{M}$ and $\dot{G}$ are available. The forces f contain an expression $\dot{M}v$ representing generalized Coriolis forces, and the acceleration constraints (3) include $\dot{G}$. Hence, (5) does not require additional derivative information.
- In contrast to the formulation of Gear et al. [12] where the additional multipliers vanish along any solution, in (5) only the derivatives of the additional multipliers vanish. This is an essential requirement to derive the built-in projection mechanism.

4. Numerical algorithms

The above introduced projecting descriptor form consists of $2n_p$ differential and $3(n_p+n_\lambda)$ constraint equations. In most practical applications, a direct discretization of this index 1 system is rather expensive. Yet it can be used in an indirect way to derive numerical methods since it contains all information of the equations of motion in an explicit form.

4.1. Computing consistent data

The projection of inconsistent data q,u with $g(q)\neq 0$ and $G(q)u\neq 0$ to the constraint manifold given by (1c) and (2) plays a key role in the following. Basically, there are various ways to perform such a projection step [9], but only one corresponds to the *natural metric* of the multibody system, which is the quadratic form (10) and (11) in the mass matrix M. In this way, the projecting descriptor form supplies with (5e) and (5d) the following algorithm for the computation of consistent data p and v:

Start: q,u;

$$\textit{Solve} \quad \begin{aligned} 0 &= M(p)(p-q)+G(p)^{\mathrm{T}}\tau \\ 0 &= g(p) \end{aligned} \quad \text{for } p,\tau; \tag{12a}$$

$$\textit{Solve} \quad \begin{aligned} 0 &= M(p)(v-u)+G(p)^{\mathrm{T}}\eta \\ 0 &= G(p)v \end{aligned} \quad \text{for } v,\eta. \tag{12b}$$

The nonlinear system (12a) can be solved efficiently by a chord-Newton process of the form $p^{(j+1)}=p^{(j)}-\Delta p^{(j)}$, $\tau^{(j+1)}=\tau^{(j)}-\Delta\tau^{(j)}$, $j=0,1,\ldots$, with $\Delta p^{(j)},\Delta\tau^{(j)}$ given by

$$\begin{bmatrix}\Delta p^{(j)}\\ \Delta\tau^{(j)}\end{bmatrix} = Q(q)^{-1}\begin{bmatrix}M(p^{(j)})(p^{(j)}-q)+G(p^{(j)})^{\mathrm{T}}\tau^{(j)}\\ g(p^{(j)})\end{bmatrix}. \tag{13}$$

For q sufficiently close to the constraint manifold, the iteration converges at least linearly to the solution p [19]. Of course, the initial guess q may not be arbitrary but must be somehow related to the configuration of the mechanical system. This requirement, however, is also essential for the simulation since otherwise the results have no technical interpretation.

Combined with the solution of the linear system (12b) for v and η, (13) can be employed both for the computation of consistent initial values, cf. Task 1, and a stabilization of the numerical integration process, see below. It should be remarked that the terminology *natural metric* was first introduced by Alishenas [1] and Lubich [17] who both give a similar algorithm with $M(q)$ and $G(q)$ in (12a) instead of $M(p)$ and $G(p)$.

4.2. Numerical integration

For the simulation of the wheel suspension, a Runge–Kutta method is presented which is based on the information of the projecting descriptor form. We start with the index 1 system given by (1a) and (4). Standard ODE methods [28] can be applied to it by solving the linear system (4) for $\dot{v}$ and λ. In order to guarantee that the constraints are satisfied, the integration process must be stabilized by applying additional projections. After each integration step the numerical solution is projected to the manifold of position and velocity constraints and then the next integration step is performed [1, 10]. In the more general setting of ODEs with *invariants*, the convergence of such a two-stage algorithm was shown in [24] for one-step methods and in [10] for multistep methods. The basic assumption is that the projected solution p, v satisfies

$$q - p = \mathrm{O}(h^{k+1}), \qquad u - v = \mathrm{O}(h^{k+1}) \tag{14}$$

for an integration method of order k and stepsize h. The standard convergence proof can be carried over in this case. Condition (14) is the only one on the projection step, and thus several strategies have been developed [1, 3, 9]. Here, the projection (12) induced by the projecting descriptor form is used. It reflects the natural metric of the mechanical system and is independent of a specific discretization scheme.

The algorithm presented below for the numerical integration is based on the fifth-order explicit Runge–Kutta method of Dormand and Prince [8] with coefficients a_{ij}, c_i, $i = 2, \ldots, 7$, $j = 1, \ldots, i-1$ and embedded error coefficients a_{8j}, $j = 1, \ldots, 7$. It employs a standard weighted root mean square norm for error estimation

$$\|x\|^2 := \frac{1}{n} \sum_{l=1}^{n} \left(\frac{x^l}{\mathrm{RTOL} \cdot \mathrm{WT}^l + \mathrm{ATOL}} \right)^2 \quad \text{for } x = (x^1, \ldots, x^n)^{\mathrm{T}}, \tag{15}$$

where RTOL and ATOL stand for the prescribed relative and absolute tolerances and the weights $\mathrm{WT}^l = |\tilde{x}^l|$ are defined by some reference data $\tilde{x}$. With these preparations, the basic algorithm for one time step from p_0, v_0, λ_0 at time t_0 to p_1, v_1, λ_1 at time $t_1 = t_0 + h$ with stepsize h reads

For $i = 1, \ldots, 6$
$P_i = p_0 + h \sum_{j=1}^{i-1} a_{ij} P'_j$; $V_i = v_0 + h \sum_{j=1}^{i-1} a_{ij} V'_j$;
$P'_i = V_i$;

Solve

$$Q(P_i)\begin{bmatrix} V_i' \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} f(P_i, V_i, t_0 + c_i h) \\ -z(P_i, V_i) \end{bmatrix} \quad \text{for } V_i', \Lambda_i;$$

$\bar{P}_7 = p_0 + h\sum_{j=1}^{6} a_{7j}P_j'$; $\bar{V}_7 = v_0 + h\sum_{j=1}^{6} a_{7j}V_j'$;
Projection $\bar{P}_7 \to P_7$, $\bar{V}_7 = V_7$ *using* (12);
if (*no convergence*): *reduce stepsize and repeat step*;
$P_7' = V_7$;
Solve

$$Q(P_7)\begin{bmatrix} V_7' \\ \Lambda_7 \end{bmatrix} = \begin{bmatrix} f(P_7, V_7, t_0 + h) \\ -z(P_7, V_7) \end{bmatrix} \quad \text{for } V_7', \Lambda_7;$$

Error estimation: $\hat{P}_7 = p_0 + h\sum_{j=1}^{7} a_{8j}P_j'$; $\hat{V}_7 = v_0 + h\sum_{j=1}^{7} a_{8j}V_j'$;
if ($\|P_7 - \hat{P}_7\| > 1$ *or* $\|V_7 - \hat{V}_7\| > 1$): *reduce stepsize and repeat step*;
$p_1 := P_7$, $v_1 := V_7$, $\lambda_1 := \Lambda_7$;

Several remarks on this algorithm should be given. For one, the application of a projection procedure like (12) is not straightforward since the end of one integration step and the beginning of the next step are not separated. The Dormand and Prince method consists of 7 stages where the last stage yields an approximation $\bar{P}_7, \bar{V}_7$ and where the increments P_7', V_7' required for error estimation enter the next integration step. In fact, only 6 effective stages arise per step and consequently, two possibilities for the projection step exist: Either the projection is performed before the evaluation of P_7', V_7' or thereafter. The first alternative is favorable since it provides P_7', V_7' already evaluated for the projected values $P_7 = p_1$, $V_7 = v_1$, in contrast to the second alternative. Of course, the embedded solution used for error estimation is then perturbed, but due to (14) the leading term of the fourth-order solution $\hat{P}_7, \hat{V}_7$ is not affected.

Next, practical experience shows that projection after each integration step is often more expensive but does not yield more accurate results than skipping the projection (12a) for some steps and performing only the "velocity projection" [1] of (12b) (cf. Section 5). The crucial point is the *feedback* of the drift, i.e. the deviation in the constraints, on the integration error.

In this context, the projecting descriptor form gives both an interpretation of the integration process and information on the feedback of the drift. For this purpose, consider again (5). As long as the constraints are satisfied, the additional multipliers η, τ vanish and the differential equations in (5) correspond to the index 1 system (1a) and (4). If the constraints are violated, η and τ measure the deviation and enter the differential equations. If η and τ remain small compared to the discretization error, their influence or feedback can be neglected. If they grow larger, a projection step must be performed, and η and τ are set to zero again. Performing only the "velocity projection" now means that η is set to zero after each step, which is inexpensive, while τ is neglected. In case of constant matrices M and G, however, (5) shows that there is no feedback of the drift on the integration error.

A measure of the feedback of the drift is given by $\Delta\tau^{(0)}$, the first increment in the iteration (13). Yet the integration error is controlled in terms of the variables p, and thus $\Delta p^{(0)} = -M^{-1}G^{\mathrm{T}}\Delta\tau^{(0)}$ is used instead. The following criterion has proved to yield an efficient projection control. It is

based on $\|\Delta p^{(0)}\|$ and employs an integer variable k which specifies the number of steps to be performed without the projection (12a). Default values are $k = 4$, $k_{\max} = 8$, $k_{\min} = 1$:

$$\begin{aligned}
&\textit{Perform the projection } (12);\\
&\textit{if } \|\Delta p^{(0)}\| < 0.009\text{: } k_{\text{new}} = \min(2k, k_{\max})\\
&\textit{else if } \|\Delta p^{(0)}\| < 0.02\text{: } k_{\text{new}} = k\\
&\textit{else } k_{\text{new}} = \max(\tfrac{1}{2}k, k_{\min})\\
&f_i
\end{aligned} \tag{16}$$

Considering the average computational effort for one integration step, it turns out that only 6 decompositions of matrix Q are necessary to solve 9 arising systems of linear equations including the projection procedure (12). For the projection of position variables (12a), $Q(P_6)$ of the previous stage can be applied as iteration matrix. An average of 2 iterations is sufficient for convergence. Moreover, the evaluation of $M(p^{(j)})$ and $G(p^{(j)})$ on the right-hand side of (13) can usually be skipped by taking $M(P_6)$ and $G(P_6)$ instead. For the projection of velocity variables (12b), the matrix $Q(P_7)$ must be evaluated and decomposed, but this does not enter the operation count since $Q(P_7)$ is also used for the evaluation of the last stage, which in turn forms the first stage of the next step. In this way, the additional cost for the whole projection step (12), on average, sums up to the solution of 3 linear systems with given matrix decompositions. Two linear systems can be saved if the above control mechanism is used. Accordingly, per step 6 evaluations of mass matrix M, constraint Jacobian G, forces f, and second derivative terms z are required and about 2 additional evaluations of the constraints g if (12a) is performed.

Finally, it is worth mentioning that special algorithms exploiting the structure have been developed for the decomposition of the symmetric indefinite matrix Q [3, 18]. In case of the wheel suspension, however, the linear algebra plays not the most important role. Standard solvers like DGETRF of LAPACK [2] are sufficient here.

4.3. Root finding

In order to locate efficiently zeros of switching functions, a continuous solution representation is required. For the Dormand and Prince method, the dense output formulas of [13] provide a fourth-order representation. However, in the algorithm above, the dense output is based on the formulation of index 1. Applying a standard root-finding procedure to this representation results in a solution which, again, may not satisfy the constraints. Hence, the projection procedure (12) comes one more time into play. As proposed in [18], the location of zeros is split into two phases:

(1) Monitor the signs of the switching functions using the standard dense output formulas. In case of a sign change, apply a few iterations of the root-finding algorithm to get a smaller enclosing interval.

(2) Continue the iteration process with projected dense output until the length of the enclosing interval is below the given tolerance.

Of course, the critical step is the switching from phase 1 to phase 2 where the enclosing interval may be lost, resulting in a stepsize reduction and repetition of the step. On the other hand, the application of the projection (12) to the dense output is expensive due to an additional matrix

decomposition for (12b) and thus the savings obtained by splitting the root finding can be substantial.

Remarks.

- The overall algorithm for computing consistent initial values, numerical integration, and root finding has been implemented as FORTRAN77 code MDOP5. It is via e-mail available to interested readers.
- MDOP5 is part of the program library MBSPACK — "MultiBody System PACKage" [25]. MBSPACK is divided into two groups. The algorithms of group 1 including MDOP5 are based on explicit or half-explicit discretization schemes while group 2 contains more general solvers with implicit discretization schemes. Since each method has its merits and drawbacks, this package provides a versatile tool for advanced simulation software.

5. Simulation results

Numerical results of the code MDOP5 when applied to the wheel suspension example are now presented. All experiments were run on an HP-Apollo DN4500 workstation in double precision arithmetic. Fig. 3 shows the simulation result for a first driving maneuver where the wheel passes a smooth bump with a speed of 30 m/s. Here no discontinuities appear. The oscillations are damped out by the shock absorber and the suspension returns to the initial configuration within

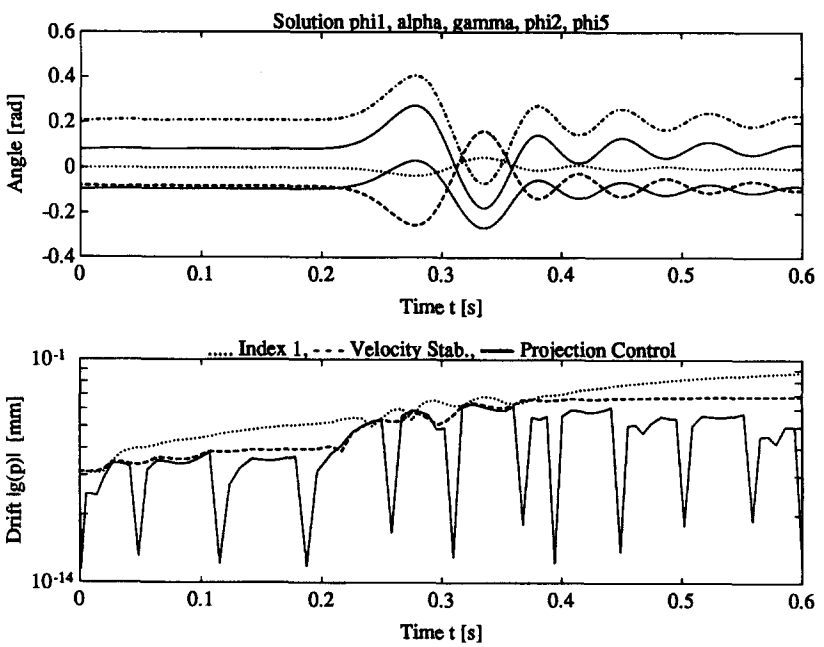

Fig. 3. Solution of wheel suspension for smooth bump and drift effect.

Table 1

	# Proj. (12a)	$\|g(p)\|_2$	$\|G(p)v\|_2$	Rel. error (p)	CPU (s)
Index 1	0	1.8E − 2	1.1E − 1	7.7E − 5	40.1
Velocity stabilization	0	7.4E − 4	8.2E − 12	4.0E − 5	40.8
Projection control	12	1.5E − 13	1.0E − 11	3.9E − 5	41.0
Projection each step	83	1.6E − 13	5.1E − 12	3.9E − 5	42.6

less than 1 s. In addition, Fig. 3 gives information on the projection algorithm (12) and the control (16).

With a prescribed tolerance of RTOL $= 10^{-4}$, four different versions were tested where in each case 83 integration steps were performed. Clearly, integrating the formulation of index 1 without stabilization results in substantial drift-off, as shown in the second diagram and Table 1. Applying only the velocity stabilization (12b) leads already to a significant improvement. The projection control (16) is as precise as projection in each step and requires only the solution of 12 nonlinear systems (12a) instead of 83. For the wheel suspension, the savings are not that important since the evaluation of the equations, in particular of the second derivative terms z, is the most expensive part [25].

MDOP5 was also compared with the code HEM5 of Brasey [5], a fifth-order half-explicit Runge–Kutta method which avoids the evaluation of z. In contrast to MDOP5, the arising systems of linear equations are not symmetric. Using the same tolerance RTOL and the same linear algebra method, HEM5 computed the solution in 49.2 s with a relative error of $1.7 \cdot 10^{-5}$. Though HEM5 saves about 50% computing time per function evaluation since z is not evaluated, these savings are neutralized by additional stages (8 instead of 6 for MDOP5) and larger error constants. Hence, both codes show comparable efficiency for this problem. But if the linear algebra becomes dominant, MDOP5 can be improved by exploiting the symmetry of matrix Q, which is not possible for HEM5. Further comparisons verifying the good performance of MDOP5 are given in [21].

For the second driving maneuver, the smooth driveway is replaced by a stair (Fig. 1) applying thus a discontinuous excitation on the wheel. Furthermore, with the resulting forces being much larger, another discontinuity of the tire force law has to be taken into account now. Due to the tire elasticity, the wheel loses contact with the driveway if the tire has been compressed very rapidly — a well-known phenomenon in vehicle dynamics, especially if the shock absorber is defect. For this reason, two switching functions have to be monitored during the numerical integration. The first describes the jump of the driveway and the second the different modes of the tire force law.

An animation sequence of the simulation result is shown in Fig. 4, where the wheel has been scaled in order to demonstrate the effects better. Passing the stair at $t_1 = 0.2$ s, the wheel loses contact and hits the driveway again at time $t_2 = 0.2342$. Then the tire is compressed, and at $t_3 = 0.2897$ the wheel bounces back and loses contact one more time until $t_4 = 0.3289$. Another bounce occurs between $t_5 = 0.3900$ and $t_6 = 0.4096$. All roots $t_1, \ldots, t_6$ were found by the above algorithm without any problems.

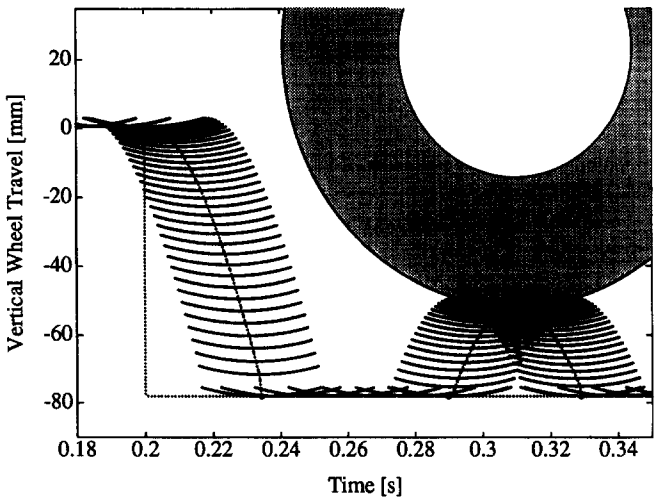

Fig. 4. Animation of wheel travel over stair.

Acknowledgements

The author thanks P. Rentrop for his great encouragement and G. Leister for generating and providing the equations of motion of the wheel suspension. He is indebted to the Stiftung Volkswagenwerk for their support in the project *Identifizierungs-, Entwurfs- und Analysemethoden für mechanische Mehrkörpersysteme in Deskriptorform.*

References

[1] T. Alishenas and Ö. Olafsson, Modeling and velocity stabilization of constrained mechanical systems, *BIT* **34** (1994) 455–483.

[2] R. Anderson et al., *LAPACK User's Guide* (SIAM, Philadelphia, 1992).

[3] T. Andrzejewski, H.G. Bock, E. Eich and R. von Schwerin, Recent advances in the numerical integration of multibody systems, in: W. Schiehlen, Ed., *Advanced Multibody System Dynamics* (Kluwer Academic Publishers, Stuttgart, 1993) 127–151.

[4] U. Ascher and L.R. Petzold, Stability of computational methods for constrained dynamics systems, *SIAM J. Sci. Statist. Comput.* **14** (1993) 95–120.

[5] V. Brasey, A half-explicit Runge–Kutta method of order 5 for solving constrained mechanical systems, *Computing* **48** (1992) 191–201.

[6] K.E. Brenan, S.L. Campbell and L.R. Petzold, *The Numerical Solution of Initial Value Problems in Ordinary Differential-Algebraic Equations* (North-Holland, Amsterdam, 1989).

[7] S.L. Campbell and C.W. Gear, High index differential-algebraic equations, Report, North Carolina State University, 1993.

[8] J.R. Dormand and P.J. Prince, A family of embedded Runge–Kutta formulae, *J. Comput. Appl. Math.* **6** (1980) 19–26.

[9] E. Eich, Projizierende Mehrschrittverfahren zur numerischen Lösung der Bewegungsgleichungen von Mehrkörpersystemen mit Zwangsbedingungen und Unstetigkeiten, Fortschritt-Berichte VDI, Series 18, No. 109 (VDI-Verlag, Düsseldorf, 1992).
[10] E. Eich, Convergence results for a coordinate projection method applied to mechanical systems with algebraic constraints, *SIAM J. Numer. Anal.* **30** (1993) 1467–1482.
[11] C. Führer and B.J. Leimkuhler, Numerical solution of differential-algebraic equations for constrained mechanical motion, *Numer. Math.* **59** (1991) 55–69.
[12] C.W. Gear, G.K. Gupta and B.J. Leimkuhler, Automatic integration of the Euler–Lagrange equations with constraints, *J. Comput. Appl. Math.* **12&13** (1985) 77–90.
[13] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations, I* (Springer, Berlin, 2nd ed., 1993).
[14] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations, II* (Springer, Berlin, 1991).
[15] E. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods* (Allyn and Bacon, Boston, 1989).
[16] M. Hiller and S. Frik, Road vehicle benchmark 2: five link suspension, in: W. Kortüm, S. Sharp and A. de Pater, Eds., *Application of Multibody Computer Codes to Vehicle System Dynamics*, Progress Report to the 12th IAVSD Symposium, Lyon, 1991.
[17] C. Lubich, Extrapolation integrators for constrained multibody systems, *Impact Comput. Sci. Eng.* **3** (1991) 213–234.
[18] Ch. Lubich, U. Nowak, U. Pöhle and Ch. Engstler, MEXX — Numerical software for the integration of constrained mechanical multibody systems, Report SC 92-12, ZIB, Berlin, 1992.
[19] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).
[20] F. Potra and W.C. Rheinboldt, Differential-geometric techniques for solving differential-algebraic equations, in: E. Haug and R. Deyo, Eds., *NATO Advanced Research Workshop on Real-Time Integration Methods for Mech. System Simulation*, Snowbird, Utah, 1989 (Springer, Berlin, 1990).
[21] W.C. Rheinboldt and B. Simeon, Performance analysis of some methods for solving Euler–Lagrange equations, *Appl. Math. Lett.* **8** (1995) 77–82.
[22] R.E. Roberson and R. Schwertassek, *Dynamics of Multibody Systems* (Springer, Berlin, 1988).
[23] W.O. Schiehlen, Ed., *Multibody System Handbook* (Springer, Berlin, 1990).
[24] L.F. Shampine, Conservation laws and the numerical solution of ODEs, *Comput. Math. Appl.* **12** (1986) 1287–1296.
[25] B. Simeon, Numerische Integration mechanischer Mehrkörpersysteme: Projizierende Deskriptorformen, Algorithmen und Rechenprogramme, Thesis, Fortschritt-Berichte VDI, Series 20, No. 130 (VDI-Verlag, Düsseldorf, 1994).
[26] B. Simeon, C. Führer and P. Rentrop, Differential algebraic equations in vehicle system dynamics, *Surv. Math. Ind.* **1** (1991) 1–37.
[27] B. Simeon, F. Grupp, C. Führer and P. Rentrop, A nonlinear truck model and its treatment as a multibody system, *J. Comput. Appl. Math.* **50** (1994) 523–532.
[28] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer, New York, 2nd ed., 1993).