A class of linearly-implicit Runge-Kutta methods for multibody systems

J. Wensch, K. Strehmel and R. Weiner

Abstract

We consider a special class of partitioned linearly-implicit Runge-Kutta methods for the solution of multibody systems in index 3 formulation. In contrast to implicit methods these methods require only the solution of linear systems for the algebraic variables. We study convergence and consistency of the methods and give numerical results for a special method of order 4 and comparisons.

Key words. partitioned linearly-implicit Runge-Kutta methods, differential-algebraic equations, multibody systems
AMS(MOS) subject classifications. 65L05

1 Introduction

Since the paper of Runge [22] the so called Runge-Kutta methods (RK methods) have been widely used for the numerical solution of initial value problems of ordinary differential equations. Until the sixties mainly explicit RK methods have been considered. However, the solution of stiff systems has shown that, due to the lack of A-stability ([7]), explicit methods are not suitable for this problem class. As a consequence the development of implicit methods became necessary. This work was done mainly by Butcher, who founded the theory of implicit RK methods, see [5]. With the growing interest in the solution of differential-algebraic equations (DAEs) implicit RK methods have been applied to these problems. Many papers are devoted to this subject, see [12], [14], [3]. Whereas the solution of DAEs of index 1 or 2 is possible with most standard methods for stiff equations, index 3 systems offer more difficulties. This is due to the fact, that perturbations of the solution are multiplied with a factor $1/h^2$, h the stepsize.
In this paper we consider a special class of partitioned linearly-implicit RK methods for the solution of multibody systems in their index 3 formulation. In contrast to implicit RK methods they require only the solution of linear systems of algebraic equations at every stage, furthermore by partitioning the dimension of these systems is reduced to the dimension of the algebraic variables.
In section 2 we consider linearly-implicit RK methods for stiff systems and discuss partitioned methods.

In section 3 we formulate the methods for the class of multibody systems of index 3 and give results on convergence and consistency by applying the theory of Butcher trees. We present simplifying assumptions and the number of order conditions up to order 4. We give numerical results of a particular method of order 4 and comparisons with other methods. Finally we make some remarks concerning further investigations.

2 Partitioned linearly-implicit Runge-Kutta methods

The numerical solution of stiff initial value problems

$$y' = f(y) \tag{2.1a}$$

$$y(t_0) = y_0 \in \mathbb{R}^{n_y} \tag{2.1b}$$

requires certain stability properties like A- or L-stability of the integration method.
The most stable methods can be found among the implicit Runge-Kutta methods (RK methods), what makes them very attractive. Unfortunately, the possibility to use large stepsizes is connected with a high effort per integration step. In general, an s-stage implicit Runge-Kutta method requires the solution of nonlinear equations of dimension sn_y. Therefore attention has been paid to diagonally- and singly-implicit RK methods, which require the solution of nonlinear equations of dimension n_y only. In contrast, linearly-implicit RK methods require only the solution of linear equations of dimension n_y. This is achieved by a direct incorporation of some approximation T of the Jacobian into the scheme. There are several classes of linearly-implicit RK methods, we will consider here ROW- and W-methods. They can be interpreted as the result of a diagonally-implicit RK method, when only one Newton-step with special starting value is performed at every stage. For example, consider the one-stage RK method

$$\begin{aligned} Y_1' &= f(y_n + h\gamma Y_1') \\ y_{n+1} &= y_n + hY_1' \end{aligned}$$

with Butcher-scheme

$$\begin{array}{c|c} \gamma & \gamma \\ \hline & 1 \end{array}.$$

One Newton-step with the matrix $T \approx f_y(y_n)$ and starting value $Y_1'^{(0)} = 0$ results in

$$\begin{aligned} (I - h\gamma T)Y_1' &= f(y_n) \quad \text{resp.} \quad Y_1' = f(y_n) + h\gamma T Y_1' \\ y_{n+1} &= y_n + hY_1' \end{aligned} \tag{2.2}$$

Analogously we obtain for an s-stage method with coefficients α_{ij}, γ_{ij}

$$\begin{aligned} Y_i &= y_n + h\sum_{j=1}^{i-1} \alpha_{ij} Y_j', \quad i = 1, \ldots, s+1, \\ y_{n+1} &= Y_{s+1} \\ Y_i' &= f(Y_i) + hT\sum_{j=1}^{i} \gamma_{ij} Y_j', \quad \gamma_{ii} = \gamma, \quad i = 1, \ldots, s. \end{aligned} \tag{2.3}$$

At each stage we must solve a linear system with the same coefficient matrix $I - h\gamma T$, only one LU-decomposition per integration step is necessary. For $T = f_y(y_n)$ we have a ROW-method (e.g. [15]), with arbitrary T the methods are called W-methods ([24]). Note, that for $T = 0$ the W-method reduces to an explicit RK method, for instance (2.2) reduces to the explicit Euler method.
For W-methods the number of order conditions increases strongly, because the Butcher series contain additional differentials associated with the matrix T. So it is in general difficult to construct W-methods of high order. On the other hand, W-methods allow to keep the Jacobian fixed for several integration steps and are well suited for the construction of so called *partitioned* methods. Partitioned methods are designed for systems where only a (small) part of the whole system is stiff. Consider for simplicity the coupled system

$$\begin{aligned} y' &= f(y,z), \quad y \in \mathbb{R}^{n_y} \\ z' &= g(y,z), \quad z \in \mathbb{R}^{n_z}, \end{aligned}$$

where only the second subsystem is stiff, i.e. $\|f_y\|, \|f_z\| = \mathcal{O}(1)$, $\|g_z\| \gg 1$. The choice

$$T = \begin{pmatrix} 0 & 0 \\ 0 & g_z(y_n, z_n) \end{pmatrix}$$

yields a partitioned W-method. It requires only the solution of equations of dimension n_z, the nonstiff part is integrated by an explicit RK method. For partitioned systems with correct arrangement of the stiff and nonstiff parts partitioned linearly-implicit RK methods are therefore very efficient ([26]).
An additional attractive feature is that via singularly perturbed systems partitioned linearly-implicit RK-methods for semi-explicit differential-algebraic equations can be derived, e.g. [19], [20], [25] for index-1-systems. In section 3 we will extend these investigations to the class of multibody systems (index 3).
There exist A- and L-stable W-methods. The stability function of (2.3) is given by ([14])

$$R(z) = 1 + z\alpha_{s+1}^T (I - z\beta)^{-1} \mathbb{1},$$

with

$$\alpha_{s+1}^T = (\alpha_{s+1,1}, \ldots, \alpha_{s+1,s}), \quad \beta = (\beta_{ij})_{i,j=1}^s, \quad \beta_{ij} = \alpha_{ij} + \gamma_{ij}.$$

Especially in view of differential-algebraic equations the value

$$R(\infty) = 1 - \alpha_{s+1}^T \beta^{-1} \mathbb{1} = 1 - \sum_{i,j} \alpha_{s+1,i}\omega_{ij}, \tag{2.4}$$

is of interest, where ω_{ij} are the elements of β^{-1}.

3 W-Methods for Multibody-Systems

3.1 Multibody-Systems

Consider the Lagrange-equations for the motion of rigid bodies with position coordinates x under holonomic constraints g, generalized forces f, symmetric mass matrix M and Lagrange multipliers λ:

$$\begin{aligned} 0 =& M(x)x'' - f(x,x') + G^T(x)\lambda \\ 0 =& g(x) \qquad \text{with} \quad g:\mathbb{R}^{n_x} \to \mathbb{R}^{n_g} \end{aligned} \tag{3.1}$$

The matrix G is the Jacobian of the constraints g. The descriptor form (3.1) is a differential algebraic system, consisting of differential and algebraic equations.
Differential algebraic systems are characterized by their index. Up to now there are many index concepts, cf. [10], [6]. The most common are the differentiation index [9] and the perturbation index [12]. Roughly speaking, the differentiation index is the necessary number di of differentiations of the original system so, that the resulting $di+1$ systems may be solved with respect to the derivatives of all incorporated variables, resulting in a system of ordinary differential equations. The perturbation index pi characterizes the behaviour of the solution with respect to perturbations of the system. The higher the perturbation index, the higher the order of derivatives of the perturbation that appear in the solution of the perturbed system. This makes the numerical solution of higher index DAEs so complicated.
The equations of motion form a differential-algebraic system of index 3 under the assumption that the so-called optimization matrix

$$\begin{pmatrix} M(x) & G^T(x) \\ G(x) & 0 \end{pmatrix} \tag{3.2}$$

is regular in a neighbourhood of the exact solution.
The effective numerical solution of systems of index 3 is an open question up to now. There have been made several proposals how to reduce the index of the system. One obvious way is to differentiate the constraints once – to get an index 2 system – or twice – to get a system of index one. A drawback is that the numerical solution will not fulfil the original constraints, it drifts off. To avoid this drift, projections on the original manifold may be done. The index 2 formulation may be solved by direct application of one-step or multistep codes, e.g. RADAU5 [14], DASSL [3], or by the extrapolation code MEXX [16]. Very efficient are the half-explicit Runge-Kutta methods of Brasey and Hairer, e.g. the code HEM5 [2]. Other ways are the incorporation of the derivatives of the constraints and the original constraints both in a new formulation of the problem. This is done in the Gear/Gupta/Leimkuhler stabilization (reduction to index 2) [11] and in the projecting descriptor form (reduction to index 1). The first case is implemented in ODASSL [8], the second case has been investigated in [23]. We mention, that standard methods for ODEs may be applied without problems to index 1 systems.
There are only a few methods that can be applied directly to index 3 systems. Among them are the implicit Runge-Kutta methods, e.g. the code RADAU5. Ostermann [18] developed half-explicit methods, which only treat the algebraic variables implicitly. In contrast to the methods of the following subsections they require the solution of nonlinear equations.

3.2 Definition of the Method

The application of linearly-implicit methods to differential algebraic systems of index 1 and 2 has been investigated in several papers, cf. [1], [17], [21], [25]. We will consider the

application of W-methods to the index 3 system (3.1).
The system (3.1) is written as a singularly perturbed system of first order by introducing velocities and accelerations and artificial derivatives (direct approach):

$$\begin{aligned} x' &= v \\ v' &= w \\ \varepsilon w' &= M(x)w - f(x,v) + G^T(x)\lambda \\ \varepsilon \lambda' &= g(x) \end{aligned}$$

To this system the W-method (2.3) is applied. We then put $\varepsilon = 0$ and obtain:

$$\begin{aligned} X_i &= x_n + h\sum_{j=1}^{i-1} \alpha_{ij} X'_j & V_i &= v_n + h\sum_{j=1}^{i-1} \alpha_{ij} V'_j \\ W_i &= w_n + h\sum_{j=1}^{i-1} \alpha_{ij} W'_j & \Lambda_i &= \lambda_n + h\sum_{j=1}^{i-1} \alpha_{ij} \Lambda'_j \end{aligned}$$

$$\begin{pmatrix} X'_i \\ V'_i \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} V_i \\ W_i \\ M(X_i)W_i - f(X_i, V_i) + G^T(X_i)\Lambda_i \\ g(X_i) \end{pmatrix} + hT\sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} X'_j \\ V'_j \\ W'_j \\ \Lambda'_j \end{pmatrix}$$

$$x_{n+1} = X_{s+1} \qquad v_{n+1} = V_{s+1} \qquad w_{n+1} = W_{s+1} \qquad \lambda_{n+1} = \Lambda_{s+1}. \tag{3.3}$$

In the approximation T of the Jacobian the blocks T_{31} and T_{32} are set to zero:

$$T = \begin{pmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & M(x_n) & G^T(x_n) \\ G(x_n) & 0 & 0 & 0 \end{pmatrix} \tag{3.4}$$

The approximation T contains only those blocks that are necessary to guarantee a stable (convergent) solution. In difference to partitioned methods for stiff problems, were the blocks corresponding to the dynamics of the system are included, here the blocks corresponding to the index 3 chain $x, v, (w, \lambda)$ are included.
We have to solve a linear system with the matrix

$$\begin{pmatrix} I & -h\gamma I & 0 & 0 \\ 0 & I & -h\gamma I & 0 \\ 0 & 0 & -h\gamma M(x_n) & -h\gamma G^T(x_n) \\ -h\gamma G(x_n) & 0 & 0 & 0 \end{pmatrix}$$

at each stage. This results via an obvious transformation in the solution of s systems with the optimization matrix (3.2) of dimension $n_x + n_g$ for the unknowns W_i', Λ_i'.
A further partitioning is done by allowing different method coefficients α_{ij}, γ_{ij} in the different components x, v, w, λ of the solution. We obtain the following W-method:

$$\begin{aligned}
X_i &= x_n + h \sum_{j=1}^{i-1} \alpha_{ij}^{(x)} X_j' & V_i &= v_n + h \sum_{j=1}^{i-1} \alpha_{ij}^{(v)} V_j' \\
W_i &= w_n + h \sum_{j=1}^{i-1} \alpha_{ij}^{(w)} W_j' & \Lambda_i &= \lambda_n + h \sum_{j=1}^{i-1} \alpha_{ij}^{(w)} \Lambda_j' \\
X_i' &= V_i + h \sum_{j=1}^{i} \gamma_{ij}^{(v)} V_j' & V_i' &= W_i + h \sum_{j=1}^{i} \gamma_{ij}^{(w)} W_j' \\
0 &= M(X_i) W_i - f(X_i, V_i) + G^T(X_i)\Lambda_i \\
&\quad + hM(x_n) \sum_{j=1}^{i} \gamma_{ij}^{(w)} W_j' + hG^T(x_n) \sum_{j=1}^{i} \gamma_{ij}^{(w)} \Lambda_j' \\
0 &= g(X_i) + hG(x_n) \sum_{j=1}^{i} \gamma_{ij}^{(x)} X_j'
\end{aligned} \tag{3.5}$$

We have chosen the same α_{ij} for W and Λ and the same γ_{ij} for W' and Λ'. This reduces the consistency conditions without loosing any degree of freedom. Further, we have chosen the γ_{ij} for V' and for W', Λ' to be the same. This gives a slightly better convergence result.

3.3 Convergence of the Method and Order Conditions

We assume that the mass matrix M is regular in a neighbourhood of the exact solution. Let (cf. (2.4))

$$\begin{aligned}
R_*(\infty) &= 1 - \sum_{j,k} \alpha_{s+1,j}^{(*)} \omega_{jk}^{(*)} \qquad \text{with} \quad * \in \{x, v, w\} \\
Q &= M^{-1}G^T(GM^{-1}G^T)^{-1}G, \qquad P = I - Q.
\end{aligned}$$

Theorem 3.1 *The W-method (3.5) is applied to the problem (3.1) with consistent initial values. Further assume that*

$$|R_x(\infty)|, |R_v(\infty)|, |R_w(\infty)| < 1.$$

If the local error satisfies

$$\begin{aligned}
le_{x,h} &= \mathcal{O}\left(h^{p+2}\right) \\
Qle_{v,h} &= \mathcal{O}\left(h^{p+1}\right) \qquad & Ple_{v,h} &= \mathcal{O}\left(h^{p+2}\right) \\
le_{w,h} &= \mathcal{O}\left(h^{p}\right) \\
le_{\lambda,h} &= \mathcal{O}\left(h^{p}\right),
\end{aligned}$$

then the global error is

$$
\begin{aligned}
Qe_{x,h} =& \mathcal{O}\left(h^{p+2}\right) & Pe_{x,h} =& \mathcal{O}\left(h^{p+1}\right) \\
e_{v,h} =& \mathcal{O}\left(h^{p+1}\right) & & \\
e_{w,h} =& \mathcal{O}\left(h^{p}\right) & & \\
e_{\lambda,h} =& \mathcal{O}\left(h^{p}\right). & &
\end{aligned}
\tag{3.6}
$$

Proof. The theorem is proved by using "Lady Windermere's fan" ([13]). We estimate the partial Lipschitz constants of the components $Px, Pv, Pw, Qx, Qv, Qw, \lambda$. The estimation of the eigenvalues of the resulting 7-by-7 matrix gives a Lipschitz constant for the method and the convergence result. The details of the proof can be found in [27]. ■

Remark 3.1 *Because in general only the components x and v are of interest (Px and Pv give the dynamics of the system) we say that with (3.6) the method is of order $p+1$.*

We mention, that the theorem exploits the special structure of partitioning in the coefficients of the method.
To study the order of the local error the expansions of the numerical and the analytical solution in t_n have to be compared. For the total time derivatives of the analytical solution we have:

$$
\begin{aligned}
x^{(1)} =& v & & \\
x^{(2)} =& w & v^{(1)} =& w
\end{aligned}
$$

The derivatives of higher order satisfy the following recursions ($m \geq 1$):

$$
\begin{aligned}
Px^{(m+2)} =& Pw^{(m)} & Qx^{(m+2)} =& (-M^{-1})G^T J^{(m+2)} \\
Pv^{(m+1)} =& Pw^{(m)} & Qv^{(m+1)} =& Qx^{(m+2)} \\
Pw^{(m)} =& PI^{(m)} & Qw^{(m)} =& Qx^{(m+2)} \\
\lambda^{(m)} =& SI^{(m)} + J^{(m+2)} & &
\end{aligned}
$$

where we have used the abbreviations

$$
\begin{aligned}
I^{(m)} :=& (-M^{-1})[(Mw - f(x,v) + G^T\lambda)^{(m)} - Mw^{(m)} - G^T\lambda^{(m)}] \\
J^{(m)} :=& (GM^{-1}G^T)^{-1}[g(x)^{(m)} - Gx^{(m)}] \\
S :=& (GM^{-1}G^T)^{-1}G.
\end{aligned}
$$

The expression $J^{(m+2)}$ contains derivatives of x up to order $m+1$ ($x^{(m+2)}$ cancels), $I^{(m)}$ derivatives of x and v up to order m and of w and λ up to order $m-1$. This allows to express the derivatives of higher order recursively by those of lower order.
The values Λ_j' and W_j' in (3.5) are uniquely defined for $h > 0$ only. In the following we denote therefore by an upper subscript (m) the limit of the m-th derivative with respect to

h for $h \to 0$. From (3.5) we obtain immediately

$$\begin{aligned} X_i^{(0)} &= x \\ X_i^{(1)} &= \alpha_i^{(x)} v & \qquad V_i^{(0)} &= v \end{aligned}$$

with $\alpha_i^{(x)} = \sum_j \alpha_{ij}^{(x)}$. By differentiating the last equation in (3.5) twice we get

$$0 = g_{xx}(X_i^{(1)}, X_i^{(1)}) + GX_i^{(2)} + G\sum_{j=1}^{i} \gamma_{ij}^{(x)} (hX_j')^{(2)}.$$

With

$$g_{xx}(v,v) + Gw = 0 \quad \text{and} \quad X_i^{(2)} = \sum_j \alpha_{ij}^{(x)} (hX_j')^{(2)}$$

follows

$$Q(hX_i')^{(2)} = \sum_j \omega_{ij}^{(x)} (\alpha_j^{(x)})^2 Qw.$$

In similar manner we get

$$P(hW_i')^{(0)} = 0.$$

Then we can compute

$$\begin{aligned} X_i^{(2)} &= 2\sum_{j,k} \alpha_{ij}^{(x)} \beta_{jk}^{(v)} w + 2\sum_{j,k} \alpha_{ij}^{(x)} \beta_{jk}^{(v)} \Delta_k Qw \\ V_i^{(1)} &= \alpha_i^{(v)} w + \sum_j \alpha_{ij}^{(v)} \Delta_j Qw \\ W_i^{(0)} &= w + \sum_{i,j} \alpha_{ij}^{(w)} \omega_{jk}^{(w)} \Delta_k Qw \\ \Lambda_i^{(0)} &= \lambda + \sum_{j,k} \alpha_{ij}^{(w)} \omega_{jk}^{(w)} \Delta_k (GM^{-1}G^T)^{-1} Gw \end{aligned}$$

where $\Delta_i = 1/2 \sum_{j,k} \omega_{ij}^{(v)} \omega_{jk}^{(x)} (\alpha_k^{(x)})^2 - 1$. For details we refer to [27].
We see, that the internal stages W_i, Λ_i are not necessarily consistent. This phenomenon is observed for all methods of diagonally implicit type, when applied directly to index 3 systems [12], [18]. Here this may be avoided to some extent by the partitioning of the coefficients. The derivatives of higher order satisfy the following recursions:

$$\begin{aligned} PX_i^{(m+2)} &= (m+1)(m+2) \sum_{j,k} \alpha_{ij}^{(x)} \beta_{jk}^{(v)} PI_k^{(m)} \\ PV_i^{(m+1)} &= (m+1) \sum_j \alpha_{ij}^{(v)} PI_j^{(m)} \\ PW_i^{(m)} &= \sum_{j,k} \alpha_{ij}^{(w)} \omega_{jk}^{(w)} PI_k^{(m)} \\ QX_i^{(m+2)} &= \sum_{j,k} \alpha_{ij}^{(x)} \omega_{jk}^{(x)} (-M^{-1}) G^T J_k^{(m+2)} \end{aligned}$$

$$
\begin{aligned}
QV_i^{(m+1)} =& \frac{1}{n+2}\sum_{j,k,l}\alpha_{ij}^{(v)}\omega_{jk}^{(v)}\omega_{kl}^{(x)}(-M^{-1})G^T J_l^{(m+2)} \\
QW_i^{(m)} =& \frac{1}{(m+1)(m+2)}\sum_{j,k,l,l_1}\alpha_{ij}^{(w)}\omega_{jk}^{(w)}\omega_{kl}^{(v)}\omega_{ll_1}^{(x)}(-M^{-1})G^T J_{l_1}^{(m+2)} \\
\Lambda_i^{(m)} =& \sum_{j,k}\alpha_{ij}^{(w)}\omega_{jk}^{(w)}SI_k^{(m)} \\
&+\frac{1}{(m+1)(m+2)}\sum_{j,k,l,l_1}\alpha_{ij}^{(w)}\omega_{jk}^{(w)}\omega_{kl}^{(v)}\omega_{ll_1}^{(x)}J_{l_1}^{(m+2)}
\end{aligned}
$$

with the abbreviations

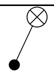
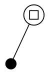
$$
\begin{aligned}
I_i^{(m)} :=& (-M^{-1})[(M(X_i)W_i - f(X_i,V_i) + G^T(X_i)\Lambda_i)^{(m)} - M(x)W_i^{(m)} - G^T(x)\Lambda_i^{(m)}] \\
J_i^{(m)} :=& (GM^{-1}G^T)^{-1}[g(X_i)^{(m)} - GX_i^{(m)}].
\end{aligned}
$$

To represent the differentials occuring in the derivatives of the analytical and the numerical solution we choose generalized Butcher trees [4]. The following table gives an overview on the nodes and the differentials represented by them.

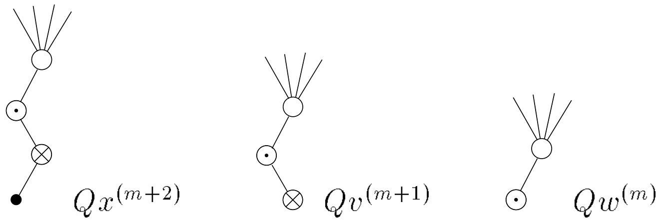
node	represented derivative	represented differential
$\bullet$	x	$\frac{\partial^{n_v}}{(\partial v)^{n_v}}v$
$\otimes$	v	$\frac{\partial^{n_w}}{(\partial w)^{n_w}}w$
$\odot$	w	$(-M^{-1})\frac{\partial^{n_x+n_v+n_w+n_\lambda}}{(\partial x)^{n_x}(\partial v)^{n_v}(\partial w)^{n_w}(\partial\lambda)^{n_\lambda}}[Mw-f+G^T\lambda]$
$\circledcirc$	w	$P(-M^{-1})\frac{\partial^{n_x+n_v+n_w+n_\lambda}}{(\partial x)^{n_x}(\partial v)^{n_v}(\partial w)^{n_w}(\partial\lambda)^{n_\lambda}}[Mw-f+G^T\lambda]$
$\bigcirc$	λ	$(GM^{-1}G^T)\frac{\partial^{n_x+n_v+n_w+n_\lambda}}{(\partial x)^{n_x}(\partial v)^{n_v}(\partial w)^{n_w}(\partial\lambda)^{n_\lambda}}[g(x)]$
⧇	v	Qw
◉	$w^{(0)}$	Qw

The last tree is used to represent the inconsistency in the components W_i.
By use of the trees we are able to formulate the consistency conditions. We start with the derivatives of lower order. Our special coefficient partitioning allows us to drop the consistency conditions for λ, because the coefficients, occuring in the expansion of the Λ_i are the same as the coefficients occuring in the expansion of the W_i, although the corresponding differentials are different. Our consistency conditions are given in such a way, that the coefficients for the differentials have to be equal one, except in the case, when the tree contains a node ⧇ or ◉. These coefficients have to vanish. The coefficients are given by

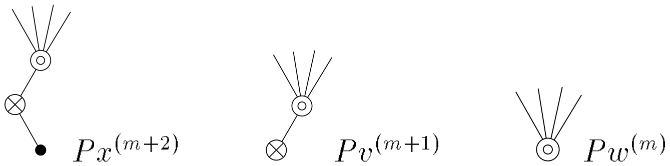
using the following sum convention: the summation goes over all indices except i.

derivative	tree	coefficient
$X_i^{(1)}$	•	$\alpha_i^{(x)}$
$X_i^{(2)}$		$2\alpha_{ij}^{(x)}\beta_{jk}^{(v)}$
		$2\alpha_{ij}^{(x)}\beta_{jk}^{(v)}\Delta_k$
$V_i^{(1)}$	$\otimes$	$\alpha_i^{(v)}$
	⧇	$\alpha_{ij}^{(v)}\Delta_j$
$W_i^{(0)}$	–	1
	◙	$\alpha_{ij}^{(w)}\omega_{jk}^{(w)}\Delta_k$

For the derivatives of higher order we have to represent the expressions $(-M^{-1})G^T J_k^{(m+2)}$ and $PI_k^{(m)}$. We get the first expression by the trees of type



with k_x branches leaving to subtrees, representing derivatives of x, where $k_x \geq 2$. The sum of the orders of these k_x trees has to be equal to $m+2$. The second expression is represented by trees of type



with k_x, k_v, k_w branches leaving to subtrees, representing derivatives of x, v, w. The sum of the orders of the trees has to be equal to m. The definition of $I_k^{(m)}$ implies, that either $k_w = 0$ or $(k_w = 1 \wedge k_v = 0 \wedge k_x > 0)$. In the case of $k_w = 1$ the subtree representing a derivative of w, may be the tree ◙ of order zero, i.e. the tree representing the inconsistency in $W_i^{(0)}$.

3.4 Construction of Partitioned W-Methods

Higher order implicit Runge-Kutta methods have been constructed by the use of simplifying assumptions. These simplifying assumptions on the coefficients of the method reduce the number of consistency conditions. However, in the case of methods of explicit or diagonally implicit type it is much more difficult to find suitable simplifying assumptions.
In Table 3.1 we first present the set of order conditions for a method of order 2. These

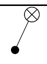
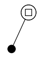
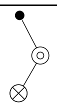
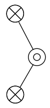
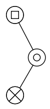
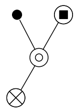
derivative	tree	order condition
$x^{(1)}$	$\bullet$	$\sum \alpha^{(x)}_{s+1,i} = 1$
$x^{(2)}$		$2\sum \alpha^{(x)}_{s+1,i}\beta^{(v)}_{ij} = 1$
		$\sum \alpha^{(x)}_{s+1,i}\beta^{(v)}_{ij}\Delta_j = 0$
$v^{(1)}$	$\otimes$	$\sum \alpha^{(v)}_{s+1,i} = 1$
		$\sum \alpha^{(v)}_{s+1,i}\Delta_i = 0$
$w^{(0)}$		$\sum \alpha^{(w)}_{s+1,i}\omega^{(w)}_{ij}\Delta_j = 0$
$Pv^{(2)}$		$2\sum \alpha^{(v)}_{s+1,i}\alpha^{(x)}_{ij} = 1$
		$2\sum \alpha^{(v)}_{s+1,i}\alpha^{(v)}_{ij} = 1$
		$\sum \alpha^{(v)}_{s+1,i}\alpha^{(v)}_{ij}\Delta_j = 0$
		$\sum \alpha^{(v)}_{s+1,i}\alpha^{(x)}_{ij}\alpha^{(w)}_{ik}\omega^{(w)}_{kl}\Delta_l = 0$

Table 3.1: Consistency conditions for order 2.

conditions allow the construction of a method of order 2 with 3 stages. There exists no method of order 2 with 2 stages because conditions 8 and 9 in Table 3.1 are contradictory. For higher order the number of conditions grows rapidly. We reduce the number of conditions by simplifying assumptions. At first we demand the values W_i to be consistent, i.e.,

$$\sum \alpha^{(w)}_{ij}\omega^{(w)}_{jk}\Delta_k = 0 \quad i = 1, \ldots, s.$$

This gives set (I) of simplifying assumptions. The second set (II) consists of (I) and additionally

$$\begin{aligned} \sum \alpha^{(x)}_{ij}\beta^{(v)}_{jk}\Delta_k &= 0 \quad i = 1, \ldots, s \\ \sum \alpha^{(v)}_{ij}\Delta_j &= 0 \quad i = 1, \ldots, s. \end{aligned}$$

Note that the number of simplifying conditions depends on the stage number s. Case (I) gives $s-1$ additional conditions, case (II) gives $3(s-1)$ additional conditions. We display the corresponding number of consistency conditions in Table 3.2. We have derived two methods, one of order 3 with 5 stages and one of order 4 with 11 stages using the simplifying assumptions (I).

	used simplifying assumptions		
order	none	(I)	(II)
2	10	9+s–1	8+3(s–1)
3	52	44+s–1	30+3(s–1)
4	292	242+s–1	139+3(s–1)
5	1743	1405+s–1	707+3(s–1)

Table 3.2: Number of order conditions

Unfortunately, the solution of the equations with symbolic computation programs failed. Neither MAPLE nor MATHEMATICA were able to solve the equations for orders greater than two. We used the groebner-package of MAPLE to get at least some suggestions for other sets of simplifying assumptions or to detect dependencies in the conditions, but these attempts failed, too.
Our sets of coefficients have been computed by MATLAB solving the corresponding least squares problem. After long computation we achieved absolute errors less than $1.E-13$ in the consistency conditions.

3.5 Numerical test and comparison

We present some test results for a partitioned W-method (3.5) of order 4 with 11 stages. Because of their large number we do not give the parameters of the method here. Details can be found in [27].
First we tested the order of the method with constant stepsize at the pendulum problem [12]. Figure 3.1 shows the achieved accuracy in the components x, v, w and λ versus the stepsize. The components are denoted by the following symbols

$$+\ldots v, \quad \times\ldots x, \quad \circ\ldots w, \quad *\ldots\lambda.$$

Figure 3.1 clearly shows the predicted orders of 3 for w and λ and 4 for v. The order of x turns out to be 5 instead of 4 as stated by Theorem 3.1. The reason for this is the simple structure of the problem.
Then we tested the method with stepsize control. The stepsize selection is performed by local error estimation for the x components by an embedded method of order 3. As test example we used "Andrew's squeezing mechanism", which frequently serves as test problem for DAE-codes. It consists of 7 rigid bodies connected by joints without friction in plane motion. The equations can be found e.g. in [14]. We applied our method to the index 3

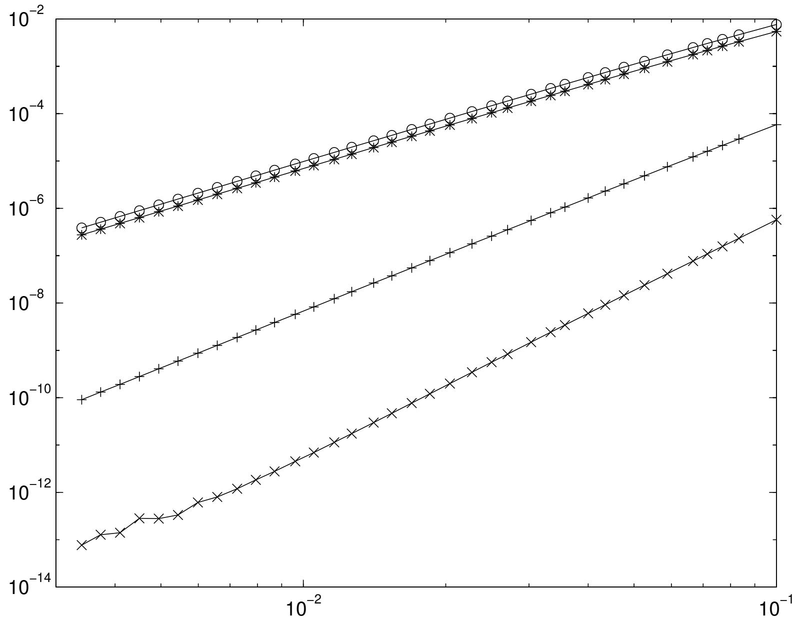


Figure 3.1: Order test for the pendulum problem

formulation in the form

$$
\begin{aligned}
x' &= v \\
v' &= w \\
0 &= M(x)w - f(x,v) + G^T(x)\lambda \\
0 &= g(x)
\end{aligned}
$$

With $x \in \mathbb{R}^7$, $\lambda \in \mathbb{R}^6$ the dimension of the DAE is 27, by the special choice of T (cf. section 3.2) in the partitioned W-method at every stage a linear system of dimension 13 has to be solved.
We solved this problem for different tolerances in the range $10^{-2} - 10^{-10}$. For comparison we applied the half-explicit RK methods HEM4 and HEM5 [2] without additional projection, which use the index 2 formulation, and the implicit RK method RADAU5 [14] applied to both the index 3 and index 2 formulation. Figure 3.2 and Figure 3.3 show the computing time versus the achieved accuracy in the components Pe_x and Qe_x in double logarithmic scale. The results for v are nearly identical to those for Pe_x. The codes are represented by the following symbols

$+$	HEM4
$\times$	HEM5
$\circ$	RADAU5 for index 2
$*$	RADAU5 for index 3
$-$	11-stage partitioned W method

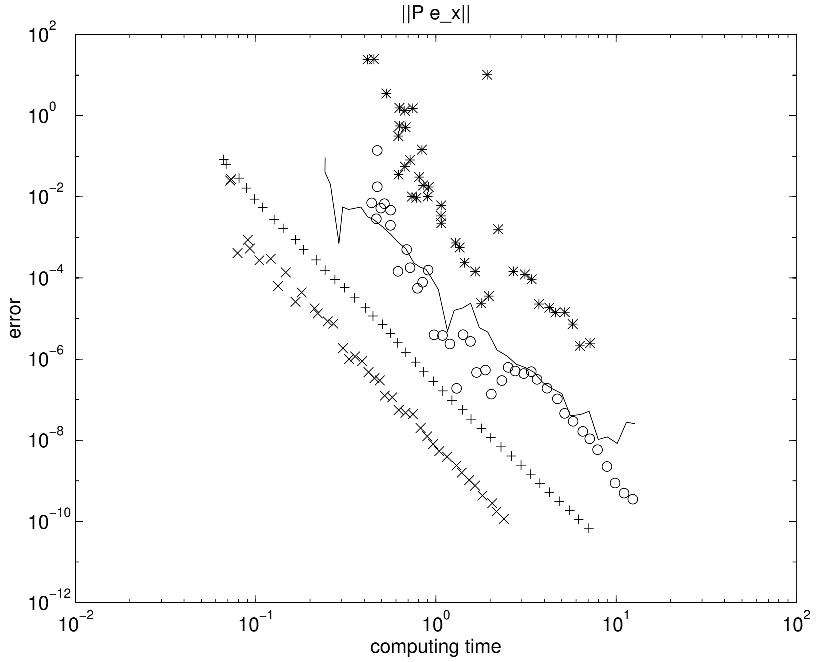


Figure 3.2: Euclidean norm of the global error Pe_x

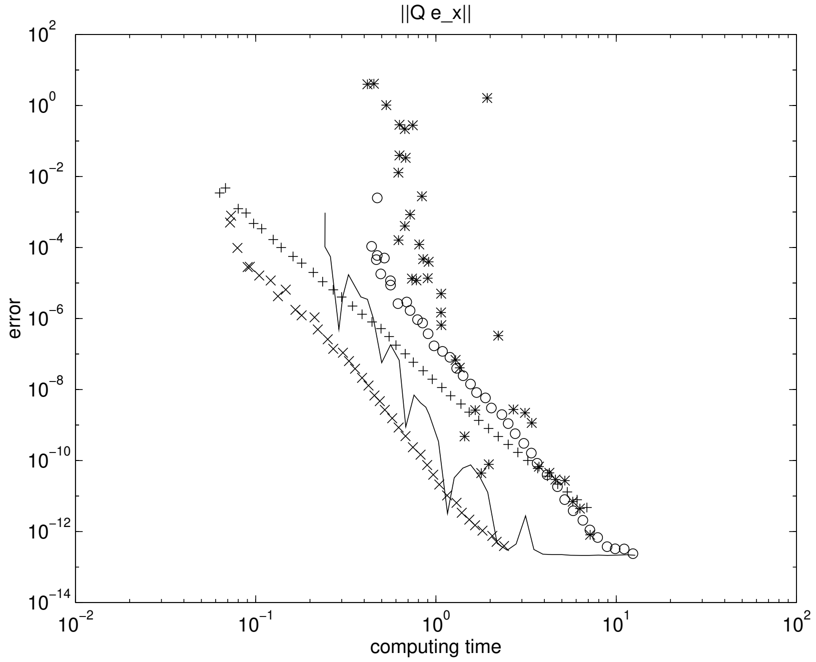


Figure 3.3: Euclidean norm of the global error Qe_x

The figures confirm the order 4 of the partitioned W-method for Pe_x and order 5 for Qe_x. The W-method is more efficient than RADAU5 for the index 3 system and comparable to RADAU5 for index 2. The most efficient codes for this problem are HEM4 and HEM5 which use the index 2 formulation.

4 Concluding remarks

In our opinion this work is a first attempt to apply linearly-implicit RK methods to multibody systems in index 3 formulation. Our results suggest, that these methods can be used successfully to solve mechanical multibody systems directly without transforming to an index 2 system. The methods require only the solution of linear equations of dimension $n_x + n_g$, the computation of g_{xx} is not necessary.
On the other hand, due to the large number of consistency conditions the construction of higher order methods is a difficult task. So we found after long computations a particular solution for the conditions of order 4 with 11 stages, which is surely not optimal. We suppose that by use of simplifying assumptions like (II) order 4 can be achieved with a smaller number of stages, probably 8. Furthermore we did not yet try to minimize the error constants of the method. This will be the topic of further studies and will make the method more efficient. Another possibility is the adaptation of the method to stiff multibody systems by exploiting the blocks T_{31} and T_{32} in the matrix T in (3.4).

References

[1] M. Arnold, K. Strehmel, and R. Weiner, *Half-explicit Runge-Kutta methods for semi-explicit differential-algebraic equations of index 1*, Numer. Math. **64** (1993), 409–431.

[2] V. Brasey and E. Hairer, *Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2*, SIAM J. Numer. Anal. **30** (1993), 538–552.

[3] K.E. Brenan, S.L. Campbell, and L.R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*, North-Holland Publ. Co., Amsterdam, 1989.

[4] J.C. Butcher, *Coefficients for the study of Runge-Kutta integration processes*, J. Austral. Math. Soc. **3** (1963), 185–201.

[5] ______, *The numerical analysis of ordinary differential equations*, John Wiley & Sons, 1987.

[6] S.L. Campbell and C.W. Gear, *The index of general nonlinear DAEs*, Numer. Math. **72** (1995), 173–196.

[7] G. Dahlquist, *A special stability problem for linear multistep methods*, BIT **3** (1963), 27–43.

[8] C. Führer, *Differential-Algebraische Gleichungssysteme in Mechanischen Mehrkörpersystemen*, Ph.D. thesis, TU München, 1988.

[9] C.W. Gear, *Differential-algebraic index transformations*, SIAM J. Sci. Stat. Comp. **9** (1988), 39–47.

[10] ———, *Differential-algebraic equations, indices, and integral algebraic equations*, SIAM J. Numer. Anal. **27** (1990), 1527–1534.

[11] C.W. Gear, B. Leimkuhler, and G.K. Gupta, *Automatic integration of Euler-Lagrange equations with constraints*, J. Comp. Appl. Math. **12 & 13** (1985), 77–90.

[12] E. Hairer, Ch. Lubich, and M. Roche, *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Lecture Notes in Mathematics **1409** (1989), Springer-Verlag.

[13] E. Hairer, S.P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, 2. ed., Springer-Verlag, 1993.

[14] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer-Verlag, 1991.

[15] P. Kaps and P. Rentrop, *Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations*, Numer. Math. **38** (1979), 55–68.

[16] Ch. Lubich, *h^2-extrapolation methods for differential-algebraic systems of index 2*, Impact Comp. Sci. Eng. **1** (1989), 260–268.

[17] Ch. Lubich and M. Roche, *Rosenbrock methods for differential algebraic systems with solution-dependent singular matrix multiplying the derivative*, Computing **43** (1990), 325–342.

[18] A. Ostermann, *A class of half-explicit Runge-Kutta methods for differential-algebraic systems of index 3*, Appl. Numer. Math. **13** (1993), 165–179.

[19] P. Rentrop, M. Roche, and G. Steinebach, *The application of Rosenbrock-Wanner type methods with stepsize control in differential-algebraic equations*, Numer. Math. **55** (1989), 545–563.

[20] M. Roche, *Rosenbrock methods for differential-algebraic systems*, Numer. Math. **52** (1988), 45–63.

[21] ———, *Runge-Kutta and Rosenbrock methods for differential-algebraic equations and stiff ODEs*, Université de Geneve, Dept. de math., These Nr. 2331 (1988).

[22] C. Runge, *Über die numerische Auflösung von Differentialgleichungen*, Math. Ann. **46** (1895), 167–178.

[23] B. Simeon, *MBSPACK – numerical integration software for constrained mechanical motion*, Surv. Math. Ind. **5** (1995), 169–202.

[24] T. Steihaug and A. Wolfbrandt, *An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff ordinary differential equations*, Math. Comp. **33** (1979), 521–534.

[25] K. Strehmel, R. Weiner, and I. Dannehl, *On error behaviour of partitioned linearly implicit Runge-Kutta methods for stiff and differential algebraic systems*, BIT **30** (1990), 358–375.

[26] R. Weiner, M. Arnold, P. Rentrop, and K. Strehmel, *Partitioning strategies in Runge-Kutta type methods*, IMA Journal of Numerical Analysis **13** (1993), 303–319.

[27] J. Wensch, *Zur numerischen Behandlung differential-algebraischer Systeme von höherem Index*, Ph.D. thesis, Universität Halle, in preparation.

FB Mathematik und Informatik
Martin-Luther-Universität Halle-Wittenberg
Postfach
D-06099 Halle/Saale
Germany

Jörg Wensch, e-mail: wensch@mail.mathematik.uni-halle.de
Karl Strehmel, e-mail: strehmel@mathematik.uni-halle.d400.de
Rüdiger Weiner, e-mail: rw@mail.mathematik.uni-halle.de