
Numerical Algorithms **31:** 171–191, 2002.

Iterative solution of SPARK methods applied to DAEs

Laurent O. Jay *

Department of Mathematics, 14 MacLean Hall, The University of Iowa, Iowa City, IA 52242-1419, USA
E-mail: ljay@math.uiowa.edu; na.ljay@na-net.ornl.gov

Received 8 January 2001; revised 4 October 2001

In this article a broad class of systems of implicit differential–algebraic equations (DAEs) is considered, including the equations of mechanical systems with holonomic and nonholonomic constraints. Solutions to these DAEs can be approximated numerically by applying a class of super partitioned additive Runge–Kutta (SPARK) methods. Several properties of the SPARK coefficients, satisfied by the family of Lobatto IIIA-B-C-C*-D coefficients, are crucial to deal properly with the presence of constraints and algebraic variables. A main difficulty for an efficient implementation of these methods lies in the numerical solution of the resulting systems of nonlinear equations. Inexact modified Newton iterations can be used to solve these systems. Linear systems of the modified Newton method can be solved approximately with a preconditioned linear iterative method. Preconditioners can be obtained after certain transformations to the systems of nonlinear and linear equations. These transformations rely heavily on specific properties of the SPARK coefficients. A new truly parallelizable preconditioner is presented.

Keywords: differential–algebraic equations, holonomic constraints, inexact modified Newton method, Lobatto coefficients, mechanical systems, nonholonomic constraints, overdetermined DAEs, perturbation index, preconditioning, Runge–Kutta methods, stiffness

AMS subject classification: 65F10, 65H10, 65L05, 65L06, 65L80, 70F20, 70F25, 70H03, 70H45

1. Introduction

In this article a broad class of systems of possibly stiff and implicit differential–algebraic equations (DAEs) is considered, including Hessenberg DAEs of index 1, 2, and 3 [1,5,6,8,9]. These equations encompass the formulation of mechanical systems with mixed constraints of holonomic, nonholonomic, scleronomic, and rheonomic types [7,16,17]. Solutions to these DAEs can be approximated numerically by applying a class of super partitioned additive Runge–Kutta (SPARK) methods, such as the combination of Lobatto IIIA-B-C-C*-D methods [9]. SPARK methods can take advantage of splitting the differential equations into different terms and of partitioning the variables into different classes. Several properties of the SPARK coefficients, satisfied by the Lo-

* This material is based upon work supported by the National Science Foundation under Grant No. 9983708. This research was also supported in part by NASA Award No. 1210679, JPL subcontract No. 1213833.

batto family, are essential to treat the constraints and the algebraic variables properly. A main difficulty for an efficient implementation of these methods lies in the numerical solution of the resulting systems of nonlinear equations. For this purpose inexact modified Newton iterations can be used, extending techniques proposed for the solution of implicit Runge–Kutta (IRK) methods applied to implicit systems of stiff ordinary differential equations (ODEs) [10–12]. Such an extension is not straightforward, as the presence of constraints and algebraic variables adds some extra difficulty. Linear systems of the modified Newton method can be solved approximately with a preconditioned linear iterative method after certain transformations to the systems of nonlinear and linear equations. These transformations rely heavily on specific properties of the SPARK coefficients. For an s-stage SPARK method and stiff DAEs the decomposition of at most $s + 1$ independent submatrices of the same dimension as the DAEs is required to build an efficient preconditioner. The main purpose of this paper is to present the steps involved to put the system of linear equations of the modified Newton method in a form such that preconditioning these linear equations can be done in a straightforward manner following the results of [10–12].

In section 2, the class of implicit DAEs considered in this article is presented. In section 3, the definition of SPARK methods applied to these DAEs is given. Some properties of the SPARK coefficients are given which are crucial for an efficient implementation of these methods. In section 4, an approximate Jacobian to the system of nonlinear equations is derived after certain linear transformations. Section 5 describes the steps involved to transform the approximate Jacobian before application of a preconditioner. A new truly parallelizable preconditioner is succinctly presented.

2. The system of implicit differential–algebraic equations

Consider the following class of systems of implicit differential–algebraic equations (DAEs)

$$\frac{\mathrm{d}}{\mathrm{d}t}q(t, y) = v(t, y, z), \tag{1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}p(t, y, z) = f(t, y, z, u, \gamma, \lambda, \psi), \tag{1b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t, y, z, u) = d(t, y, z, u, \gamma, \lambda, \psi), \tag{1c}$$

$$m(t, y, z, u, \gamma) = 0, \tag{1d}$$

$$h(t, y, z) = 0, \tag{1e}$$

$$g(t, y) = 0, \tag{1f}$$

which may present some stiffness. These equations encompass Hessenberg DAEs of index 1, 2, and 3 [1,5,6,8,9]. They also include the formulation of mechanical systems with mixed constraints of holonomic, nonholonomic, scleronomic, and rheonomic types [7,13,16,17]. In mechanics the quantities q, v, p represent respectively gener-

alized coordinates, generalized velocities, and generalized momenta. The right-hand side f of (1b) contains generalized forces acting on the system and (1c) describes the dynamics of external variables u. The algebraic variables λ and ψ are Lagrange multipliers associated respectively to the nonholonomic constraints (1e) and the holonomic constraints (1f). The equations of constrained systems in mechanics can be derived from Newton's law of motion and the generalized Gauss variational principle of least constraint [13]. It is assumed that the constraints (1f) can also be expressed as

$$g(t, y) = r(t, q(t, y)) = 0. \tag{1g}$$

The variable $t \in \mathbb{R}$ is the independent variable and

$$\begin{aligned}
y &= \left(y^1, \ldots, y^{n_y}\right)^{\mathrm{T}} \in \mathbb{R}^{n_y}, \\
z &= \left(z^1, \ldots, z^{n_z}\right)^{\mathrm{T}} \in \mathbb{R}^{n_z}, \\
u &= \left(u^1, \ldots, u^{n_u}\right)^{\mathrm{T}} \in \mathbb{R}^{n_u}, \\
\gamma &= \left(\gamma^1, \ldots, \gamma^{n_\gamma}\right)^{\mathrm{T}} \in \mathbb{R}^{n_\gamma}, \\
\lambda &= \left(\lambda^1, \ldots, \lambda^{n_\lambda}\right)^{\mathrm{T}} \in \mathbb{R}^{n_\lambda}, \\
\psi &= \left(\psi^1, \ldots, \psi^{n_\psi}\right)^{\mathrm{T}} \in \mathbb{R}^{n_\psi}, \\
q &: \mathbb{R} \times \mathbb{R}^{n_y} \longrightarrow \mathbb{R}^{n_y}, \\
p &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \longrightarrow \mathbb{R}^{n_z}, \\
c &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \longrightarrow \mathbb{R}^{n_u}, \\
m &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\gamma} \longrightarrow \mathbb{R}^{n_\gamma}, \\
h &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \longrightarrow \mathbb{R}^{n_\lambda}, \\
g &: \mathbb{R} \times \mathbb{R}^{n_y} \longrightarrow \mathbb{R}^{n_\psi}, \\
r &: \mathbb{R} \times \mathbb{R}^{n_y} \longrightarrow \mathbb{R}^{n_\psi}, \\
v &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \longrightarrow \mathbb{R}^{n_y}, \\
f &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\gamma} \times \mathbb{R}^{n_\lambda} \times \mathbb{R}^{n_\psi} \longrightarrow \mathbb{R}^{n_z}, \\
d &: \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\gamma} \times \mathbb{R}^{n_\lambda} \times \mathbb{R}^{n_\psi} \longrightarrow \mathbb{R}^{n_u}.
\end{aligned}$$

The variables y, z, u are called the *differential* variables and the variables γ, λ, ψ are called the *algebraic* variables. The latter often correspond to Lagrange multipliers when the DAEs are derived from some constrained variational principle [7,13]. The initial values $y_0, z_0, u_0, \gamma_0, \psi_0, \lambda_0$ at t_0 are supposed to be given. Some differentiability conditions on the above functions and consistency of the initial values are assumed to ensure existence and uniqueness of the solution. In a neighborhood of the solution the following conditions are supposed to be satisfied

$$q_y \text{ is invertible,} \tag{2a}$$

$$p_z \text{ is invertible,} \tag{2b}$$

$$c_u \text{ is invertible,} \tag{2c}$$

$$m_\gamma \text{ is invertible,} \tag{2d}$$

$$\begin{pmatrix} p_z & -f_\lambda & -f_\psi \\ h_z & O & O \\ r_q v_z & O & O \end{pmatrix} \text{ is invertible.} \tag{2e}$$

Notice that from (1g) $g_y = r_q q_y$ holds, hence $r_q v_z = g_y (q_y)^{-1} v_z$ in (2e). Under conditions (2a)–(2c) explicit expressions for the derivatives of the differential variables can be obtained. The differential equations (1a)–(1c) lead to

$$q_t(t, y) + q_y(t, y)\frac{\mathrm{d}y}{\mathrm{d}t} = v(t, y, z), \tag{3a}$$

$$p_t(t, y, z) + p_y(t, y, z)\frac{\mathrm{d}y}{\mathrm{d}t} + p_z(t, y, z)\frac{\mathrm{d}z}{\mathrm{d}t} = f(t, y, z, u, \gamma, \lambda, \psi), \tag{3b}$$

$$\begin{aligned} &c_t(t, y, z, u) + c_y(t, y, z, u)\frac{\mathrm{d}y}{\mathrm{d}t} + c_z(t, y, z, u)\frac{\mathrm{d}z}{\mathrm{d}t} + c_u(t, y, z, u)\frac{\mathrm{d}u}{\mathrm{d}t} \\ &\quad = d(t, y, z, u, \gamma, \lambda, \psi). \end{aligned} \tag{3c}$$

For example, (3a) leads to

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \big(q_y(t, y)\big)^{-1}\big(v(t, y, z) - q_t(t, y)\big).$$

Implicit expressions for the algebraic variables can be obtained by application of the implicit function theorem. From (1d), (2d) the algebraic variables γ can be implicitly expressed as $\gamma = \gamma(t, y, z, u)$. Differentiating the constraints (1e) once gives

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}h(t, y, z) = h_t(t, y, z) + h_y(t, y, z)\frac{\mathrm{d}y}{\mathrm{d}t} + h_z(t, y, z)\frac{\mathrm{d}z}{\mathrm{d}t}. \tag{4a}$$

Differentiating the constraints (1g) twice leads successively to

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}g(t, y) = r_t\big(t, q(t, y)\big) + r_q\big(t, q(t, y)\big)v(t, y, z), \tag{4b}$$

$$\begin{aligned} 0 = \frac{\mathrm{d}^2}{\mathrm{d}t^2}g(t, y) &= r_{tt}\big(t, q(t, y)\big) + 2r_{tq}\big(t, q(t, y)\big)v(t, y, z) \\ &\quad + r_{qq}\big(t, q(t, y)\big)\big(v(t, y, z), v(t, y, z)\big) + r_q\big(t, q(t, y)\big)v_t(t, y, z) \\ &\quad + r_q\big(t, q(t, y)\big)v_y(t, y, z)\frac{\mathrm{d}y}{\mathrm{d}t} + r_q\big(t, q(t, y)\big)v_z(t, y, z)\frac{\mathrm{d}z}{\mathrm{d}t}. \end{aligned} \tag{4c}$$

Hence from (2e), (3b), (4a), (4c) implicit expressions for the algebraic variables λ, ψ can be obtained. The exact solution must satisfy these additional so-called *underlying constraints* (4). To be consistent the initial values $y_0, z_0, u_0, \gamma_0, \psi_0, \lambda_0$ at t_0 must satisfy the whole set of constraints (1d)–(1f), (4). After one more differentiation of the constraints (1d), (4a), (4c) explicit expressions for the derivatives of the algebraic variables can be obtained, forming together with (1a)–(1c) an underlying system of ODEs.

The *overdetermined* system of DAEs (1), (4) is therefore of *differential* and *perturbation index* 1 [8]. The constraints (1d)–(1f) are often called the index 1, 2, 3 constraints respectively, although the notion of index, especially of perturbation index, is more closely related to variables than to equations [8, p. 10]. DAEs of perturbation index less than or equal to 1 such as (1), (4) are well-posed contrary to higher perturbation index DAEs such as (1). From a numerical and computational perspective and from the mathematical point of view of well- and ill-posedness the perturbation index is certainly one of the most relevant notions of index.

With the equations of mechanical systems in mind where different types of forces are present, see [7,9,16,17], decompositions of the right-hand sides of (1a)–(1c) can be considered

$$v(t, y, z) = \sum_{m=1}^{m_{\max}} v_m(t, y, z), \tag{5a}$$

$$f(t, y, z, u, \gamma, \lambda, \psi) = \sum_{m=1}^{m_{\max}} f_m(t, y, z, u, \gamma, \lambda, \psi), \tag{5b}$$

$$d(t, y, z, u, \gamma, \lambda, \psi) = \sum_{m=1}^{m_{\max}} d_m(t, y, z, u, \gamma, \lambda, \psi). \tag{5c}$$

The functions v_m, f_m, d_m are supposed to have distinct properties and can therefore be numerically treated in a different way. The value of $m_{\max}$ corresponds to different classes of certain types of right-hand side terms. This value should be reasonably small. For example, mechanical systems may include different types of forces such as conservative, dissipative, explosive, and highly oscillatory forces, hence typically $m_{\max} = 4$. For the application of the numerical methods considered in this paper, the following additional assumptions are made

$$f_1(t, y, z, u, \gamma, \lambda, \psi) = f_1(t, y, z, u, \gamma), \qquad d_1(t, y, z, u, \gamma, \lambda, \psi) = d_1(t, y, z, u, \gamma). \tag{5d}$$

This is obviously not a restriction on the system (1) per se, but it is used as a restriction on the application of SPARK methods, see section 3.

3. SPARK methods

The application of SPARK methods to the overdetermined system of implicit DAEs (1), (4) is tentatively given as follows, generalizing the definition of [9]:

Definition 1. One step of an *s-stage super partitioned additive Runge–Kutta (SPARK) method* applied with stepsize h to the overdetermined system of implicit differential–algebraic equations (1), (4) satisfying the assumptions with decompositions (5) reads

$$Q_i - q_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{ij,m} v_m(T_j, Y_j, Z_j) = 0 \quad \text{for } i = 1, \ldots, s, \tag{6a}$$

$$P_i - p_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{ij,m} f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) = 0 \quad \text{for } i = 1, \ldots, s, \tag{6b}$$

$$C_i - c_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{ij,m} d_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) = 0 \quad \text{for } i = 1, \ldots, s, \tag{6c}$$

$$m(T_i, Y_i, Z_i, U_i, \Gamma_i) = 0 \qquad \text{for } i = 1, \ldots, s, \tag{6d}$$

$$h(T_i, Y_i, Z_i) = 0 \qquad \text{for } i = 1, \ldots, s, \tag{6e}$$

$$r\left(T_i, q_0 + h \sum_{j=1}^{s} a_{ij,1} v(T_j, Y_j, Z_j)\right) = 0 \qquad \text{for } i = 1, \ldots, s, \tag{6f}$$

$$q_1 - q_0 - h \sum_{j=1}^{s} b_j v(T_j, Y_j, Z_j) = 0, \tag{6g}$$

$$p_1 - p_0 - h \sum_{j=1}^{s} b_j f(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) = 0, \tag{6h}$$

$$c_1 - c_0 - h \sum_{j=1}^{s} b_j d(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) = 0, \tag{6i}$$

$$m(t_1, y_1, z_1, u_1, \gamma_1) = 0, \tag{6j}$$

$$h(t_1, y_1, z_1) = 0, \tag{6k}$$

$$r(t_1, q_1) = 0, \tag{6l}$$

$$r_t(t_1, q_1) + r_q(t_1, q_1) v(t_1, y_1, z_1) = 0, \tag{6m}$$

where

$$\begin{aligned} q_0 &:= q(t_0, y_0), \\ p_0 &:= p(t_0, y_0, z_0), \\ c_0 &:= c(t_0, y_0, z_0, u_0), \\ T_i &:= t_0 + c_i h \qquad \text{for } i = 1, \ldots, s, \\ Q_i &:= q(T_i, Y_i) \qquad \text{for } i = 1, \ldots, s, \end{aligned}$$

$$
\begin{aligned}
P_i &:= p(T_i, Y_i, Z_i) && \text{for } i = 1, \ldots, s,\\
C_i &:= c(T_i, Y_i, Z_i, U_i) && \text{for } i = 1, \ldots, s,\\
t_1 &:= t_0 + h,\\
q_1 &:= q(t_1, y_1),\\
p_1 &:= p(t_1, y_1, z_1),\\
c_1 &:= c(t_1, y_1, z_1, u_1).
\end{aligned}
$$

The RK coefficients matrices of $m_{\max}$ Runge–Kutta (RK) methods based on the same quadrature formula $(b_i, c_i)_{i=1,\ldots,s}$ are denoted by $A_m := (a_{ij,m})_{i,j=1,\ldots,s}$ for $m = 1, \ldots, m_{\max}$. To ensure existence and uniqueness of the numerical solution, only a certain linear combination of equations (6e), (6k) is actually considered, see equations (14e).

From this tentative definition of SPARK methods results a system of nonlinear equations to be solved for the internal stages $Y_i, Z_i, U_i, \Gamma_i, \Lambda_i, \Psi_i$ for $i = 1, \ldots, s$ and for the numerical approximation at t_1 given by y_1, z_1, u_1, γ_1. In general existence and uniqueness to these nonlinear equations cannot be shown unless some assumptions on the SPARK coefficients are made. Also equations (6e), (6k) for the constraints (1e) cannot be all satisfied. The actual definition of SPARK methods is given in definition 5. Accurate values for the algebraic variables $\gamma_1, \lambda_1, \psi_1$ are not necessary for the step by step integration. In any case the accuracy of these algebraic variables does not influence the convergence of the other variables and the properties of SPARK methods since the values $\gamma_0, \lambda_0, \psi_0$ do not enter explicitly the definition of SPARK methods. For SPARK methods satisfying $c_s = 1$ the approximations given by $\gamma_1 := \Gamma_s, \lambda_1 := \Lambda_s, \psi_1 := \Psi_s$ are adequate.

3.1. Properties of SPARK coefficients

In this article I_m denotes the $m \times m$ identity matrix, $O_{m,n}$ denotes the $m \times n$ zero matrix, $b := (b_1, \ldots, b_s)^{\mathrm{T}}$ is the weight vector, $e_i := (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ is the ith s-dimensional unit basis vector, and $0_s := (0, \ldots, 0)^{\mathrm{T}}$ is the s-dimensional zero vector. It is assumed that the number of internal stages satisfies $s \geqslant 2$. SPARK methods (6) satisfying the following assumptions are considered

$$e_1^{\mathrm{T}} A_1 = 0_s^{\mathrm{T}}, \tag{7a}$$

$$e_s^{\mathrm{T}} A_1 = b^{\mathrm{T}}, \tag{7b}$$

$$e_s^{\mathrm{T}} A_3 = b^{\mathrm{T}}, \tag{7c}$$

$$A_1 A_m = \begin{pmatrix} 0_s^{\mathrm{T}} \\ N \end{pmatrix} \quad \text{for } m = 2, \ldots, m_{\max}, \tag{7d}$$

$$\begin{pmatrix} N \\ b^{\mathrm{T}} \end{pmatrix} \text{ is invertible,} \tag{7e}$$

$$A_3 \text{ is invertible.} \tag{7f}$$

These assumptions are satisfied for example by the Lobatto family with $m_{\max} = 5$ and A_1, A_2, A_3, A_4, A_5 being the RK matrices of Lobatto IIIA-B-C-C*-D coefficients, respectively [9]. The assumptions (7b), (7c) are *stiff accuracy* conditions. Notice that (7a) is also a direct consequence of (7f) and (7d) for $m = 3$. Let $\widetilde{A}_1$ be the $(s-1) \times s$ submatrix of A_1 given by the relation

$$A_1 = \begin{pmatrix} 0_s^{\mathrm{T}} \\ \widetilde{A}_1 \end{pmatrix}. \tag{8}$$

The following lemmas will be extremely useful to obtain an efficient implementation of SPARK methods applied to DAEs.

Lemma 2. The relation

$$\begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} N \\ b^{\mathrm{T}} \end{pmatrix} = A_3 \tag{9}$$

follows from (7c)–(7f).

Proof. The relation $\widetilde{A}_1 A_3 = N$ follows from (7d) for $m = 3$. Hence, together with (7c) it leads to

$$\begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix} A_3 = \begin{pmatrix} N \\ b^{\mathrm{T}} \end{pmatrix}.$$

The invertibility of the left matrix on the left-hand side follows from the conditions (7e), (7f). □

Under the assumptions of lemma 2 the following $s \times (s+1)$ matrices are defined

$$Q_{s,s+1} := \begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{A}_1 & 0_{s-1} \\ 0_s^{\mathrm{T}} & 1 \end{pmatrix} = \begin{pmatrix} I_s & 0_s \end{pmatrix} + P_{s,s+1}, \tag{10a}$$

$$P_{s,s+1} := \begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} O_{s,s-1} & -e_s & e_s \end{pmatrix} = \begin{pmatrix} O_{s,s-1} & -p_s & p_s \end{pmatrix}, \tag{10b}$$

where

$$p_s := \begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} e_s = \begin{pmatrix} v_{s-1} \\ 1 \end{pmatrix}. \tag{10c}$$

Lemma 3. From (7c)–(7f) it follows that

$$Q_{s,s+1} \begin{pmatrix} A_m & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} = \begin{pmatrix} A_3 & 0_s \end{pmatrix} \quad \text{for } m = 2, \ldots, m_{\max}.$$

Moreover, if in addition (7b) holds then the following relation is obtained

$$Q_{s,s+1}\begin{pmatrix} A_1 & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} = \begin{pmatrix} A_1 & 0_s \end{pmatrix}.$$

Proof. The relations

$$\begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{A}_1 & 0_{s-1} \\ 0_s^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} A_m & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} = \begin{pmatrix} A_3 & 0_s \end{pmatrix} \quad \text{for } m = 2, \ldots, m_{\max}$$

follow directly from (7d) and lemma 2. For $m = 1$ the relation

$$\begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{A}_1 & 0_{s-1} \\ 0_s^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} A_1 & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} = \begin{pmatrix} A_1 & 0_s \end{pmatrix}$$

follows from

$$\begin{pmatrix} O_{s,s-1} & -e_s & e_s \end{pmatrix} \begin{pmatrix} A_1 & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} = O_{s,s+1}$$

which is a simple consequence of (7b). □

Lemma 4. Consider the matrix $Q_{s,s+1}$ as defined in (10) and the $(s+1)$-dimensional vector $q_{s+1} := (0, \ldots, 0, -1, 1)^{\mathrm{T}}$. Then the following $(s+1) \times (s+1)$ matrix

$$Q_{s+1,s+1} := \begin{pmatrix} Q_{s,s+1} \\ q_{s+1}^{\mathrm{T}} \end{pmatrix} \tag{11}$$

is invertible.

Proof. The expressions (10) give

$$Q_{s,s+1} = \begin{pmatrix} I_{s-1} & -v_{s-1} & v_{s-1} \\ 0_{s-1}^{\mathrm{T}} & 0 & 1 \end{pmatrix}, \tag{12}$$

hence $\det(Q_{s+1,s+1}) = 1$ is easily obtained. In fact, the inverse to $Q_{s+1,s+1}$ can be given explicitly by using the factorization

$$Q_{s+1,s+1} = R_{s+1,s+1} S_{s+1,s+1} \tag{13a}$$

where

$$R_{s+1,s+1} := \begin{pmatrix} 1 & & O & \\ & \ddots & & p_s \\ & & 1 & \\ O & & & 1 \end{pmatrix}, \qquad S_{s+1,s+1} := \begin{pmatrix} 1 & & & O \\ & \ddots & & \\ & & 1 & \\ O & & -1 & 1 \end{pmatrix}. \tag{13b}$$

Hence, $Q_{s+1,s+1}^{-1} = S_{s+1,s+1}^{-1} R_{s+1,s+1}^{-1}$ holds where

$$S_{s+1,s+1}^{-1} = \begin{pmatrix} 1 & & & O \\ & \ddots & & \\ & & 1 & \\ O & & 1 & 1 \end{pmatrix}, \qquad R_{s+1,s+1}^{-1} = \begin{pmatrix} 1 & & O & \\ & \ddots & & -p_s \\ & & 1 & \\ O & & & 1 \end{pmatrix}. \qquad \square$$

3.2. *The system of nonlinear equations*

It is essential to put the system of nonlinear equations in a form such that preconditioning the linear equations of the modified Newton method can be done in a straightforward manner following the results of [10–12], see section 5. From the assumption (7a) and the consistency condition $r(t_0, q_0) = 0$ the equation (6f) for $i = 1$ is automatically satisfied. A consequence of the assumption (7b) is

$$r(t_1, q_1) = r\left(T_s, q_0 + h \sum_{j=1}^{s} a_{sj,1} v(T_j, Y_j, Z_j)\right),$$

therefore by (6f) for $i = s$ equation (6l) is also automatically satisfied. Instead of solving directly the remaining set of equations of (6), some specific linear transformations are applied. The remaining equations are expressed by making use of the matrices $Q_{s,s+1}$ (10) and $Q_{s+1,s+1}$ (11) as follows:

Definition 5. The actual definition of SPARK methods applied to (1)–(4)–(5) is taken as

$$(Q_{s+1,s+1} \otimes I_{n_y}) \begin{pmatrix} Q_1 - q_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{1j,m} v_m(T_j, Y_j, Z_j) \\ \vdots \\ Q_s - q_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{sj,m} v_m(T_j, Y_j, Z_j) \\ q_1 - q_0 - h \sum_{j=1}^{s} b_j v(T_j, Y_j, Z_j) \end{pmatrix} = 0, \qquad (14a)$$

$$
\begin{aligned}
&(Q_{s+1,s+1} \otimes I_{n_z}) \\
&\quad \times \begin{pmatrix} P_1 - p_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{1j,m} f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \\ \vdots \\ P_s - p_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{sj,m} f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \\ p_1 - p_0 - h \sum_{j=1}^{s} b_j f(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \end{pmatrix} = 0, \quad (14b) \\
&(Q_{s+1,s+1} \otimes I_{n_u}) \\
&\quad \times \begin{pmatrix} C_1 - c_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{1j,m} d_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \\ \vdots \\ C_s - c_0 - h \sum_{j=1}^{s} \sum_{m=1}^{m_{\max}} a_{sj,m} d_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \\ c_1 - c_0 - h \sum_{j=1}^{s} b_j d(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \end{pmatrix} = 0, \quad (14c) \\
&(Q_{s+1,s+1} \otimes I_{n_\gamma}) \begin{pmatrix} m(T_1, Y_1, Z_1, U_1, \Gamma_1) \\ \vdots \\ m(T_s, Y_s, Z_s, U_s, \Gamma_s) \\ m(t_1, y_1, z_1, u_1, \gamma_1) \end{pmatrix} = 0, \quad (14d) \\
&(Q_{s,s+1} \otimes I_{n_\lambda}) \begin{pmatrix} h(T_1, Y_1, Z_1) \\ \vdots \\ h(T_s, Y_s, Z_s) \\ h(t_1, y_1, z_1) \end{pmatrix} = 0, \quad (14e) \\
&\left(\begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \otimes I_{n_\psi} \right) \begin{pmatrix} \frac{1}{h} r\big(T_2, q_0 + h \sum_{j=1}^{s} a_{2j,1} v(T_j, Y_j, Z_j)\big) \\ \vdots \\ \frac{1}{h} r\big(T_s, q_0 + h \sum_{j=1}^{s} a_{sj,1} v(T_j, Y_j, Z_j)\big) \\ r_t(t_1, q_1) + r_q(t_1, q_1) v(t_1, y_1, z_1) \end{pmatrix} = 0. \quad (14f)
\end{aligned}
$$

Equations (14e) correspond only to a linear combination of the constraints (6e), (6k). From (12) equation (6k) is truly the only one preserved among (6e), (6k), this implies that the numerical solution at t_1 still satisfies the constraint (1e). This linear combination (14e) of (6e), (6k) is somehow necessary to ensure existence and uniqueness of the numerical solution since in (6) there are only s algebraic variables Λ_i ($i = 1, \ldots, s$) for $s + 1$ sets of equations (6e), (6k). In order to combine the constraints (6f) and (6l) properly the equations (6f) have been multiplied by $1/h$ where h is the stepsize. Since $r(t_0, q_0) = 0$, this can be interpreted as a finite difference approximation to $\mathrm{d}r(t, q(t, y))/\mathrm{d}t = 0$ at T_i

$$\frac{\mathrm{d}}{\mathrm{d}t} r\big(T_i, q\big(T_i, y(T_i)\big)\big) \approx \frac{r(T_i, q_0 + h\sum_{j=1}^{s} a_{ij,1} v(T_j, Y_j, Z_j)) - r(t_0, q_0)}{h}.$$

The main reason for these linear transformations is to obtain an advantageous structure of the approximate Jacobian, given in section 4, for the construction of efficient preconditioners, to be discussed in section 5.

Notice that because of the factorization (13a), (13b) multiplication by $Q_{s+1,s+1}$ of equations (14a)–(14c) has an implication easily interpretable. For example, (14b) can be rewritten as

$$\big(R_{s+1,s+1} \otimes I_{n_z}\big) \times \begin{pmatrix} P_1 - p_0 - h\sum_{j=1}^{s}\sum_{m=1}^{m_{\max}} a_{1j,m} f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \\ \vdots \\ P_s - p_0 - h\sum_{j=1}^{s}\sum_{m=1}^{m_{\max}} a_{sj,m} f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \\ p_1 - P_s - h\sum_{j=1}^{s}\sum_{m=1}^{m_{\max}} (b_j - a_{sj,m}) f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) \end{pmatrix} = 0. \quad (14g)$$

One effect is to remove stiffness from the last set of equations, provided the terms causing stiffness are treated with coefficients $a_{ij,m}$ satisfying the *stiff accuracy* condition $a_{sj,m} = b_j$ for $j = 1, \ldots, s$, see (7b), (7c).

A set of dummy equations can be appended to (14e), (14f)

$$h(t_1, y_1, z_1) - h(T_s, Y_s, Z_s) + \lambda_1 = 0, \quad (14h)$$

$$r_t(t_1, q_1) + r_q(t_1, q_1) v(t_1, y_1, z_1) - \big(r_t(T_s, Q_s) + r_q(T_s, Q_s) v(T_s, Y_s, Z_s)\big) + \psi_1 = 0. \quad (14i)$$

These equations must be taken as a dummy definition of λ_1, ψ_1. Of course the exact solution $\lambda(t), \psi(t)$ at $t = t_0 + h$ must not be approximated by λ_1, ψ_1 as given above, but, for example, by Λ_s, Ψ_s when $c_s = 1$. The unknowns λ_1, ψ_1 have been included only for ease of presentation in the derivation hereafter, but they are actually not unknowns

of the nonlinear system (14a)–(14f). In (14) a system of nonlinear equations has been obtained for the following vector collecting all unknowns

$$X := (Y_1, Z_1, U_1, \Gamma_1, \Lambda_1, \Psi_1, \ldots, Y_s, Z_s, U_s, \Gamma_s, \Lambda_s, \Psi_s, y_1, z_1, u_1, \gamma_1, \lambda_1, \psi_1)^{\mathrm{T}}. \tag{15}$$

4. Inexact modified Newton iterations for SPARK methods

The iteration schemes proposed to solve the system of nonlinear equations corresponding to the application of IRK methods to DAEs have generally been based on ad hoc modifications of the simplified Newton method [5]. For SPARK methods there are additional difficulties to obtain an efficient implementation due to their additive and partitioned nature. This paper proposes the use of inexact modified Newton iterations or more precisely, using another terminology, of modified Newton-iterative methods, extending techniques developed in [10–12]. Instead of solving exactly the linear systems of the modified Newton method, they can be solved approximately and iteratively after application of specific linear transformations and the use of a preconditioner.

4.1. Inexact modified Newton iterations

Modified Newton iterations applied to the set of equations (14) read as follows

$$M\Delta X^k = -G\big(X^k\big), \qquad X^{k+1} = X^k + \Delta X^k, \quad k = 0, 1, 2, \ldots, \tag{16}$$

where M is a modified Jacobian, i.e., roughly speaking an approximation to the exact Jacobian,

$$\Delta X := (\Delta X_1, \ldots, \Delta X_s, \Delta x_1)^{\mathrm{T}}, \tag{17a}$$

$$\Delta X_i := (\Delta Y_i, \Delta Z_i, \Delta U_i, \Delta \Gamma_i, \Delta \Lambda_i, \Delta \Psi_i)^{\mathrm{T}} \quad \text{for } i = 1, \ldots, s, \tag{17b}$$

$$\Delta x_1 := (\Delta y_1, \Delta z_1, \Delta u_1, \Delta \gamma_1, \Delta \lambda_1, \Delta \psi_1)^{\mathrm{T}}, \tag{17c}$$

and $G(X)$ corresponds to the expressions in (14) reordered accordingly.

A direct decomposition of the modified Jacobian M may be inefficient. In the *inexact modified Newton method* the linear systems (16) are solved only approximately, generally by a preconditioned linear iterative method, such as preconditioned versions of Richardson or GMRES iterations [3,4,11,14,15]. This requires the construction of a good preconditioner, see section 5. A sequence of iterates $\widehat{X}^k$ with a residual error $r_k := M\Delta\widehat{X}^k + G(\widehat{X}^k)$ is obtained at each iteration. Sufficient a priori and a posteriori conditions to ensure convergence of the inexact modified Newton iterates toward the solution of a system of nonlinear equations have been given in [11]. In combination with a linear iterative method the inexact Newton method is called a *modified Newton-iterative method*. The use of preconditioned linear iterative methods for the numerical solution of ODEs and DAEs was first considered in the context of implicit multistep methods by Brown et al. [2].

4.2. The "simplified" Jacobian

In a standard approach, the system of nonlinear equations (14) is solved by simplified Newton iterations. This requires the *simplified Jacobian* which is the Jacobian at the initial guess. The simplified Jacobian corresponding to the equations (14a)–(14f) with respect to the variables in (15) can be expressed as follows

$$
\begin{aligned}
&Q_{s+1,s+1} \otimes \begin{pmatrix} q_y & O & O & O & O & O \end{pmatrix} \\
&\quad -h \sum_{m=1}^{m_{\max}} Q_{s+1,s+1} \begin{pmatrix} A_m & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} \otimes \begin{pmatrix} v_{my} & v_{mz} & O & O & O & O \end{pmatrix}, && \text{(18a)} \\
&Q_{s+1,s+1} \otimes \begin{pmatrix} p_y & p_z & O & O & O & O \end{pmatrix} \\
&\quad -h \sum_{m=1}^{m_{\max}} Q_{s+1,s+1} \begin{pmatrix} A_m & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} \otimes \begin{pmatrix} f_{my} & f_{mz} & f_{mu} & f_{m\gamma} & f_{m\lambda} & f_{m\psi} \end{pmatrix}, && \text{(18b)} \\
&Q_{s+1,s+1} \otimes \begin{pmatrix} c_y & c_z & c_u & O & O & O \end{pmatrix} \\
&\quad -h \sum_{m=1}^{m_{\max}} Q_{s+1,s+1} \begin{pmatrix} A_m & 0_s \\ b^{\mathrm{T}} & 0 \end{pmatrix} \otimes \begin{pmatrix} d_{my} & d_{mz} & d_{mu} & d_{m\gamma} & d_{m\lambda} & d_{m\psi} \end{pmatrix}, && \text{(18c)} \\
&Q_{s+1,s+1} \otimes \begin{pmatrix} m_y & m_z & m_u & m_\gamma & O & O \end{pmatrix}, && \text{(18d)} \\
&Q_{s,s+1} \otimes \begin{pmatrix} h_y & h_z & O & O & O & O \end{pmatrix}, && \text{(18e)} \\
&\begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{A}_1 & 0_{s-1} \\ 0_s^{\mathrm{T}} & 1 \end{pmatrix} \otimes \begin{pmatrix} r_q v_y & r_q v_z & O & O & O & O \end{pmatrix} \\
&\qquad + \begin{pmatrix} \widetilde{A}_1 \\ e_s^{\mathrm{T}} \end{pmatrix}^{-1} \begin{pmatrix} O_{s-1,s} & 0_{s-1} \\ 0_s^{\mathrm{T}} & 1 \end{pmatrix} \otimes \begin{pmatrix} r_{tq} q_y + r_{qq}(q_y \cdot, v) & O & O & O & O & O \end{pmatrix} \\
&\quad = Q_{s,s+1} \otimes \begin{pmatrix} r_q v_y & r_q v_z & O & O & O & O \end{pmatrix} \\
&\qquad + \begin{pmatrix} O_{ss} & p_s \end{pmatrix} \otimes \begin{pmatrix} r_{tq} q_y + r_{qq}(q_y \cdot, v) & O & O & O & O & O \end{pmatrix}, && \text{(18f)}
\end{aligned}
$$

where the symbol $\otimes$ denotes the tensor product. The arguments of the expressions q_y, v_{my}, etc., which have been omitted, are given by the initial values t_0, y_0, z_0, u_0, γ_0, λ_0, ψ_0. Strictly speaking this is in fact not really the simplified Jacobian since the initial guess of the iterations is generally not given by the initial values. Nevertheless, the terminology of simplified Jacobian to refer to (18) will be kept, since this is how it is generally called following, e.g., [6, section IV.8]. The terminology of *modified Jacobian* would actually be more correct.

4.3. An approximate Jacobian

Some modifications to the simplified Jacobian (18) can actually be used, since it is not necessary to keep the full simplified Jacobian to ensure convergence of the modified

Newton iterates (16). The expression of a so-called *approximate Jacobian L* is given here. It will be used in the discussion given in section 5 about the construction of preconditioners. It should be noticed that it is not necessary to use exactly the approximate Jacobian presented here to ensure convergence of the whole inexact modified Newton procedure, additional modifications are possible.

A main point in the specification of the approximate Jacobian concerns the equations

$$q_1 - Q_s - h\sum_{j=1}^{s}\sum_{m=1}^{m_{\max}}(b_j - a_{sj,m})v_m(T_j, Y_j, Z_j) = 0, \tag{19a}$$

$$p_1 - P_s - h\sum_{j=1}^{s}\sum_{m=1}^{m_{\max}}(b_j - a_{sj,m})f_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) = 0, \tag{19b}$$

$$c_1 - C_s - h\sum_{j=1}^{s}\sum_{m=1}^{m_{\max}}(b_j - a_{sj,m})d_m(T_j, Y_j, Z_j, U_j, \Gamma_j, \Lambda_j, \Psi_j) = 0, \tag{19c}$$

$$m(t_1, y_1, z_1, u_1, \gamma_1) - m(T_s, Y_s, Z_s, U_s, \Gamma_s) = 0, \tag{19d}$$

$$h(t_1, y_1, z_1) - h(T_s, Y_s, Z_s) + \lambda_1 = 0, \tag{19e}$$

$$\begin{aligned} &r_t(t_1, q_1) + r_q(t_1, q_1)v(t_1, y_1, z_1) \\ &\quad - \big(r_t(T_s, Q_s) + r_q(T_s, Q_s)v(T_s, Y_s, Z_s)\big) + \psi_1 = 0, \end{aligned} \tag{19f}$$

which involve the numerical solution at the endpoint t_1, see (14). Denoting minus the left-hand side of equations (19) by r_1, one step of the simplified Newton method for these equations reads

$$(E + F)(\Delta x_1 - \Delta X_s) + J_d\Delta x_1 - h\sum_{m=1}^{m_{\max}}\big(\big(b^{\mathrm{T}} - e_s^{\mathrm{T}}A_m\big)\otimes J_m\big)\Delta\widetilde{X} = r_1, \tag{20}$$

where

$$\Delta\widetilde{X} := (\Delta X_1, \ldots, \Delta X_s)^{\mathrm{T}} \tag{21}$$

with ΔX_i for $i = 1, \ldots, s$ and Δx_1 as in (17), and

$$E := \begin{pmatrix} q_y & O & O & O & O & O \\ p_y & p_z & O & O & O & O \\ c_y & c_z & c_u & O & O & O \\ m_y & m_z & m_u & m_\gamma & O & O \\ h_y & h_z & O & O & O & O \\ r_q v_y & r_q v_z & O & O & O & O \end{pmatrix},$$

$$F := \begin{pmatrix} O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ \big(r_{tq} q_y + r_{qq}(q_y \cdot, v)\big) & O & O & O & O & O \end{pmatrix},$$

$$J_m := \begin{pmatrix} v_{my} & v_{mz} & O & O & O & O \\ f_{my} & f_{mz} & f_{mu} & f_{m\gamma} & f_{m\lambda} & f_{m\psi} \\ d_{my} & d_{mz} & d_{mu} & d_{m\gamma} & d_{m\lambda} & d_{m\psi} \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \end{pmatrix},$$

$$J_d := \begin{pmatrix} O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & I & O \\ O & O & O & O & O & I \end{pmatrix}.$$

The quantities y_1, z_1, u_1, γ_1 and Y_s, Z_s, U_s, Γ_s both approximate the exact solution $y(t)$, $z(t)$, $u(t)$, $\gamma(t)$ at $t = t_0 + h = t_1 = T_s$. By (19a)–(19d) they are close to each other provided stiff terms are treated by stiffly accurate RK coefficients. Hence, some terms in (20) can be neglected, leading to some sort of fixed-point iterations for y_1, z_1, u_1, γ_1. Since the variables λ_1, ψ_1 are dummy variables which are directly defined by (14h), (14i), they should have no influence on the numerical solution to the system of equations (14a)–(14f). Therefore, in (20) the term $J_d \Delta x_1$ and the components of r_1 corresponding to (19e), (19f) can be neglected. The dummy values λ_1, ψ_1 can be set explicitly after each modified Newton iteration such that the equations (14h), (14i) are satisfied exactly for the current iterate X^k. This means that the corresponding components of r_1 can be assumed to vanish. Provided stiff terms are treated by stiffly accurate coefficients, i.e., satisfying $a_{sj,m} = b_j$ for $j = 1, \ldots, s$, the linear equation (20) can be simplified, for example, to

$$-E \Delta X_s + E \Delta x_1 = r_1. \tag{22}$$

From (18) the whole set of equations (14) and variables can be reordered according to (15) such that after application of lemma 3 the corresponding approximate Jacobian L considered here is expressed as follows

$$L := Q_{s+1,s+1} \otimes E - h \begin{pmatrix} A_1 & 0_s \\ 0_s^{\mathrm{T}} & 0 \end{pmatrix} \otimes J_1 - h \begin{pmatrix} A_3 & 0_s \\ 0_s^{\mathrm{T}} & 0 \end{pmatrix} \otimes (J_0 + J_\Sigma) \tag{23}$$

where from (5d)

$$J_0 := \sum_{m=2}^{m_{\max}} \begin{pmatrix} O & O & O & O & O & O \\ O & O & O & O & f_{m\lambda} & f_{m\psi} \\ O & O & O & O & d_{m\lambda} & d_{m\psi} \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \end{pmatrix} = \begin{pmatrix} O & O & O & O & O & O \\ O & O & O & O & f_\lambda & f_\psi \\ O & O & O & O & d_\lambda & d_\psi \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \end{pmatrix}, \tag{24a}$$

$$J_1 := \begin{pmatrix} v_{1y} & v_{1z} & O & O & O & O \\ f_{1y} & f_{1z} & f_{1u} & f_{1\gamma} & O & O \\ d_{1y} & d_{1z} & d_{1u} & d_{1\gamma} & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \end{pmatrix}, \tag{24b}$$

$$J_\Sigma := \sum_{m=2}^{m_{\max}} \begin{pmatrix} v_{my} & v_{mz} & O & O & O & O \\ f_{my} & f_{mz} & f_{mu} & f_{m\gamma} & O & O \\ d_{my} & d_{mz} & d_{mu} & d_{m\gamma} & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \\ O & O & O & O & O & O \end{pmatrix}. \tag{24c}$$

The particular tensorial structure of the approximate Jacobian L can now be used to obtain good preconditioners. It will be discussed in section 5. It would not be possible to do so if the simplified Jacobian (18) was kept. Some other parts of the approximate Jacobian, such as h_y and $r_q v_y$, can actually be neglected without jeopardizing convergence of the inexact modified Newton iterates.

5. Preconditioning the approximate Jacobian

In this section the discussion is based on the approximate Jacobian L (23), though as previously mentioned, other choices are possible. Consider a linear system with matrix L

$$L\Delta X = R, \tag{25}$$

ΔX as in (17), and right-hand side R decomposed as

$$R = (R_1, \ldots, R_s, r_1)^{\mathrm{T}}.$$

5.1. Transforming the linear system

As a first step, the relation (22) can be introduced into the first s subequations of (25). A reduced linear system

$$\widetilde{L}\Delta\widetilde{X} = \widetilde{R} \tag{26}$$

is obtained for $\Delta\widetilde{X}$ given in (21) where

$$\widetilde{L} := I_s \otimes E - hA_1 \otimes J_1 - hA_3 \otimes (J_0 + J_\Sigma)\,,$$
$$\widetilde{R} := (R_1, \ldots, R_s)^{\mathrm{T}} - p_s \otimes r_1,$$

by using the decomposition (13) for $Q_{s+1,s+1}$ in (23).

As a second step, the quantities $\Delta\Lambda_i$ and $\Delta\Psi_i$ for $i = 1, \ldots, s$ can be replaced in (17) as follows

$$\begin{pmatrix} \Delta\Theta_1 \\ \vdots \\ \Delta\Theta_s \end{pmatrix} = h\left(A_3 \otimes I_{n_\lambda}\right) \begin{pmatrix} \Delta\Lambda_1 \\ \vdots \\ \Delta\Lambda_s \end{pmatrix}, \qquad \begin{pmatrix} \Delta\Omega_1 \\ \vdots \\ \Delta\Omega_s \end{pmatrix} = h\left(A_3 \otimes I_{n_\psi}\right) \begin{pmatrix} \Delta\Psi_1 \\ \vdots \\ \Delta\Psi_s \end{pmatrix}.$$

Hence, a linear system

$$\widetilde{K}\tilde{x} = \tilde{b} \tag{27}$$

is obtained with matrix

$$\widetilde{K} = I_s \otimes (E - J_0) - hA_1 \otimes J_1 - hA_3 \otimes J_\Sigma \tag{28}$$

where

$$E - J_0 = \begin{pmatrix} q_y & O & O & O & O & O \\ p_y & p_z & O & O & -f_\lambda & -f_\psi \\ c_y & c_z & c_u & O & -d_\lambda & -d_\psi \\ m_y & m_z & m_u & m_\gamma & O & O \\ h_y & h_z & O & O & O & O \\ r_q v_y & r_q v_z & O & O & O & O \end{pmatrix}.$$

Under the assumptions (2) the matrix $E - J_0$ is invertible. For nonstiff DAEs the terms $-hA_1 \otimes J_1$ and $-hA_3 \otimes J_\Sigma$ in (28) can be neglected.

Once an approximation to the solution of the linear system (26) is obtained, it remains to define an approximation to Δy_1, Δz_1, Δu_1, $\Delta\gamma_1$. The relation (20) or (22) can be used for that purpose. The current iterates for λ_1, ψ_1 can be defined directly from (14h), (14i), but this need not be done explicitly.

The linear system (25) with approximate Jacobian L (23) has been transformed under a form such that preconditioning the linear equations (27) can now be done in a straightforward manner following the results of [10–12]. The preconditioner developed in [10,11] will not be described here. The main drawback of this preconditioner is the

fact that although its decomposition is parallelizable, the solution of the linear systems involved is not. Instead, a new truly parallel preconditioner is presented in the next subsection.

5.2. A truly parallel preconditioner

In this subsection some recent results of [12] are followed and briefly presented. The linear system (27) is solved approximately by application of linear iterative methods with a preconditioner $\widetilde{Q} \approx \widetilde{K}^{-1}$ of the form

$$\widetilde{Q} := \widetilde{H}^{-1} \widetilde{G} \widetilde{H}^{-1}$$

where

$$\widetilde{H} := I_s \otimes (E - J_0) - h\Gamma_1 \otimes J_1 - h\Gamma_3 \otimes J_\Sigma,$$
$$\widetilde{G} := I_s \otimes (E - J_0) - h\Omega_1 \otimes J_1 - h\Omega_3 \otimes J_\Sigma.$$

The coefficients matrices Γ_1 and Γ_3 are chosen to be diagonal

$$\Gamma_1 := \operatorname{diag}(\gamma_{1,1}, \gamma_{2,1}, \ldots, \gamma_{s,1}), \qquad \Gamma_3 := \operatorname{diag}(\gamma_{1,3}, \gamma_{2,3}, \ldots, \gamma_{s,3})$$

with $\gamma_{1,1} = 0$ because of (7a), leading to

$$\widetilde{H}^{-1} = \begin{pmatrix} H_1^{-1} & & & O \\ & H_2^{-1} & & \\ & & \ddots & \\ O & & & H_s^{-1} \end{pmatrix}$$

where

$$H_i := (E - J_0) - h\gamma_{i,1} J_1 - h\gamma_{i,3} J_\Sigma \quad \text{for } i = 1, \ldots, s.$$

The matrices H_i can be decomposed independently, hence in parallel. Solving a linear system with matrix $\widetilde{H}$ can also be done in parallel since it is block-diagonal. This is the main advantage of this preconditioner compared to the one presented in [10,11].

The coefficients of Γ_1, Γ_3, Ω_1, and Ω_3 still remain to be fixed to some values. Assuming the coefficients of Γ_1 and of Γ_3 to be given, the coefficients of Ω_1 and Ω_3 are taken as

$$\Omega_1 = \begin{pmatrix} 0 & 0_s^{\mathrm{T}} \\ -\widehat{\Gamma}_1 \widehat{A}_1^{-1} \widehat{A}_{1,1} & \widehat{\Gamma}_1 \widehat{A}_1^{-1} \widehat{\Gamma}_1 \end{pmatrix}, \qquad \Omega_3 = \Gamma_3 A_3^{-1} \Gamma_3$$

where

$$\widetilde{A}_1 = \begin{pmatrix} \widehat{A}_{1,1} & \widehat{A}_1 \end{pmatrix}, \qquad \widehat{\Gamma}_1 := \operatorname{diag}(\gamma_{2,1}, \ldots, \gamma_{s,1})$$

with $\widetilde{A}_1$ as in (8). The coefficients matrices Ω_1 and Ω_3 have been separately determined such that the preconditioner $\widetilde{Q}$ is asymptotically correct when considering the Dahlquist test equation

$$y' = \lambda y, \quad \operatorname{Re}(\lambda) \leqslant 0.$$

The coefficients $\gamma_{i,1}$ for $i = 2, \ldots, s$ and $\gamma_{i,3}$ for $i = 1, \ldots, s$ are free. They are required to satisfy $\gamma_{1,i} > 0$ for $i = 2, \ldots, s$ and $\gamma_{i,3} > 0$ for $i = 1, \ldots, s$ which is a natural assumption to ensure the invertibility of the matrices H_i. These coefficients can be chosen, for example, to minimize

$$\max_{\mathrm{Re}(z)\leqslant 0} \Big(\max_{i=1,\ldots,s} \big|\lambda_i\big(M(z)\big) - 1\big| \Big)$$

where $z = h\lambda$, $M(z) = \widetilde{Q}(z)\widetilde{K}(z)$, and $\lambda_i(M(z))$ for $i = 1, \ldots, s$ are the s eigenvalues of $M(z)$. When $\gamma_{i,1} = \gamma_1$ for $i = 2, \ldots, s$ and $\gamma_{i,3} = \gamma_3$ for $i = 1, \ldots, s$ only one or two matrix decompositions beside $E - J_0$ are needed. This can be quite advantageous on a serial computer. The cost of computing a matrix–vector product $\widetilde{Q}v$ with at least s processors on a parallel computer consists of one decomposition of H_i on each processor, two linear systems with matrix H_i to be solved, one local matrix–vector product with each matrix $E - J_0$, hJ_1, and hJ_Σ, and some communication between processors according to the nonzero elements of the coefficients matrices Ω_1 and Ω_3.

6. Conclusion

The approximation of a certain class of DAEs by SPARK methods has been considered. The main difficulty of these methods resides in finding a way to implement them efficiently. Certain linear transformations are applied to the resulting system of nonlinear equations such that an efficient preconditioner to the linear systems of the modified Newton method can be constructed. These linear transformations rely heavily on specific properties of the SPARK coefficients. An approximate Jacobian of the system of nonlinear equations is presented. Based on this approximate Jacobian a new truly parallelizable preconditioner is given.

References

[1] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential–Algebraic Equations*, SIAM Classics in Applied Mathematics, 2nd ed. (SIAM, Philadelphia, PA, 1996).

[2] P.N. Brown, A.C. Hindmarsh and L.R. Petzold, Using Krylov methods in the solution of large-scale differential–algebraic systems, SIAM J. Sci. Comput. 15 (1994) 1467–1488.

[3] G. Golub and C.F. van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Science, 3rd ed. (Johns Hopkins Univ. Press, Baltimore/London, 1996).

[4] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers in Applied Mathematics, Vol. 17 (SIAM, Philadelphia, PA, 1997).

[5] E. Hairer, Ch. Lubich and M. Roche, *The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods*, Lecture Notes in Mathematics, Vol. 1409 (Springer, Berlin, 1989).

[6] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential–Algebraic Problems*, Comput. Math., Vol. 14, 2nd revised ed. (Springer, Berlin, 1996).

[7] E.J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol. I: Basic Methods* (Allyn and Bacon, Boston, USA, 1989).

[8] L.O. Jay, Runge–Kutta type methods for index three differential–algebraic equations with applications to Hamiltonian systems, Ph.D. thesis, Department of Mathematics, University of Geneva, Switzerland (1994).
[9] L.O. Jay, Structure preservation for constrained dynamics with super partitioned additive Runge–Kutta methods, SIAM J. Sci. Comput. 20 (1998) 416–446.
[10] L.O. Jay, A parallelizable preconditioner for the iterative solution of implicit Runge–Kutta type methods, J. Comput. Appl. Math. 111 (1999) 63–76.
[11] L.O. Jay, Inexact simplified Newton iterations for implicit Runge–Kutta methods, SIAM J. Numer. Anal. 38 (2000) 1369–1388.
[12] L.O. Jay, Preconditioning and parallel implementation of implicit Runge–Kutta methods, Technical Report, Dept. of Math., University of Iowa, USA (2001) in preparation.
[13] P.J. Rabier and W.C. Rheinboldt, *Nonholonomic Motion of Rigid Mechanical Systems from a DAE Viewpoint* (SIAM, Philadelphia, PA, 2000).
[14] Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS, Boston, MA, 1996).
[15] Y. Saad and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[16] W.O. Schiehlen, ed., *Multibody Systems Handbook* (Springer, Berlin, 1990).
[17] W.O. Schiehlen, ed., *Advanced Multibody System Dynamics, Simulation and Software Tools* (Kluwer Academic, London, 1993).