Quasi Stage Order Conditions for SDIRK Methods

Frank Cameron [a], Mikko Palmroth [b], Robert Piché [b]

[a] *Pori School of Technology and Economics, P.O. Box 300, 28101 Pori, FINLAND*

[b] *Tampere University of Technology, P.O. Box 692, 33101 Tampere, FINLAND*

Abstract

The stage order condition is a simplifying assumption that reduces the number of order conditions to be fulfilled when designing a Runge Kutta (RK) method. Because a DIRK (diagonally implicit RK) method cannot have stage order greater than 1, we introduce quasi stage order conditions and derive some of their properties for DIRKs. We use these conditions to derive a low-order DIRK method with embedded error estimator. Numerical tests with stiff ODEs and DAEs of index 1 and 2 indicate that the method is competitive with other RK methods for low accuracy tolerances.

Key words: Differential-algebraic systems, Runge-Kutta methods

1 Introduction

Of all the classes of implicit Runge-Kutta methods (*IRK*s), diagonally implicit RK methods (*DIRK*s) are arguably the easiest to implement. In the past however, when DIRKs have been compared with other IRKs in tests on DAEs, they have performed poorly [9,13]. One obvious reason for this is that DIRKs tested were not originally designed for DAEs. In particular, they were not designed to attain a certain order for DAEs.

To attain a certain order RK methods must satisfy order conditions. The stage order property is useful since it reduces the number of distinct order conditions. Other classes of IRKs used for solving DAEs usually have some stage order property as an integral part of their design [1,2,8]. However, DIRKs can only have the lowest stage order possible. This makes solving the order conditions for DIRKs an unappealing task. To remedy this situation for DIRKs, we use the concept of quasi stage order.

The idea behind quasi stage order is not new. Although they did not give it a name, Cooper and Sayfy [5] used it in developing DIRKs for ODEs. Verner [16] called it dominant stage order and used it to develop high-order explicit Runge-Kutta methods for ODEs. Our contribution is to extend the idea for use in developing DIRKs for DAEs.

The paper is organised as follows. In section 2 we give background notation and basic results. The quasi stage order concepts are defined in section 3, and various results for DIRKs are given. A DIRK that was derived using quasi stage order is presented in section 4, together with some numerical test results.

2 Notation

2.1 DAEs

We are mainly interested in finding the numerical solution of the implicit index 1 differential algebraic equation (DAE) initial value problem

$$F(y',\, y) = 0\,, \quad y(t_0) = y_0\,, \quad y'(t_0) = y'_0 \tag{1}$$

where $y : \mathbf{R} \to \mathbf{R}^N$ and $F : \mathbf{R}^N \times \mathbf{R}^N \to \mathbf{R}^N$. Although this DAE is autonomous, the discussion and results apply equally for non-autonomous DAEs. We assume that y is sufficiently differentiable. Kværnø [9] gives conditions for DAE (1) to be of index 1. A special case of (1) is the quasi-linear DAE

$$B(y)y' - f(y) = 0\,, \tag{2}$$

where $B(y)$ is singular.

We will also be interested in solving the index 2 DAE initial value problem

$$y' = f(y, z)\,, \quad y(t_0) = y_0 \in \mathbf{R}^n \tag{3a}$$
$$0 = g(y)\,, \quad z(t_0) = z_0 \in \mathbf{R}^m \tag{3b}$$

This DAE has been widely researched [1,6,8]. It is known that the same Runge Kutta (RK) method order conditions are valid for both (1) and for the y component of (3) [6, pg. 506]. Thus if we have a candidate method for solving (1), then we can use it for solving for the y component of (3), assuming the method satisfies any other requirements arising from (3).

This section contains some RK terminology. Hairer and Wanner [6] should be consulted for more details.

The *stage variables* $Y_i, i = 1, 2, \ldots s$ and the *stage derivatives* $Y'_j, j = 1, 2, \ldots s$ of an s-stage RK method are related by

$$Y_i \;=\; y_n + h \sum_{j=1}^{s} a_{ij} Y'_j \;,\; i = 1, 2, \ldots s \tag{4}$$

In (4) h is the *stepsize*. When (4) is applied to (1) the equation for stage i is

$$F(Y'_i,\; y_n + h \sum_{j=1}^{s} a_{ij} Y'_j \,) = 0 \;,\;\; i = 1, \ldots, s \tag{5}$$

Once all the stage derivatives have been computed the state can be updated using

$$y_{n+1} \;=\; y_n + h \sum_{i=1}^{s} b_i Y'_i \tag{6}$$

The parameters of an RK method are the matrix $A \in \mathbf{R}^{s \times s}$ of (4) and the vector $b \in \mathbf{R}^{s \times 1}$ of (6). A *FIRK* (fully implicit RK method) has an invertible A. We will use $W \triangleq A^{-1}$. A *DIRK* (diagonally implicit RK method) is a FIRK whose A is lower triangular. An *SDIRK* (singularly diagonally implicit RK method) is a DIRK whose A has one real s-fold eigenvalue.

We let $\breve{y}(t_{n+1})$ represent the exact solution of (1) after one time step when starting from consistent initial conditions $\{\, t_n, y_n, y'_n \,\}$. The *local truncation error* (LTE) for the RK solution of (6) and (5) is

$$\delta y_{n+1} \triangleq y_{n+1} - \breve{y}(t_{n+1}) \tag{7}$$

The *local order* of the RK method is κ_L if

$$\delta y_{n+1} = O(h^{\kappa_L}) \tag{8}$$

The LTE can be expressed as

$$\delta y_{n+1} = \sum_{i=\kappa_L}^{\infty} h^i \left(\sum_{j=1}^{\chi_i} T_{ij}\, D_{ij} \right) \;, \tag{9}$$

where the D_{ij} are elementary differentials or sums and products of partial differentials of F. The *truncation error coefficients* (TECs) T_{ij} are functions of

A, b and c. For an RK method to have local order κ_L we must have $T_{ij} = 0, \forall j$, $i = 1, \ldots, \kappa_L - 1$. Typically however, an *order condition* is presented as some multiple of its corresponding TEC, e.g. $\alpha\, T_{ij} = 0$, where α makes the order condition neater.

We define the *abscissae* $c \in \mathbf{R}^{s\times 1}$ of an RK method by $c \triangleq A e_s$, where e_s is a s-vector of ones. A *non-confluent* RK method is one with distinct c_i.

In an *embedded RK pair* there is a second $\hat{b} \in \mathbf{R}^{s\times 1}$ vector. The local order associated with the $(A, \hat{b})$-RK method is $\hat{\kappa}_L$. Assuming $\hat{\kappa}_L$ and κ_L are different, then we can estimate the LTE from

$$| \epsilon_{n+1} | \triangleq | h \sum_{i=1}^{s} (b_i - \hat{b}_i)\, Y_i' | \approx |\delta y_{n+1}| \tag{10}$$

We assume that the (A, b)-RK method is used for updating the state via (6). We refer to the (A, b) and $(A, \hat{b})$-RK methods as the *updating* and the *auxiliary* methods respectively.

There is a one-to-one correspondence between order conditions and certain trees. The *order* of tree t is denoted $\rho(t)$. An RK method has local order κ_L when all order conditions are satisfied for trees having $\rho(t) \leq \kappa_L - 1$. Table 1 contains the trees for $\rho(t) \leq 4$. The notation $t = [t_1,\ \ldots,\ t_k,\ u_1, \ldots, u_\ell]_y$ means the tree obtained by connecting the roots of $t_1, \ldots, t_k, u_1, \ldots, u_\ell$ to a new light vertex, which becomes the root of tree t. Analogously, $u = [t_1, \ldots, t_k]_z$ indicates the tree obtained by connecting the roots of $t_1, \ldots, t_k$ to a new heavy vertex, which becomes the root of tree u. Further details on trees and their corresponding order conditions can be found from Kværnø [9].

3 Quasi-stage order

3.1 Definitions

In the literature the following condition is typically associated with stage order:

$$k\, A\, c^{k-1} = c^k\ ,\ k = 2, 3, \ldots\, q \tag{11}$$

We will say an RK method has *complete* stage order q, denoted $C(q)$, when (11) is satisfied. Note that complete stage order requires something from all rows of A, and depends only on A (because $c = A\, e_s$), not on b.

It is known that a DIRK cannot have $C(q)$ for $q \geq 2$. However, a DIRK may enjoy some quasi stage order properties, which are defined as follows.

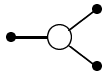
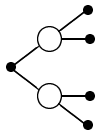
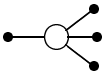
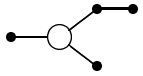
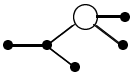
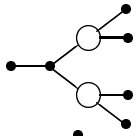
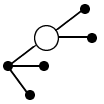
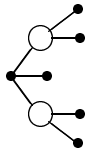
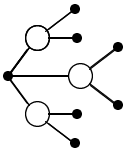
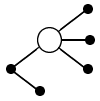
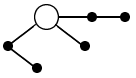
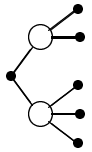
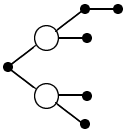
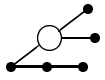
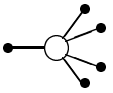
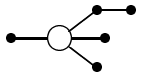
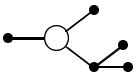
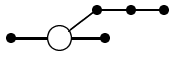
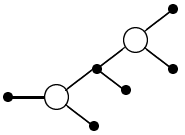
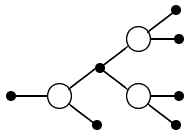
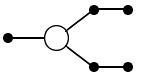
tree	$\rho(t)$	graph	tree	$\rho(t)$	graph
t_1	1		t_2	2	
t_3	2		t_4	3	
t_5	3		t_6	3	
t_7	3		t_8	3	
t_9	3		t_{10}	4	
t_{11}	4		t_{12}	4	
t_{13}	4		t_{14}	4	
t_{15}	4		t_{16}	4	
t_{17}	4		t_{18}	4	
t_{19}	4		t_{20}	4	
t_{21}	4		t_{22}	4	
t_{23}	4		t_{24}	4	
t_{25}	4		t_{26}	4	
t_{27}	4		t_{28}	4	
t_{29}	4		t_{30}	4	

Table 1
The order trees up to $\rho(t) = 4$. The root vertex is labelled by ξ_r.

Definition 1 *Stage i of an RK method has* individual forward stage order ℓ_i, *or* $\tilde{C}(\ell_i)$, *when*

$$k\, A_{i\bullet}\, c^{k-1} = c_i^k, \quad k = 2, 3, \ldots \ell_i$$

Definition 2 *An RK method has* forward quasi stage order q, *or* $\tilde{C}(q)$, *when* $\tilde{C}(\ell_i)$, $\ell_i \geq q$ *for every stage where* $b_i \neq 0$.

Note that forward quasi stage order requires something only from the rows of A where $b_i \neq 0$, and depends not only on A but also on (the sparsity pattern of) b.

For a FIRK, condition (11) is equivalent to

$$W\, c^k = k\, c^{k-1}, \; k = 2, 3, \ldots q \tag{12}$$

This can be used as the basis for another form of quasi stage order.

Definition 3 *Stage i of an RK method has* individual reverse stage order ℓ_i, *or* $\check{C}(\ell_i)$, *when*

$$W_{i\bullet}\, c^k = k\, c_i^{k\,-\,1}, \quad k = 2, 3, \ldots \ell_i$$

Definition 4 *An RK method has* reverse quasi stage order q, *or* $\check{C}(q)$, *when* $\check{C}(\ell_i)$, $\ell_i \geq q$ *for every stage where* $b_i \neq 0$.

Obviously, complete stage order implies both forward and backward quasi stage order. However, forward quasi stage order does not in general imply reverse quasi stage order or individual reverse stage order.

As we shall see later, the number of distinct order conditions can be reduced by requiring $\tilde{C}(q)$ or $\check{C}(q)$, and DIRKs can have $\tilde{C}(q)$ and/or $\check{C}(q)$ for $q \geq 2$. However, both forward and reverse quasi stage order need their own set of conditions, which is in contrast to complete stage order, where (11) and (12) are equivalent.

3.2 DIRKs and quasi stage order 2

In this section we present some results that pertain to quasi stage order 2. All proofs are relegated to the appendix.

We will use the following partition:

$$A = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix}, \quad W = \begin{bmatrix} W_1 & 0 \\ W_2 & W_3 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \beta_2 \end{bmatrix}, \quad c = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{13}$$

where $A_1 \in \mathbf{R}^{r\times r}$, $A_3 \in \mathbf{R}^{(s-r)\times(s-r)}$, and the other matrices are appropriately dimensioned.

Row 2 of a DIRK can have either $\tilde{C}(2)$ or $\check{C}(2)$, but not both. This restriction can be seen when we parameterize the first 2 rows of A and W as follows:

$$\begin{bmatrix} A_{1\bullet} \\ A_{2\bullet} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \ \ldots \ 0 \\ c_2 - a_{22} & a_{22} & 0 \ \ldots \ 0 \end{bmatrix}$$

$$\begin{bmatrix} W_{1\bullet} \\ W_{2\bullet} \end{bmatrix} = \begin{bmatrix} 1/a_{11} & 0 & 0 \ \ldots \ 0 \\ (a_{22} - c_2)/(a_{11}\, a_{22}) & 1/a_{22} & 0 \ \ldots \ 0 \end{bmatrix}$$

With this parameterization the conditions for $\tilde{C}(2)$ and $\check{C}(2)$ for row 2 are

$$c_2^2/2 = (c_2 - a_{22})\, a_{11} + a_{22}\, c_2 \tag{14}$$

$$2\, c_2\, a_{11}\, a_{22} = (a_{22} - c_2)\, c_1^2 + a_{11}\, c_2^2 \tag{15}$$

It can be shown that all solutions of (14)–(15) imply a singular A and hence are not permissible.

Another restriction is the following.

Lemma 5 *If a DIRK is stiffly accurate, has forward quasi stage order $\tilde{C}(2)$ and has an A matrix whose last row is*

$$A_{s\bullet} = \begin{bmatrix} 0 & 0 & \ldots 0 & 1 - a_{ss} & a_{ss} \end{bmatrix} ,$$

then this DIRK cannot have $\kappa_L = 5$, even for ODEs.

Both forward and reverse quasi stage order make redundant some of the order conditions corresponding to the trees of Table 1. If a DIRK has $\tilde{C}(2)$, then the redundant order conditions are those corresponding to trees having the form $t = [t_2, t(\eta)]_y$ where t_2 is given in Table 1 and $t(\eta)$ are all other branches in the remainder of the tree. The order condition for this tree has the form

$$b^T\, [\, A\, c \odot \eta\,] = 1/\lambda(t)$$

From defn. 2 and (13) we can rewrite this order condition as

$$\begin{bmatrix} 0 & \beta_2^T \end{bmatrix} \left(\begin{bmatrix} A_1\, \gamma_1 \\ \gamma_2^2/2 \end{bmatrix} \odot \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \right) \;=\; \beta_2^T\, [\, (\gamma_2^2/2) \odot \eta_2\,] \;=\; \frac{1}{2}\, b^T (c^2 \odot \eta) = 1/\lambda(t)$$

This expression corresponds to the tree $t = [t_1, t_1, t(\eta)]_y$.

to attain local order κ_L	number distinct order conditions when RK method has properties				
	no properties	stiff acc.	$\tilde{C}(2)$ and stiff acc.	$\check{C}(2)$ and stiff acc.	$\tilde{C}(2)$, $\check{C}(2)$ and stiff acc.
4	9	6	4	3	2
5	30	20	16	11	9

Table 2
The number of distinct order conditions given various properties.

The order conditions made redundant by reverse quasi stage order 2, $\check{C}(2)$, are those whose tree has the form $t = [u_1, t(\eta)]_y$, where

$u_1 =$ 

and $t(\eta)$ are all other branches in the remainder of the tree. The order condition for such a tree has the form

$$b^T \left[W\, c^2 \odot \eta \right] = 1/\lambda(t)$$

From defn. 4 and (13) we can rewrite this order condition as

$$\begin{bmatrix} 0 & \beta_2^T \end{bmatrix} \left(\begin{bmatrix} W_1\, \gamma_1^2 \\ 2\,\gamma_2 \end{bmatrix} \odot \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \right) = \beta_2^T \left[(2\,\gamma_2) \odot \eta_2 \right] = 2b^T (c \odot \eta) = 1/\lambda(t)$$

This expression corresponds to the tree $t = [t_1, t(\eta)]_y$.

If we combine forward or reverse quasi stage order with stiff accuracy, we can further reduce the number of distinct order conditions. Stiff accuracy makes redundant all order conditions corresponding to trees of the form $\left[[t_1, \ldots, t_k]_z \right]_y$. Table 2 gives the number of distinct order conditions for various situations. In particular, if a DIRK is required to have stiff accuracy, $\tilde{C}(2)$, and $\check{C}(2)$, then of the 30 trees of Table 1 all but the following 9 are redundant: t_1, t_4, t_{10}, t_{12}, t_{13}, t_{14}, t_{15}, t_{19} and t_{20}.

3.3 General results on quasi stage order

In this section we present some general results on quasi stage order.

It follows directly from defn. 2 that if a DIRK is stiffly accurate and has $\tilde{C}(q)$,

then it satisfies the order conditions

$$b^T c^k = 1/(k+1)\,,\ k = 1, 2, \ldots q$$

Similar order conditions arise from DAEs:

$$b^T W c^k = 1$$

However, any stiffly accurate FIRK will satisfy all order conditions of this latter form; we need not require reverse quasi stage order $\check{C}(q)$.

The next lemma shows that separate conditions for $\tilde{C}(\ell)$ and $\check{C}(\ell)$ are not needed for row s of A.

Lemma 6 *If a DIRK is stiffly accurate, has $\check{C}(\ell)$ and the partition of (13) holds, then row s of the DIRK has $\tilde{C}(\ell)$.*

The next two lemmas give sufficient conditions for SDIRKs to have individual forward and reverse stage order.

Lemma 7 *Let $a_{i2}, a_{i3}, \ldots, a_{iq}$ be free parameters in $A_{i\bullet}$ of a non-confluent SDIRK. By setting these parameters row i can have $\tilde{C}(q)$ for $i > q$.*

Lemma 8 *Assume that for all i, $c_i \neq 0$. Let $w_{i2}, w_{i3}, \ldots, w_{iq}$ be free parameters in $W_{i\bullet}$ of a non-confluent SDIRK. By setting these parameters row i can have $\check{C}(q)$ for $i > q$.*

In the proof of lemma 7 there were no restrictions on the elements of $A_{i\bullet}$. However, in a stiffly accurate SDIRK $A_{s\bullet}$ is constrained by partition (13), i.e. the first r elements of $A_{s\bullet}$ are zero. The following lemma shows what is attainable for row s in this situation.

Lemma 9 *Let $a_{s,r+2}, a_{s,r+3}, \ldots, a_{ss}$ be free parameters in $A_{s\bullet}$ of a non-confluent, stiffly accurate SDIRK. By setting these parameters row s can have $\tilde{C}(q)$ for $s - r \geq q$.*

Lemma 10 *Assume that for all i, $c_i \neq 0$. Let $w_{s,r+2}, w_{s,r+3}, \ldots, w_{ss}$ be free parameters in $W_{s\bullet}$ of a non-confluent stiffly accurate SDIRK. By setting these parameters row s can have $\check{C}(q)$ for for $s - r \geq q$.*

Combining lemmas 7 and 9 we get

Theorem 11 *An s-stage stiffly accurate SDIRK can have $\tilde{C}(q)$ for $s = 2\,q$.*

Similarly, we can combine lemmas 8 and 10 to get

Theorem 12 *An s-stage stiffly accurate SDIRK can have $\check{C}(q)$ for $s = 2\,q$.*

We expect that in practice individual stage orders for SDIRKs will never need to exceed 3. We have found it possible to attain both $\tilde{C}(p)$ and $\check{C}(q)$ for row i when $p+q-1 \leq i$, $1 \leq p \leq 3$ and $1 \leq q \leq 3$. For $p+q-1 = i$ however, we must assume there is one free parameter from some row j, $j < i$.

4 A new SDIRK embedded pair

We used quasi stage order in designing the new SDIRK pair, denoted SDIRK2, given by the following Butcher table:

$$\begin{array}{c|cccc} 1/4 & 1/4 & 0 & 0 & 0 \\ 11/28 & 1/7 & 1/4 & 0 & 0 \\ 1/3 & 61/144 & -49/144 & 1/4 & 0 \\ 1 & 0 & 0 & 3/4 & 1/4 \\ \hline b^T & 0 & 0 & 3/4 & 1/4 \\ \hat{b}^T & -61/600 & 49/600 & 79/100 & 23/100 \end{array} \tag{16}$$

We required the updating SDIRK (A, b) to be stiffly accurate, which implies $g(y_n) = 0$ for (3), that is, the algebraic constraints are satisfied at the end of each RK step. We also required both $\tilde{C}(2)$ and $\check{C}(2)$. After satisfying the two distinct order conditions remaining (see Table 2), we obtained an updating SDIRK with $\kappa_L = 4$. The auxiliary $(A, \hat{b})$ SDIRK has $\hat{\kappa}_L = 3$, so the local error can be estimated from (10). The updating (A, b) SDIRK is L-stable; this removes one source error in the error propagation error of FIRKs [7, Thm. 4.4] and in addition implies A-stability. The auxiliary $(A, \hat{b})$ SDIRK is A-stable.

We also considered the following quality measures for RK methods.

- Cameron [3] suggests that to obtain a good local error estimate for stiff problems, $\chi(-\infty)$ should be small and $\gamma(-\infty)$ should be in the range $(0.1, 1)$, where

$$\gamma(z) \triangleq |\, \exp(z) - R(z) \,| / |\, \hat{R}(z) - R(z) \,| \tag{17a}$$

$$\chi(z) \triangleq |\, \hat{R}(z) - R(z) \,| \tag{17b}$$

  and $R(z) \triangleq 1 + zb^T(I - zA)^{-1}e_s$ is the standard stability function.
- Shampine [15, pp. 374-5] suggests that to obtain a good local error estimate for stiff problems, the measures

$$v_1 \triangleq \| T_{\kappa_L+1,\bullet} \| / \| T_{\kappa_L,\bullet} \| \tag{18}$$

pair	$\chi(-\infty)$	$\gamma(-\infty)$	υ_1	υ_2	$\| \Delta d^T \|_2$
SDIRK1	0	0.12	1.0	1.5	1.4
SDIRK2	0.39	0	1.7	1.8	1.9

Table 3
Performance measures for the SDIRK pairs

$$\upsilon_2 \triangleq \| \widehat{T}_{\kappa_L+1,\bullet} - T_{\kappa_L+1,\bullet} \| / \| T_{\kappa_L,\bullet} \| \tag{19}$$

should be small. We choose to use the ∞-norm.

- Nørsett and Thomsen [11] suggest that if

$$\| \Delta d^T \| \triangleq \| (b^T - \hat{b}^T) A^{-1} \| \tag{20}$$

is small, then a more lax stopping criterion may be used with the iterative (e.g. Newton) solver, so that less work is needed in this part of the method.

Table 3 contains the values of these performance measures for SDIRK2. For comparison, we show the performance measures for the four stage method 2b from [4], here denoted SDIRK1. These measures all favour SDIRK1. However, numerical test results presented later tend to favour SDIRK2.

However, for solving the index 2 DAE (3), SDIRK2 is preferable, because it can at least attain local order $\kappa_L = 2$ for z, whereas SDIRK1 cannot. There is one order condition needed to obtain $\kappa_L = 2$ for z:

$$b^T W^2 c^2 = 2$$

The (A, b) SDIRK in SDIRK2 satisfies this; neither SDIRK in SDIRK1 does.

5 Tests

Numerical tests on six small ODE and DAE problems (Table 4) were performed using Olsson's C++ solver package Godess [12] running under Windows NT on an Intel PIII CPU. Most of the examples are stiff ODE or DAE benchmark problems from the literature [6,10]. The hydraulics examples are small nonlinear circuits that are modeled using the technique described in [14], which gives a sparse set of 23 DAEs of index 1. The Godess default parameter values were used throughout.

We solved these six problems using 4 different RK methods: SDIRK1 and SDIRK2 from section 4, the 5 stage SDIRK pair from from [6, pg. 100] and the RadauIIa pair from [6]. The 5 stage SDIRK pair — we refer to it as SDIRKHW — was designed for ODEs, not DAEs. For ODEs SDIRKHW has

name	number of states	description, reference	comments
Transistor amplifier	8	stiff, index-1 DAE [10]	
Robertson's DAE	3	index-1 DAE [6]	used final time 100 s; replaced third ODE in [6] with $y_1 + y_2 + y_3 = 1$
Ring modulator DAE	15	stiff, index-2 DAE [10]	ignored index-2 variables in error estimation
Ring modulator ODE	15	stiff ODE [10]	
Hydraulic circuit, stiff	23	stiff, index-1 DAE [14]	
Hydraulic circuit, nonstiff	23	nonstiff, index-1 DAE [14]	

Table 4
Description of test problems.

local orders 5(4), whereas for DAE (2) it has local orders 3(2). We set up Godess parameter files for implementing SDIRK1, SDIRK2 and SDIRKHW. Godess comes with its own files for implementing RadauIIa. These RadauIIa files however use the same values as given in [6, pgs. 74,123]. We used Godess because our purpose in these tests is to compare different RK methods in the same environment. However, other codes implementing the same methods, e.g. RADAU5 [6], may yield very different performance results. Our conclusions about these RK methods pertain only to their implementation in Godess.

The work-accuracy diagrams of Fig. 1 reveal that of the four RK methods, no one is consistently the best nor is one consistently the worst. Even though it is not designed for DAEs, SDIRKHW performs well on most of the DAE problems. SDIRK1 and SDIRK2 have similar performance, usually SDIRK2 is slightly better. Since RadauIIa has a higher order than the SDIRKs, one would expect its performance to improve relative to the SDIRKs when higher accuracies are demanded. The results in general confirm this expectation. However, for low accuracy demands the performance of SDIRK2 is usually comparable to or better than that of RadauIIa.

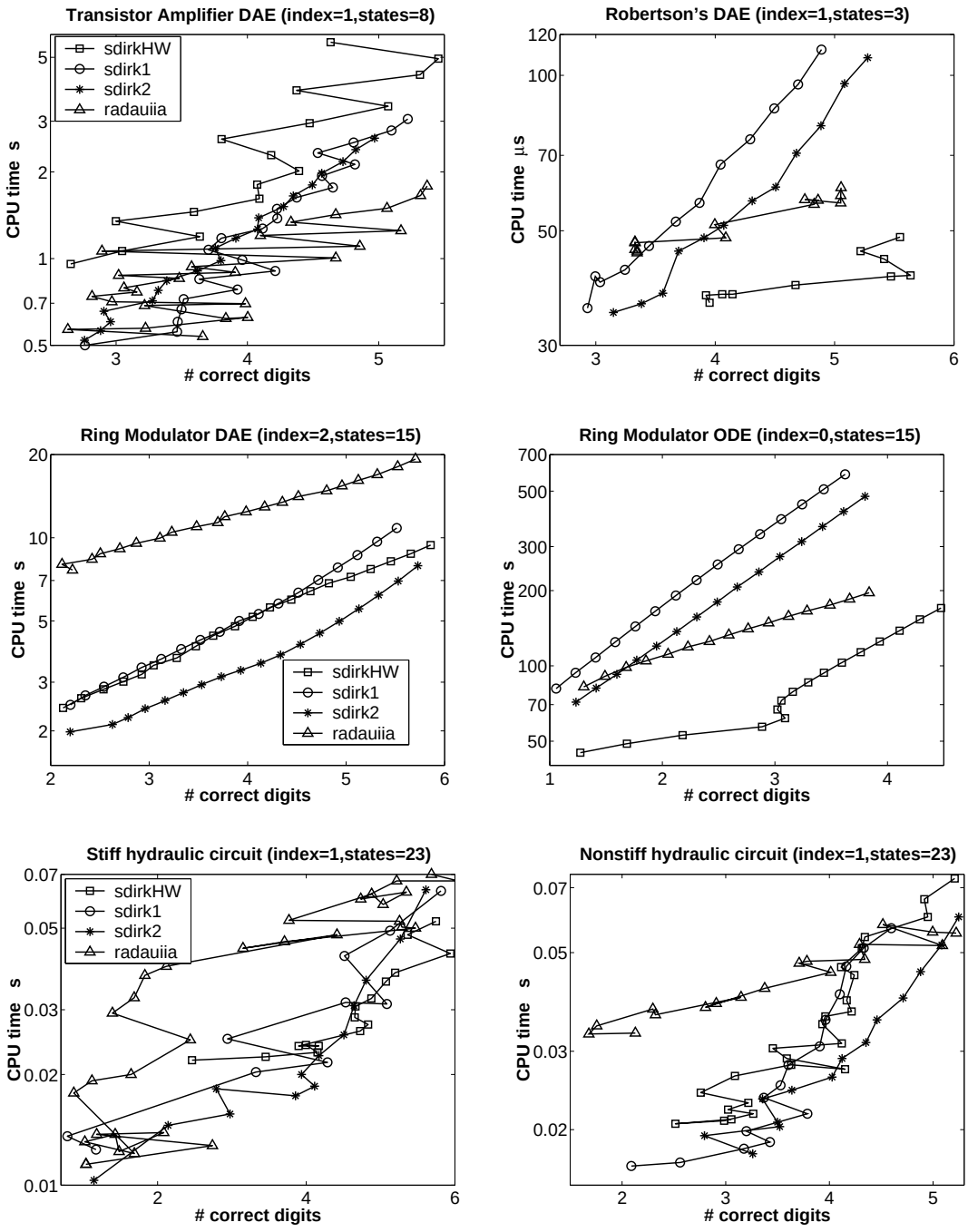


Fig. 1. Work-accuracy diagrams

References

[1] A. Aubry and P. Chartier, On improving the convergence of Radau IIA methods applied to index 2 DAEs. *SIAM J. Numer. Anal.* **35** (1998) 1347-1367.

[2] J. C. Butcher and R. P. K. Chan, Efficient Runge-Kutta integrators for index 2 differential algebraic equations. *Math. Comp.*, **67** (1998) 1001-21.

[3] F. Cameron, A class of low order DIRK methods for a class of DAEs, *Appl. Numer. Math.*, **31** (1999) 1–16.

[4] F. Cameron, Low-order Runge-Kutta methods for differential-algebraic equations, Ph.D. thesis, Tampere University of Technology, Tampere, Finland (1999).

[5] G. J. Cooper and A. Sayfy, Semi-explicit A-stable Runge-Kutta Methods, *Math. Comput.*, **33**, (1979) 541-56.

[6] E. Hairer and G. Wanner, *Solving ordinary differential equations, Vol II, Stiff and differential-algebraic problems* (Springer, Berlin, 1996).

[7] E. Hairer, C. Lubich and M. Roche, *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Lecture Notes in Math. 1409, (Springer, Berlin, 1989).

[8] L. Jay, Convergence of a class of Runge-Kutta methods for differential algebraic systems of index 2. *BIT*, **33** (1993) 137-50.

[9] A. Kværnø, Runge-Kutta methods applied to fully implicit differential-algebraic equations of index 1. *Math. Comput.*, **54** (1990) 583–625.

[10] W. M. Lioen and J. J. B. de Swart, Test set for initial value problem solvers, CWI Report MAS-R9832, Amsterdam, The Netherlands, (1998), `http://www.cwi.nl/cwi/projects/IVPtestset/`

[11] S. P. Nørsett and P. G. Thomsen, Local error control in SDIRK-methods, *BIT*, **26**, (1986), 100-13.

[12] H. Olsson, Runge-Kutta solution of initial value problems, methods, algorithms and implementation, Ph.D. thesis, Lund University, Lund, Sweden (1998).

[13] L. R. Petzold, Order results for implicit Runge-Kutta methods applied to differential/algebraic systems, *SIAM J. Numer. Anal.*, **23**, (1986), 837-52.

[14] R. Piché and M. Palmroth, Modular modelling using lagrangian DAEs, Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Orlando, Florida, Volume 69-2, (2000), 755–761.

[15] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations* (Chapman & Hall, New York, 1994).

[16] J. H. Verner, High-order explicit Runge-Kutta pairs with low stage order, *Appl. Numer. Math.*, **22** (1996) 345-357.

Appendix

Proof of Lemma 5 For row s of a DIRK $\tilde{C}(2)$ implies

$$2\,A_{s\bullet}\,c^2 - c_s = 0$$

To have $\kappa_L = 5$ a DIRK must satisfy

$$\begin{aligned} 3\,b^T\,c^2 &= 1 \\ 4\,b^T\,c^3 &= 1 \end{aligned}$$

Using stiff accuracy and the given $A_{s\bullet}$, we can rewrite the last three equations:

$$\begin{aligned} (1-a_{ss})\,c_{s-1} + a_{ss} &= 1/2 \\ (1-a_{ss})\,c_{s-1}^2 + a_{ss} &= 1/3 \\ (1-a_{ss})\,c_{s-1}^3 + a_{ss} &= 1/4 \end{aligned}$$

Solving the first equation for c_{s-1} and substituting into the other two equations yields

$$(4\,a_{ss} - 1) = 0\,, \quad (6\,a_{ss}^2 - 6\,a_{ss} + 1) = 0$$

These two equations do not have a common solution for a_{ss}.

Proof of Lemma 6 A DIRK having $\check{C}(\ell)$ and the partition of (13) satisfies

$$\begin{bmatrix} W_1 & 0 \\ W_2 & W_3 \end{bmatrix} \begin{bmatrix} \gamma_1^\ell \\ \gamma_2^\ell \end{bmatrix} = \begin{bmatrix} W_1\,\gamma_1^\ell \\ \ell\,\gamma_2^{(\ell-1)} \end{bmatrix}$$

Multiplying both sides of this equation by A we get

$$\begin{bmatrix} \gamma_1^\ell \\ \gamma_2^\ell \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix} \begin{bmatrix} W_1\,\gamma_1^\ell \\ \ell\,\gamma_2^{(\ell-1)} \end{bmatrix}$$

Owing to stiff accuracy, the last row of this expression can be written as

$$c_s^\ell \;=\; \begin{bmatrix} 0 & \beta_2^T \end{bmatrix} \begin{bmatrix} W_1\,\gamma_1^2 \\ \ell\,\gamma_2^{(\ell-1)} \end{bmatrix} \;=\; \ell\,\beta_2^T\,\gamma_2^{(\ell-1)} \;=\; \ell\,b^T\,c^{(\ell-1)}$$

So the last row of A has $\tilde{C}(\ell)$.

Proof of Lemma 7 To attain $\tilde{C}(q)$ row i must satisfy

$$A_{i\bullet}\, c^{\ell-1} = c_i^{\ell}/\ell\,,\ \ell = 2,3,\ldots q \tag{21}$$

Parameterizing $A_{i\bullet}$ by

$$A_{i\bullet} = \left[c_i - \textstyle\sum_{k=2}^{i-1} a_{ik} - c_1 \ \ a_{i2}\ a_{i3}\ \ldots\ c_1\ 0\ \ldots\ 0\right]$$

we can rewrite (21) by

$$\sum_{k=2}^{i-1} a_{ik}\,(c_k^{(\ell-1)} - c_1^{(\ell-1)}) = c_1\,(c_1^{(\ell-1)} - c_i^{(\ell-1)}) + c_i^{\ell}/\ell - c_i\, c_1^{(\ell-1)}\,,\ \ell = 2,3,\ldots q \tag{22}$$

Assuming $q < i$, then from (22) we can set up a linear equation set to solve for $\left[a_{i2}\ a_{i3}\ \ldots\ a_{iq}\right]$. The $(q-1)\times(q-1)$ coefficient matrix for this linear equation set is

$$\begin{bmatrix} c_2 - c_1 & c_3 - c_1 & \ldots & c_q - c_1 \\ c_2^2 - c_1^2 & c_3^2 - c_1^2 & \ldots & c_q^2 - c_1^2 \\ \vdots & & & \vdots \\ c_2^{(q-1)} - c_1^{(q-1)} & c_3^{(q-1)} - c_1^{(q-1)} & \ldots & c_q^{(q-1)} - c_1^{(q-1)} \end{bmatrix} = V_{22} - V_{21}V_{12} \tag{23}$$

where $\begin{bmatrix} 1 & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$ is the transpose of the Vandermonde matrix for $[c_1,\ldots,c_q]$. The Vandermonde matrix (and hence the coefficient matrix (23)) is invertible iff the abscissae c_1, c_2, c_3, $\ldots$ c_q are distinct.

Proof of Lemma 8 The parameterization of $W_{i\bullet}$ we will use is

$$W_{i\bullet} = \left[c_1^{-2}\,(c_1 - c_1 \textstyle\sum_{k=2}^{i-1} w_{ik}c_k - c_i)\ w_{i2}\ w_{i3}\ \ldots\ c_1^{-1}\ 0\ldots\ 0\right] \tag{24}$$

Row i to have $\check{C}(q)$ when the following are satisfied:

$$W_{i\bullet}\, c^{\ell} = \ell\, c_i^{\ell-1}\,,\ \ell = 2,3,\ldots q \tag{25}$$

Using (24) we can rewrite (25) as

$$\sum_{k=2}^{i-1} w_{ik}\, c_k\,(c_{k-1}^{(\ell-1)} - c_1^{(\ell-1)}) = c_i^{(\ell-2)} - c_1^{(\ell-1)} + \ell\, c_i^{(\ell-1)} - c_i^{\ell}/c_1\,,\ \ell = 2,3,\ldots q \tag{26}$$

Assuming $q < i$, then from (22) we can set up a linear equation set to solve for $\begin{bmatrix} w_{i2} & w_{i3} & \dots & w_{iq} \end{bmatrix}$. The $(q-1) \times (q-1)$ matrix for this linear equation set can be written as

$$\begin{bmatrix} c_2 - c_1 & c_3 - c_1 & \dots & c_q - c_1 \\ c_2^2 - c_1^2 & c_3^2 - c_1^2 & \dots & c_q^2 - c_1^2 \\ \vdots & & & \vdots \\ c_2^{(q-1)} - c_1^{(q-1)} & c_3^{(q-1)} - c_1^{(q-1)} & \dots & c_q^{(q-1)} - c_1^{(q-1)} \end{bmatrix} \begin{bmatrix} c_2 & 0 & \dots & 0 \\ 0 & c_3 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & c_q \end{bmatrix} \tag{27}$$

We can use the same reasoning as in the proof of Lemma 7 to argue that non-confluency and $c_i \neq 0$ are sufficient for this matrix to be invertible.

Proof of Lemma 9 We will only consider the case of $q = s - r$.

Let $A_{s\bullet}$ be given by

$$A_{s\bullet} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 - \sum_{k=r+2}^{s-1} a_{sk} - a_{ss} & a_{s,r+2} & a_{s,r+3} & \dots & a_{ss} \end{bmatrix}$$

The conditions we need to satisfy are given by (21) with $i = s$ and $c_i = 1$. We can write these conditions as

$$a_{ss}\,(1 - c_{r+1}^{(\ell-1)}) + \sum_{k=r+2}^{s-1} a_{sk}\,(c_k^{(\ell-1)} - c_{r+1}^{(\ell-1)}) = 1/\ell - c_{r+1}^{(\ell-1)}\,,\ \ell = 2, 3, \dots q \tag{28}$$

If we set up the linear equation set to solve for $\begin{bmatrix} a_{s,r+2} & a_{s,r+3} & \dots & a_{ss} \end{bmatrix}$, then the corresponding $(s-r-1) \times (s-r-1)$ matrix is

$$\begin{bmatrix} c_{r+2} - c_{r+1} & c_{r+3} - c_{r+1} & \dots & 1 - c_{r+1} \\ c_{r+2}^2 - c_{r+1}^2 & c_{r+3}^2 - c_{r+1}^2 & \dots & 1 - c_{r+1}^2 \\ \vdots & & & \vdots \\ c_{r+2}^{(q-1)} - c_{r+1}^{(q-1)} & c_{r+3}^{(q-1)} - c_{r+1}^{(q-1)} & \dots & 1 - c_{r+1}^{(q-1)} \end{bmatrix}$$

We can use the same reasoning as in the proof of Lemma 7 to argue that non-confluency is sufficient for this matrix to be invertible.

Proof of Lemma 10 The proof of lemma 9 uses the framework of the proof of lemma 7 with the following condition: we may solve only for the last $s-r-1$ elements of $A_{s\bullet}$, which includes the diagonal element. In the same sense we can prove lemma 10 using the framework of the proof of lemma 8.

Proof of Theorem 11 Consider a stiffly accurate SDIRK designed with the following properties: (i) rows $q+1$, $q+2$, $\ldots$ $s-1$ have $\tilde{C}(q)$, (ii) row s has $\tilde{C}(q)$, and (iii) the first q elements of $A_{s\bullet}$ are zero. Property (i) is possible from lemma 7. Property (iii) and the partition of (13) implies $r = q$. From $r = q$ and lemma 9, property (ii) is possible when $s \geq q + r = 2q$. From defn. 2 this SDIRK has $C(q)$.

Proof of Theorem 12 Analogous to the proof of Theorem 11.