STABILITY INVESTIGATIONS FOR INDEX-2 SYSTEMS

J. WENSCH, R. WEINER, K. STREHMEL

ABSTRACT. This paper deals with differential-algebraic equations of index two in Hessenberg form. It is an attempt to extend the concept of A-stability to those DAE's. For linear DAE's contractivity in the analytical solution is investigated. The results are compared with contractivity in numerical solutions obtained by implicit Runge-Kutta methods, projected implicit Runge-Kutta methods and BDF methods. Test equations are used to develop stability functions for these methods. The corresponding stability regions are displayed for a choice from these methods. Numerical tests verify the results for model problems as well as for non-model problems.

1. INTRODUCTION

The numerical treatment of differential algebraic equations (DAE's) became a more and more important field in numerical mathematics in recent years. Almost all attempts that have been made to develop methods for the solution of differential algebraic equations are based on the modification of methods for ODE's.

The principal behavior of the analytical resp. numerical solution with respect to perturbations of the system is mainly determined by the index of the system. DAE's of index one behave almost like ODE's, an application of methods for ODE's to these DAE's is possible without trouble. In the case of DAE's of index two the situation is more complicated. Further, we do not know any method that works well in general for DAE's of index three or higher without an index reduction of the system. Therefore DAE's of index two play an important role in the theory of numerical treatment of DAE's.

The most common methods for ODE's such as Runge-Kutta methods or BDF's may be applied to DAE's in a natural way. These methods are subject of our investigations. In the ODE-case the theory of such methods has been developed to a wide field of criterions for the method to choose. The most important of these criterions are order and stability properties of the method.

Our aim is to extend the concept of A-stability to DAE's of index two. In the ODE-case the simplest linear differential equation is used as a test equation. We will go a similar way. We consider a linear DAE and split it into two decoupled systems by a linear transformation, where the first system is an ODE and the second system is an algebraic system that has a unique solution. We will characterize the

1991 *Mathematics Subject Classification.* 65L06.

Key words and phrases. Stability, stability function, Differential-algebraic systems, projected implicit Runge-Kutta methods, BDF-methods.

stability properties of various RK-methods and BDF-methods with respect to that ODE.

2. The Essential Underlying ODE

Our subject are DAE's of index 2 in Hessenberg form. First of all we will restrict to linear, constant coefficient DAE's. We have

$$\begin{aligned} y' &= By + Cz + q(t) \\ 0 &= Dy + r(t) \end{aligned} \tag{1}$$

with the unknowns $y \in \mathbb{R}^{n_y}$ and $z \in \mathbb{R}^{n_z}$ and constant matrices $B \in \mathbb{R}^{n_y \times n_y}, C \in \mathbb{R}^{n_y \times n_z}, D \in \mathbb{R}^{n_z \times n_y}$. The system is of index 2 iff DC is regular. Further $n_y > n_z$ is assumed. Otherwise the differential variables are determined by the algebraic equations.

Following [1], [2] we construct a regular linear transformation of the variables to obtain a system in a form that is similar to the Weierstrass-Kronecker Canonical Form (see [3], [4]). We choose a full rank matrix $R \in \mathbb{R}^{(n_y - n_z) \times n_y}$ with $RC = 0$, i.e., the linear independent rows of R span the nullspace of C^T. The matrix

$$\begin{pmatrix} R \\ D \end{pmatrix}$$

is regular because the rows of D are linearly independent and

$$\operatorname{im} R^T \cap \operatorname{im} D^T = \ker C^T \cap \operatorname{im} D^T = \{0\} \quad (C^T D^T \text{is regular}).$$

We write its inverse in the form $(S\, F)$ with $S \in \mathbb{R}^{n_y \times (n_y - n_z)}, F \in \mathbb{R}^{n_y \times n_z}$. We have

$$RS = I, RF = 0, DS = 0, DF = I, SR + FD = I, F = C(DC)^{-1}. \tag{2}$$

The transformation

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} S & 0 & F \\ -(DC)^{-1}DBS & (DC)^{-1} & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} =: T \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{3}$$

gives

$$\begin{pmatrix} S & 0 & F \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix} = \begin{pmatrix} SRBS & C(DC)^{-1} & BF \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} q(t) \\ r(t) \end{pmatrix}.$$

Multiplying with

$$\begin{pmatrix} R & -RBF \\ D & 0 \\ 0 & I \end{pmatrix} \tag{4}$$

yields

$$\begin{pmatrix} \boxed{I} & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix} = \begin{pmatrix} \boxed{RBS} & 0 & 0 \\ 0 & I & DBF \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} Rq(t) - RBFr(t) \\ Dq(t) \\ r(t) \end{pmatrix} \tag{5}$$

The upper block of this system gives an ODE for the components u_1. Therefore we call them the differential part of the DAE. The lower part of the lower block gives u_3 as the solution of algebraic equations. Substituting this solution in the upper part of the lower block gives u_2 as the solution of algebraic equations, too. Therefore these components are called the algebraic part of the DAE. The ODE for the differential part

$$u_1' = (RBS)u_1 + Rq(t) - RBFr(t) \tag{6}$$

is called the *essential underlying ODE* (EUODE) [2].

These terms describe the solution behavior quite well. The differential part is given as solution of an ODE. We can prescribe arbitrary initial values for the differential part and will get a unique solution for every set of initial values if the righthand-side is continous. On the other side the algebraic part is determined by algebraic relations. We need no solution history, no initial values. The algebraic relations can be solved at every point in the integration interval and will give a unique solution for the algebraic part (whereby we have to assume the righthand-sides of the equations to be sufficiently smooth).

3. The Integration of the EUODE

We want to investigate the relation between contractivity in the analytical solution and contractivity in a numerical solution of the DAE obtained by a numerical one-step method Φ. This method gives values y_{n+1}, z_{n+1} at t_{n+1} depending on y_n, z_n at t_n and the stepsize h_n:

$$\begin{pmatrix} y_{n+1} \\ z_{n+1} \end{pmatrix} = \Phi(t_n, h_n) \begin{pmatrix} y_n \\ z_n \end{pmatrix}.$$

As explained in the section above, the term contractivity makes sense only for the differential part, because contractivity describes the behavior of the difference of two solutions of the system. There is a unique analytical solution for the components of the algebraic part and therefore no contractivity. We will restrict our investigations to the question what happens with the differential part in the process of integration of the DAE. That means, we consider instead of Φ a method Φ^* that works in the transformed space given by (3). This gives

$$\begin{pmatrix} u_{1,n+1} \\ u_{2,n+1} \\ u_{3,n+1} \end{pmatrix} = \Phi^*(t_n, h_n) \begin{pmatrix} u_{1,n} \\ u_{2,n} \\ u_{3,n} \end{pmatrix} = T^{-1} \circ \Phi(t_n, h_n) \circ T \begin{pmatrix} u_{1,n} \\ u_{2,n} \\ u_{3,n} \end{pmatrix}.$$

For a general one-step method the value $u_{1,n+1}$ may depend on $u_{1,n}, u_{2,n}, u_{3,n}$ and t_n, h_n, of course. It would be more pleasant if $u_{1,n+1}$ would be independent of the algebraic part $u_{2,n}, u_{3,n}$. In this case we have

$$u_{1,n+1} = \Phi^{**}(t_n, h_n)(u_{1,n})$$

with a welldefined numerical method Φ^{**} that depends on the original method Φ. Therefore the method Φ^{**} gives values in the differential part, depending only on initial values for the differential part. We can speak of the "process of integration of the EUODE". This separate integration process may be understood as a one-step method and will be the subject of the following investigations.

We consider the class of implicit Runge-Kutta methods with invertible coefficient matrix. The application to the linear, constant coefficient DAE (1) gives (see [6])

$$\begin{aligned}
Y_{ni} &= y_n + h \sum_{j=1}^{s} a_{ij} (BY_{nj} + CZ_{nj} + q(t_n + c_j h)) \\
Z_{ni} &= z_n + h \sum_{j=1}^{s} a_{ij} Z'_{nj} \\
0 &= DY_{ni} + r(t_n + c_i h) \\
y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i (BY_{ni} + CZ_{ni} + q(t_n + c_i h)) \\
z_{n+1} &= z_n + h \sum_{i=1}^{s} b_i Z'_{ni}
\end{aligned}$$

From the transformation (3) we obtain

$$y_n = Su_{1,n} + Fu_{3,n}, \quad y_{n+1} = Su_{1,n+1} + Fu_{3,n+1}$$

and by multiplication with R

$$u_{1,n} = Ry_n, \quad u_{1,n+1} = Ry_{n+1}.$$

We define internal values for the components u_1 by

$$U_{ni} = RY_{ni} \qquad (\Rightarrow Y_{ni} = (SR + FD)Y_{ni} = SU_{ni} - Fr(t_n + c_i h)).$$

We get:

$$
\begin{aligned}
U_{ni} &= u_{1,n} + h\sum_{j=1}^{s} a_{ij}(RBY_{nj} + Rq(t_n + c_j h)) \\
U_{ni} &= u_{1,n} + h\sum_{j=1}^{s} a_{ij}(RBSU_{nj} + (Rq(t_n + c_j h) - RBFr(t_n + c_j h))) \\
u_{1,n+1} &= Ry_{n+1} \\
&= u_n + h\sum_{i=1}^{s} b_i(RBSU_{ni} + (Rq(t_n + c_i h) - RBFr(t_n + c_i h)))
\end{aligned}
\tag{7}
$$

Obviously, the values $u_{1,n+1}$ do not depend on the initial values in the algebraic part. As described above, the essential underlying ODE is integrated separately. A closer look shows that (7) coincides with a Runge-Kutta method for ODE's. We have:

Theorem 3.1. *If a constant coefficient DAE is solved numerically by an implicit Runge-Kutta method, then the essential underlying ODE is integrated by the same Runge-Kutta formula.*

Remark 3.1. *The same is valid for Projected Runge-Kutta methods and the BDF methods. The proof is similar.*

We can extend the concept of A-stability to DAE's in a natural way. In the ODE-case a scalar, linear, constant coefficient ODE is used as a test equation. A method is called A-stable if dissipativity of the ODE implies contractivity of two numerical solutions. It is recommended to use A-stable and L-stable methods if the ODE is stiff.

In the DAE-case we use a linear, constant coefficient DAE with a scalar EUODE as test equation. An A-stable method gives contractive numerical solutions of the EUODE if the EUODE is dissipative. Hence, if the EUODE is stiff, it is recommended to use A-stable and L-stable Runge-Kutta methods to integrate the DAE.

Of course, this is not really restrictive because the most common implicit Runge-Kutta methods with invertible coefficient matrix are A-stable, such as the Gauss, Radau and LobattoIIIC-methods (see [?]).

We have to mention, that the essential underlying ODE is not unique, it could be changed by a regular linear transformation of the differential part. But such properties like stiffness can not be changed by a coordinate transformation. The existence of strongly damped solution components is invariant with respect to any regular linear transformation.

4. A More Complicated Model

Our main idea is to extend the stability concept of A-stability to DAE's of index 2 by considering the process of integration of the essential underlying ODE. In the section above we assumed the DAE to have constant coefficients. A more general approach would be to assume only the EUODE to have constant coefficients.

First of all we have to define the EUODE for linear DAE's with timedepending coefficients. In the constant coefficient case we transformed the system into a form similar to Weierstrass-Kronecker Canonical Form and the differential equation for the differential part was defined as the EUODE. For timedepending DAE's a similar way is possible.

We consider a linear DAE of index 2 in Hessenberg form

$$\begin{aligned} y' &= B(t)y + C(t)z + q(t) \\ 0 &= D(t)y + r(t) \end{aligned} \tag{8}$$

with the unknowns $y \in \mathbb{R}^{n_y}$ and $z \in \mathbb{R}^{n_z}$.

The matrices $B(t) \in \mathbb{R}^{n_y \times n_y}, C(t) \in \mathbb{R}^{n_y \times n_z}, D(t) \in \mathbb{R}^{n_z \times n_y}$ are supposed to be sufficiently smooth. The system is of index 2 iff $D(t)C(t)$ is regular in the integration interval $[t_0, t_E]$.

We follow again the idea in [2]. Let the rows of the smooth matrix $R(t) \in \mathbb{R}^{(n_y - n_z) \times n_y}$ span the left nullspace of $C(t)$, i.e.,

$$R(t)C(t) = 0. \tag{9}$$

Analogous to section 2 we define matrices $S(t), F(t)$ by (2), now timedepending and supposed to be sufficiently smooth. We introduce

$$\begin{aligned} u_1 &= Ry \\ u_2 &= (D' + DB)SRy + DCz \\ u_3 &= Dy. \end{aligned}$$

This transformation is regular because we have the inverse transformation given by

$$\begin{aligned} y &= Su_1 + Fu_3 \\ z &= -(DC)^{-1}(D' + DB)Su_1 + (DC)^{-1}u_2. \end{aligned}$$

For u_1, u_2, u_3 we get the system

$$\begin{pmatrix} I & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix} = \begin{pmatrix} (R' + RB)S & 0 & 0 \\ 0 & I & (D' + DB)F \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} Rq - (R' + RB)Fr \\ Dq \\ r \end{pmatrix}$$

We have the components $u_1 = R(t)y$ as the differential part of the DAE. In the following we will use $v = u_1$ to avoid too many subscripts. We get the EUODE as

$$v' = (RBS + R'S)v + Rq - (R' + RB)Fr \tag{10}$$

We mention, that the EUODE depends on the choice of $R(t)$. The matrix $R(t)$ should be choosen in such a way, that the transfomation does neither change stiff components to nonstiff nor reversely. We will denote this with the term moderately timevarying. If it is not possible to choose such an R, then there is no relation between the EUODE and the original DAE with respect to stiffness, error amplification and stable resp. instable integration. We restrict our investigations to those DAEs with a moderately timevarying R. Nevertheless this moderately timevarying R is in general not uniquely determined. But two different moderately timevarying matrices R_1 and R_2 (and the corresponding components of the two different EU-ODEs too) are related by a moderately timevarying regular transformation, that does not change such properties as stiffness and stable resp. unstable integration.

We will illustrate the difficulties in the integration of the EUODE in the case of timevarying coefficients by an example. We consider a homogenous DAE of form (8) under the assumptions

$$\begin{aligned} R(t)B(t)S(t) =& \text{const.} \\ R'(t)S(t) =& \text{const.} \\ R'(t) =& \text{const.} \end{aligned} \tag{11}$$

and apply the implicit Euler method. We get

$$\begin{aligned} y_{n+1} =& y_n + h(B(t_{n+1})y_{n+1} + C(t_{n+1})z_{n+1}) \\ 0 =& D(t_{n+1})y_{n+1} \end{aligned}$$

and for the differential part $v = Ry$

$$\begin{aligned} v_{n+1} =& R(t_{n+1})y_{n+1} \\ =& (R(t_n) + h_n R')y_n + hR(t_{n+1})B(t_{n+1})y_{n+1} \\ =& v_n + hR'(Sv_n + F(t_n)D(t_n)y_n) + hRBSv_{n+1} \\ v_{n+1} =& (I - hRBS)^{-1}(I + hR'S)v_n, \end{aligned}$$

where we used $D(t_n)y_n = 0$.

The term RBS of the EUODE is treated by the implicit Euler method, but the term $R'S$ is treated by the explicit Euler method (see [2]). In difference to the case of a constant coefficient DAE we would not get contractive numerical solutions for all stepsizes h, although the used method is A-stable. This appears for instance in the case of a scalar EUODE with $RBS = 0$ and $R'S < 0$. It seems that the simple model of a constant coefficient DAE does not govern all the stability problems that arise in the process of integration of a DAE. It looks natural to extend the investigations to the case where the coefficients of the DAE are timedependent, but the coefficients of the EUODE are constant. We will seek for methods that integrate the term $R'S$ in an implicite way, too.

To express v_{n+1} independent of Dy_n and z_n, we needed the assumption, that $D(t_n)y_n = 0$, i.e., we assumed the initial values in y to be consistent. Therefore we will restrict on one-step methods that give y-values on the constraints, whereby we assume exact arithmetic and the exact solution of the equations. The effects that occour if we solve the equations by Newtons method or any other numerical procedure will not be discussed.

This leads to the class of projected Runge-Kutta methods.

5. Projected Runge-Kutta Methods

Projected Runge-Kutta methods are designed especially for semi-explicit DAE's in Hessenberg form. They have been introduced and investigated by Ascher/Petzold

[1],[2] and Lubich [7]. A projected Runge-Kutta method, applied to a DAE of index 2 in Hessenberg form

$$\begin{aligned} y' &= f(t, y, z) \\ 0 &= g(t, y) \end{aligned} \tag{12}$$

reads as follows:

$$\begin{aligned}
Y_{ni} &= y_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{nj}, Z_{nj}) \quad i = 1, \ldots, s \\
Z_{ni} &= z_n + h \sum_{j=1}^{s} a_{ij} Z'_{nj} \\
0 &= g(t + c_i h, Y_{ni}) \\
\tilde{y}_{n+1} &= y_n + h \sum_{i=1}^{s} b_i f(t_n + c_i h, Y_{ni}, Z_{ni}) \\
z_{n+1} &= z_n + h \sum_{i=1}^{s} b_i Z'_{ni} \\
y_{n+1} &= \tilde{y}_{n+1} + f_z(t_{n+1}, y_{n+1}, z_{n+1}) \lambda \\
0 &= g(t_{n+1}, y_{n+1}).
\end{aligned}$$

We assume the coefficient matrix of the underlying Runge-Kutta formula to be invertible. The DAE is solved by the underlying Runge-Kutta formula and the y-components of the solution are projected onto the constraints. Therefore the method guarantees that the y-components are consistent. We mention that for stiffly accurate Runge-Kutta methods, i.e., methods with $a_{si} = b_i$, the projected method and the underlying method coincide because $\tilde{y}_{n+1} = Y_{ns}$ satisfies the constraints.

Our main goal is to investigate contractivity in the differential part of the numerical solution of linear DAE's. As easily can be seen, the difference of two numerical solutions of a linear DAE for different initial values (where the y values are assumed to be consistent) is the result given by an application of the projected method to the homogenous system, where initial values are given by the difference of the original initial values. Therefore we will integrate the homogeneous system by a projected Runge-Kutta method.

We assume the transformation matrices R, S defined as above to satisfy (11). To have a constant coefficient EUODE it would be sufficient to have $RBS + R'S =$ const. But the example above made clear, that the expressions $R'S$ and RBS will be integrated in a different way. Therefore we assume both to be constant. The reason of the third condition of (11) is not obvious. We want to describe the procedure that integrates the EUODE. This procedure should depend on the coefficients of the underlying Runge-Kutta method and on the coefficients RBS and $R'S$ of the essential underlying ODE. It should not depend on the coefficients of the original DAE. We will see, that this will be guaranteed by the third assumption of (11). Further we assume the EUODE to be scalar. Under these assumptions the EUODE is uniquely determined.

We apply one step of the projected Runge-Kutta method with stepsize h to the homogenous system. The coefficients of the underlying Runge-Kutta formula are denoted by

$$\boldsymbol{A} = (a_{ij}), \boldsymbol{A}^{-1} = (\omega_{ij}), \boldsymbol{C} = \operatorname{diag}(c_1, \dots, c_s), \boldsymbol{b}^T = (b_1, \dots, b_s).$$

We assume the simplifying assumption C(1) to be satisfied, i.e., $c_i = \sum a_{ij}$.

The projected Runge-Kutta method applied to this special DAE reads as follows:

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij}(B_j Y_j + C_j Z_j) \tag{13}$$

$$0 = D_i Y_i \tag{14}$$

$$z_{n+1} = z_n + \sum_{i,j=1}^{s,s} b_i w_{ij}(Z_j - z_n) \tag{15}$$

$$\tilde{y}_{n+1} = y_n + \sum_{i,j=1}^{s,s} b_i w_{ij}(Y_j - y_n) \tag{16}$$

$$y_{n+1} = \tilde{y}_{n+1} + C(t_{n+1})\lambda \tag{17}$$

$$0 = D(t_{n+1}) y_{n+1} \tag{18}$$

With $V_i := R_i Y_i$, $v_n := R(t_n) y_n$ and $v_{n+1} := R(t_{n+1}) y_{n+1}$ we get

$$\begin{aligned} Y_i &= S_i R_i Y_i + F_i D_i Y_i \\ &= S_i V_i \end{aligned}$$

and

$$y_n = S(t_n) v_n$$

A linear transformation of (13) gives

$$\begin{aligned} \sum_j w_{ij}(Y_j - y_n) &= h(B_i Y_i + C_i Z_i) \\ \sum_j w_{ij} R_i (Y_j - y_n) &= h R_i B_i Y_i \\ \sum_j w_{ij}(R_j + h(c_i - c_j)R')(S_j V_j - S(t_n) v_n) &= hRBSV_i. \end{aligned}$$

With $z_1 := hR'S$ and $z_2 := hRBS$ follows

$$\begin{aligned} \textstyle\sum_j w_{ij}(1 + (c_i - c_j) z_1)(V_j - v_n) &= z_2 V_i + \textstyle\sum_j w_{ij} c_j z_1 v_n \\ \textstyle\sum_j w_{ij}(1 + (c_i - c_j) z_1)(V_j - v_n) &= z_2 (V_i - v_n) + (z_1 + z_2) v_n. \end{aligned}$$

In matrix notation we get

$$\begin{aligned} \boldsymbol{A}^{-1}(I + (\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^{-1} - \boldsymbol{C}) z_1)(V - \mathbb{1} \otimes v_n) &= z_2 (V - \mathbb{1} \otimes v_n) \\ &\quad + (z_1 + z_2)(\mathbb{1} \otimes v_n) \\ \boldsymbol{A}^{-1}(I + (\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^{-1} - \boldsymbol{C}) z_1 - \boldsymbol{A} z_2)(V - \mathbb{1} \otimes v_n) &= (z_1 + z_2) \mathbb{1} \otimes v_n \end{aligned}$$

with $V = (V_1^T, \ldots, V_s^T)^T$. For the values at t_{n+1} we get

$$\begin{aligned}
v_{n+1} &= R(t_{n+1})y_{n+1} \\
&= R(t_{n+1})\tilde{y}_{n+1} \\
&= R(t_{n+1})(S(t_n)v_n + \sum_{i,j} b_i w_{ij}(Y_j - y_n)) \\
&= (R(t_n) + hR')S(t_n)v_n \\
&\quad + \sum_{i,j} b_i w_{ij}(R_j + h(1-c_j)R')(S_j V_j - (R(t_n) + hR')(S(t_n)v_n)) \\
&= (1+z_1)v_n + \sum_{i,j} b_i w_{ij}((1+(1-c_j)z_1)V_j - (1+z_1)v_n) \\
&= (1+z_1)v_n + \sum_{i,j} b_i w_{ij}(1+(1-c_j)z_1)(V_j - v_n) - z_1 \sum_{i,j} b_i w_{ij} c_j v_n \\
&= (1+(1-\sum_i b_i)z1)v_n + \sum_{i,j} b_i w_{ij}(1+(1-c_j)z_1)(V_j - v_n).
\end{aligned}$$

With $\sum_i b_i = 1$ follows

$$\begin{aligned}
v_{n+1} &= v_n + \boldsymbol{b}^T \boldsymbol{A}^{-1}(I + (I - \boldsymbol{C})z_1)(V - \mathbb{1} \otimes v_n) \\
&= (1 + (z_1+z_2)\boldsymbol{b}^T \boldsymbol{A}^{-1}(I + (I - \boldsymbol{C})z_1)(I + (\boldsymbol{ACA}^{-1} - \boldsymbol{C})z_1 - \boldsymbol{A}z_2)^{-1}\boldsymbol{A}\mathbb{1})v_n \\
&= (1 + (z_1+z_2)\boldsymbol{b}^T (I + (I - \boldsymbol{A}^{-1}\boldsymbol{CA})z_1)(I + (\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{CA})z_1 - \boldsymbol{A}z_2)^{-1}\mathbb{1})v_n.
\end{aligned}$$

This gives:

Theorem 5.1. *Let the linear DAE of index 2 in Hessenberg form*

$$\begin{aligned}
y' &= B(t)y + C(t)z + q(t) \\
0 &= D(t)y + r(t)
\end{aligned}$$

with scalar EUODE be integrated by a projected Runge-Kutta method based on a method with invertible coefficient matrix. We assume that there exist sufficiently smooth matrices R, S satisfying (2) and (11). Let

$$y_{1,i}, y_{2,i} \qquad (i = 1, \ldots, N)$$

be two numerical solutions, obtained for different initial values $y_{1,0}, y_{2,0}$. The EU-ODE is assumed to be scalar.

Then we have:

(1) *The differential part of the DAE is given by $v(t) = R(t)y(t)$.*
(2) *The differential part is the solution of the essential underlying ODE (10).*
(3) *The difference $\Delta v_n = R(t_n)y_{2,n} - R(t_n)y_{1,n}$ of the corresponding differential parts satisfies a relation*

$$\Delta v_{n+1} = \mathcal{R}(h_n R'S, h_n RBS)\Delta v_n$$

with a rational function $\mathcal{R}$.

(4) *This rational function is given by*

$$\begin{aligned}\mathcal{R}(z_1, z_2) := 1 + (z_1 + z_2) b^T (I + (I - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) z_1) \\ \cdot (I + (\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) z_1 - \boldsymbol{A} z_2)^{-1} \mathbb{1}\end{aligned} \tag{19}$$

6. The Extended Stability Function

Analogous to the concept of A-stability for ODE's we define

Definition 6.1. *The rational function (19) is called the extended stability function. The set*

$$S := \{(z_1, z_2) \in \mathbb{C}^2 : |\mathcal{R}(z_1, z_2)| \le 1\}$$

is called the extended stability domain.

For the numerical solution by projected implicit Runge-Kutta methods we get the following results:

Theorem 6.1. (1) *For stiffly accurate methods the extended stability function simplifies to*

$$\mathcal{R}(z_1, z_2) := 1 + (z_1 + z_2) b^T (I + (\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) z_1 - \boldsymbol{A} z_2)^{-1} \mathbb{1}.$$

(2) *In the case of $z_1 = 0$ the extended stability function simplifies to the stability function for ordinary differential equations.*

(3) *For DIRK methods the extended stability function is polynomial in z_1 because the matrix*

$$I + (\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) z_1 - \boldsymbol{A} z_2$$

is lower triangular but z_1 does not occur in the main diagonal.

(4) *If the extended stability function is not polynomial in z_1, i.e., it is a rational function with a nonconstant denominator with respect to z_1, then the extended stability function has singularties in the left z_1 halfplane. That means, that the denominator of $\mathcal{R}(z, 0)$ has zeros ζ with $\Re\zeta \le 0$.*

(5) *In the Lobatto IIIC case $\mathcal{R}(\infty, \infty) = 0$ holds. Strictly speaking, for almost all $\lambda \in \boldsymbol{C}$ holds*

$$\lim_{x \to \infty} \mathcal{R}(x, \lambda x) = 0.$$

We will give a short proof of the last two conclusions:

(4): The denominator of $\mathcal{R}(z, 0)$ is

$$\det(I + (\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) z).$$

We have

$$\mathrm{tr}(\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) = \mathrm{tr}\boldsymbol{C} - \mathrm{tr}(\boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A}) = 0.$$

Therefore $(\boldsymbol{C} - \boldsymbol{A}^{-1}\boldsymbol{C}\boldsymbol{A})$ has at least one nonzero eigenvalue λ with nonnegative real part, and we have a zero of the denominator of $\mathcal{R}(z, 0)$ at $-1/\lambda$.

(5): We have

$$\begin{aligned}
\mathcal{R}(x,\lambda x) &= 1+(1+\lambda)xb^T(I+(C-A^{-1}CA)x-\lambda Ax)^{-1}\mathbb{1} \\
\lim_{x\to\infty}\mathcal{R}(x,\lambda x) &= 1+(1+\lambda)b^T(C-A^{-1}CA-\lambda A)^{-1}\mathbb{1} \\
(C-A^{-1}CA-\lambda A)e_1 &= -b_1A^{-1}C\mathbb{1}-\lambda b_1\mathbb{1} \\
&= -b_1(1+\lambda)\mathbb{1} \\
\lim_{x\to\infty}\mathcal{R}(x,\lambda x) &= 1+(1+\lambda)b^T(-b_1(1+\lambda))^{-1}e_1 \\
&= 0
\end{aligned}$$

For that conclusion a necessary assumption is that the matrix

$$(C-A^{-1}CA-\lambda A)$$

is regular. Therefore we have to exclude a finite set of λ-values.

The following figures give an overview about the extended stability functions and stability regions for some classes of implicit Runge-Kutta methods. The methods named 'AV' are collocation methods that are stable on the real z_1 axis, i. e., $|R(z,0)| \leq 1$ for $z \leq 0$. The two-stage method is of order two and the three-stage method is of order four with respect to the y-component. They are defined in [8]. The SDIRK methods are of order 2 for DAE's of index 2. We will not display the coefficients of this methods or their extended stability functions. Their stability regions are displayed to illustrate part (3) of theorem 6.1.

The extended stability function has been developed by using a scalar EUODE as test equation. To extend this analysis on systems, one has to diagonalize the matrizes $R'S$ and RBS, but both by the same transformation ! In general this is impossible. Even the case of a scalar EUODE shows difficulties to gain general results. However, it allows to classify the Runge-Kutta methods with respect to their stability properties for DAE's of Index two.

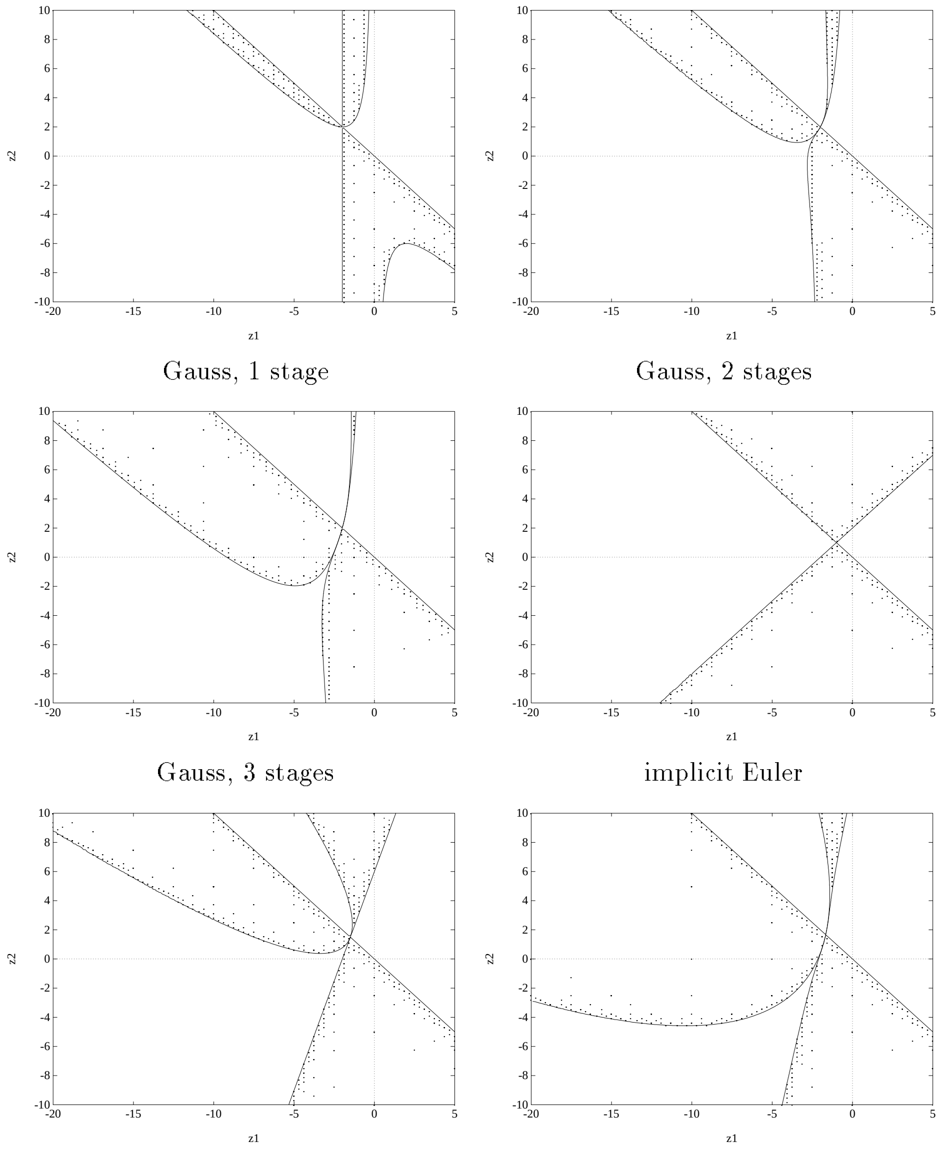
It shows too, that there is a difference in stability properties of Runge-Kutta methods by applying them to ODE's resp. DAE's.

To display the stability regions we used the projection $\Im z_1 = \Im z_2 = 0$. The analytical solution is stable for $z_1 + z_2 \leq 0$. There seems to be no method that is stable for all $z_1 + z_2 \leq 0$. Although some methods are stable on the real z_1 and z_2 axes, there is always a 'gap' between these axes. An analysis for Collocation methods shows, that there are poles of the extended stability function inside this hole.

The best stability properties are obtained by stiffly accurate methods as the Radau IIA, Lobatto IIIC and AV-methods. As expected by theorem 6.1, part (3), the extended stability region is bounded on the real z_1-axis for the DIRK methods.

Table 1. Extended Stability Functions

type	stages	$\mathcal{R}(x,y)$	$\mathcal{R}(0,y)$	$\mathcal{R}(x,0)$
$Gauß$	1	$\frac{1+x+\frac{x^2}{2}+\frac{y}{2}+\frac{x\,y}{2}}{1-\frac{y}{2}}$	$\frac{1+\frac{y}{2}}{1-\frac{y}{2}}$	$1+x+\frac{x^2}{2}$
$Gauß$	2	$\frac{1+x+\frac{5\,x^2}{12}+\frac{x^3}{12}+\frac{y}{2}+\frac{x\,y}{2}+\frac{x^2\,y}{6}+\frac{y^2}{12}+\frac{x\,y^2}{12}}{1-\frac{x^2}{12}-\frac{y}{2}+\frac{y^2}{12}}$	$\frac{1+\frac{y}{2}+\frac{y^2}{12}}{1-\frac{y}{2}+\frac{y^2}{12}}$	$\frac{1+x+\frac{5\,x^2}{12}+\frac{x^3}{12}}{1-\frac{x^2}{12}}$
$Gauß$	3	$\frac{1+x+\frac{2\,x^2}{5}+\frac{x^3}{12}+\frac{x^4}{120}+\frac{y}{2}+\frac{x\,y}{2}+\frac{7\,x^2\,y}{40}+\frac{x^3\,y}{40}+\frac{y^2}{10}+\frac{x\,y^2}{10}+\frac{x^2\,y^2}{40}+\frac{y^3}{120}+\frac{x\,y^3}{120}}{1-\frac{x^2}{10}+\frac{x^3}{60}-\frac{y}{2}+\frac{x^2\,y}{40}+\frac{y^2}{10}-\frac{y^3}{120}}$	$\frac{1+\frac{y}{2}+\frac{y^2}{10}+\frac{y^3}{120}}{1-\frac{y}{2}+\frac{y^2}{10}-\frac{y^3}{120}}$	$\frac{1+x+\frac{2\,x^2}{5}+\frac{x^3}{12}+\frac{x^4}{120}}{1-\frac{x^2}{10}+\frac{x^3}{60}}$
$RadauIIA$	1	$\frac{1+x}{1-y}$	$\frac{1}{1-y}$	$1+x$
$RadauIA, IIA$	2	$\frac{1+x+\frac{x^2}{3}+\frac{y}{3}+\frac{x\,y}{3}}{1-\frac{x^2}{6}-\frac{2\,y}{3}+\frac{y^2}{6}}$	$\frac{1+\frac{y}{3}}{1-\frac{2\,y}{3}+\frac{y^2}{6}}$	$\frac{1+x+\frac{x^2}{3}}{1-\frac{x^2}{6}}$
$RadauIA, IIA$	3	$\frac{1+x+\frac{7\,x^2}{20}+\frac{x^3}{20}+\frac{2\,y}{5}+\frac{2\,x\,y}{5}+\frac{x^2\,y}{10}+\frac{y^2}{20}+\frac{x\,y^2}{20}}{1-\frac{3\,x^2}{20}+\frac{x^3}{30}-\frac{3\,y}{5}+\frac{x^2\,y}{20}+\frac{3\,y^2}{20}-\frac{y^3}{60}}$	$\frac{1+\frac{2\,y}{5}+\frac{y^2}{20}}{1-\frac{3\,y}{5}+\frac{3\,y^2}{20}-\frac{y^3}{60}}$	$\frac{1+x+\frac{7\,x^2}{20}+\frac{x^3}{20}}{1-\frac{3\,x^2}{20}+\frac{x^3}{30}}$
$LobattoIIIC$	2	$\frac{1+x}{1-\frac{x^2}{2}-y+\frac{y^2}{2}}$	$\frac{1}{1-y+\frac{y^2}{2}}$	$\frac{1+x}{1-\frac{x^2}{2}}$
$LobattoIIIC$	3	$\frac{1+x+\frac{x^2}{4}+\frac{y}{4}+\frac{x\,y}{4}}{1-\frac{x^2}{4}+\frac{x^3}{12}-\frac{3\,y}{4}+\frac{x^2\,y}{8}+\frac{y^2}{4}-\frac{y^3}{24}}$	$\frac{1+\frac{y}{4}}{1-\frac{3\,y}{4}+\frac{y^2}{4}-\frac{y^3}{24}}$	$\frac{1+x+\frac{x^2}{4}}{1-\frac{x^2}{4}+\frac{x^3}{12}}$
AV	2	$\frac{1+x+\frac{x^2}{4}+\frac{y}{4}+\frac{x\,y}{4}}{1-\frac{x^2}{4}-\frac{3\,y}{4}+\frac{y^2}{4}}$	$\frac{1+\frac{y}{4}}{1-\frac{3\,y}{4}+\frac{y^2}{4}}$	$\frac{1+x+\frac{x^2}{4}}{1-\frac{x^2}{4}}$
AV	3	$\frac{1+x+\frac{x^2}{3}+\frac{x^3}{24}+\frac{3\,y}{8}+\frac{3\,x\,y}{8}+\frac{x^2\,y}{12}+\frac{y^2}{24}+\frac{x\,y^2}{24}}{1-\frac{x^2}{6}+\frac{x^3}{24}-\frac{5\,y}{8}+\frac{x^2\,y}{16}+\frac{y^2}{6}-\frac{y^3}{48}}$	$\frac{1+\frac{3\,y}{8}+\frac{y^2}{24}}{1-\frac{5\,y}{8}+\frac{y^2}{6}-\frac{y^3}{48}}$	$\frac{1+x+\frac{x^2}{3}+\frac{x^3}{24}}{1-\frac{x^2}{6}+\frac{x^3}{24}}$



Gauss, 1 stage Gauss, 2 stages

Gauss, 3 stages implicit Euler

RadauIA,IIA, 2 stages RadauIA,IIA, 3 stages

Figure 1. Extended Stability Domains I

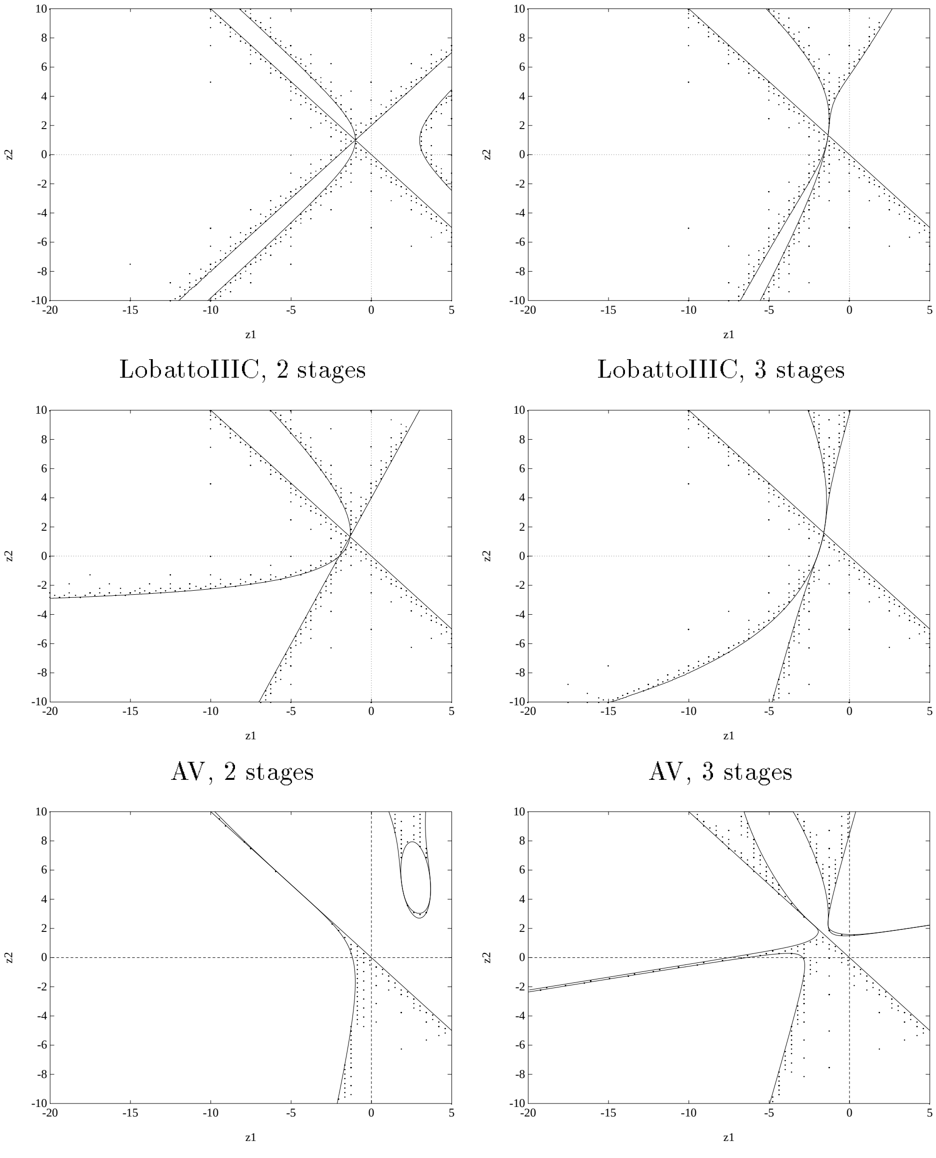


Figure 2. Extended Stability Domains II

7. BDF-Methods

We consider again the linear DAE (8) of index 2. The essential underlying ODE is given by (10) with matrices R, S and F which satisfy (2). Further, the assumptions (11) are assumed to be satisfied.

As easily can be verified, the BDF-Methods integrate the essential underlying ODE separately, because the z-values do not influence the y-values and the constraints are satisfied at all steps.

We apply one step of a BDF-Method of order s to the homogenous part of system (8) with initial values $y_1, \ldots, y_{s-1}$ on the constraints and constant stepsize h. We get

$$\begin{aligned}\sum_{i=0}^{s} \alpha_i y_i =& h(B(t_s)y_s + C(t_s)z_s)\\ 0 =& D(t_s)y_s\end{aligned}$$

with α_i given by

$$\sum_{i=0}^{s} \alpha_i \xi^i := p(\xi) := \sum_{j=1}^{s} \frac{1}{j}\xi^{s-j}(\xi-1)^j.$$

We denote the corresponding numerical solution of the EUODE by $v_i = R(t_i)y_i$. This gives

$$\sum_{i=0}^{s} \alpha_i R(t_s)y_i = hR(t_s)B(t_s)y_s$$

The consistency of the y-components allows us to substitute

$$y_i = S(t_i)v_i,$$

and we get

$$\begin{aligned}0 &= \sum_{i=0}^{s} \alpha_i(R(t_i) + h(s-i)R')S(t_i)v_i - hRBSv_s\\ 0 &= \sum_{i=0}^{s} \alpha_i(I + (s-i)z_1)v_i - z_2 v_s\end{aligned}$$

Analogous to the ODE-case we solve this difference equation for $\{v_i\}$ by the substitution $v_i = \xi^i$.

$$0 = p(\xi) + z_1(s \cdot p(\xi) - \xi p'(\xi)) - z_2\xi^s$$

We have

$$\begin{aligned}
s\,p(\xi) - \xi p'(\xi) =& \sum_{j=1}^{s} \frac{s}{j}\xi^{s-j}(\xi-1)^j - \sum_{j=1}^{s}\left(\frac{s-j}{j}\xi^{s-j}(\xi-1)^j + \xi^{s-j+1}(\xi-1)^{j-1}\right) \\
=& \sum_{j=1}^{s} \xi^{s-j}(\xi-1)^j - \sum_{j=0}^{s-1}\xi^{s-j}(\xi-1)^j \\
=& (\xi-1)^s - \xi^s
\end{aligned}$$

This gives the extended stability polynom for DAE's of index 2

$$0 = p(\xi) + z_1((\xi-1)^s - \xi^s) - z_2\xi^s. \tag{20}$$

The extended stability polynom coincides with the stability polynom for ODE's in the case $z_1 = 0$.

We define the extended stability region as the set

$$S \subset \{(z_1, z_2) \in \mathbb{C}^2\}$$

where the roots of the extended stability polynom lie inside or on the border of the unit disc with no multiple root on the border of the unit disc, i.e., the difference equation (20) has a stable solution.

To display the extended stability regions we use again the projection

$$\Im z_1 = \Im z_2 = 0.$$

We compute those points of the real (z_1, z_2) plane where the extended stability polynomial has a root on the unit circle. The resulting curves form the set of points where possibly roots leave or enter the unit disc. We use

$$\begin{aligned}
z_2 =& \xi^{-s}p(\xi) + z_1((1-\frac{1}{\xi})^s - 1) \quad |\xi| = 1 \\
z_2 =& q_1(\xi) + z_1 q_2(\xi)
\end{aligned}$$

thus

$$\begin{aligned}
0 =& \Im q_1(\xi) + z_1 \Im q_2(\xi) \\
z_2 =& \Re q_1(\xi) + z_1 \Re q_2(\xi).
\end{aligned}$$

If $\Im q_2(\xi) \neq 0$ we get a unique solution z_1, z_2. In the other case $\Im q_1(\xi) \neq 0$ is necessary to have at least one solution, but then we have a whole line as solution. For the BDF methods of order one to six the second case appears for $\xi \in \{-1, 1\}$ only. This gives the two lines

$$\begin{aligned}
z_2 =& - z_1 \\
z_2 =& (2^s - 1)z_1 + \sum_{j=1}^{s} \frac{2^j}{j}
\end{aligned}$$

The root locus curves are displayed in figure 3. The regions are denoted by the number of roots that are of modulus larger than one, i. e., the method is stable inside those regions that are denoted by number zero.

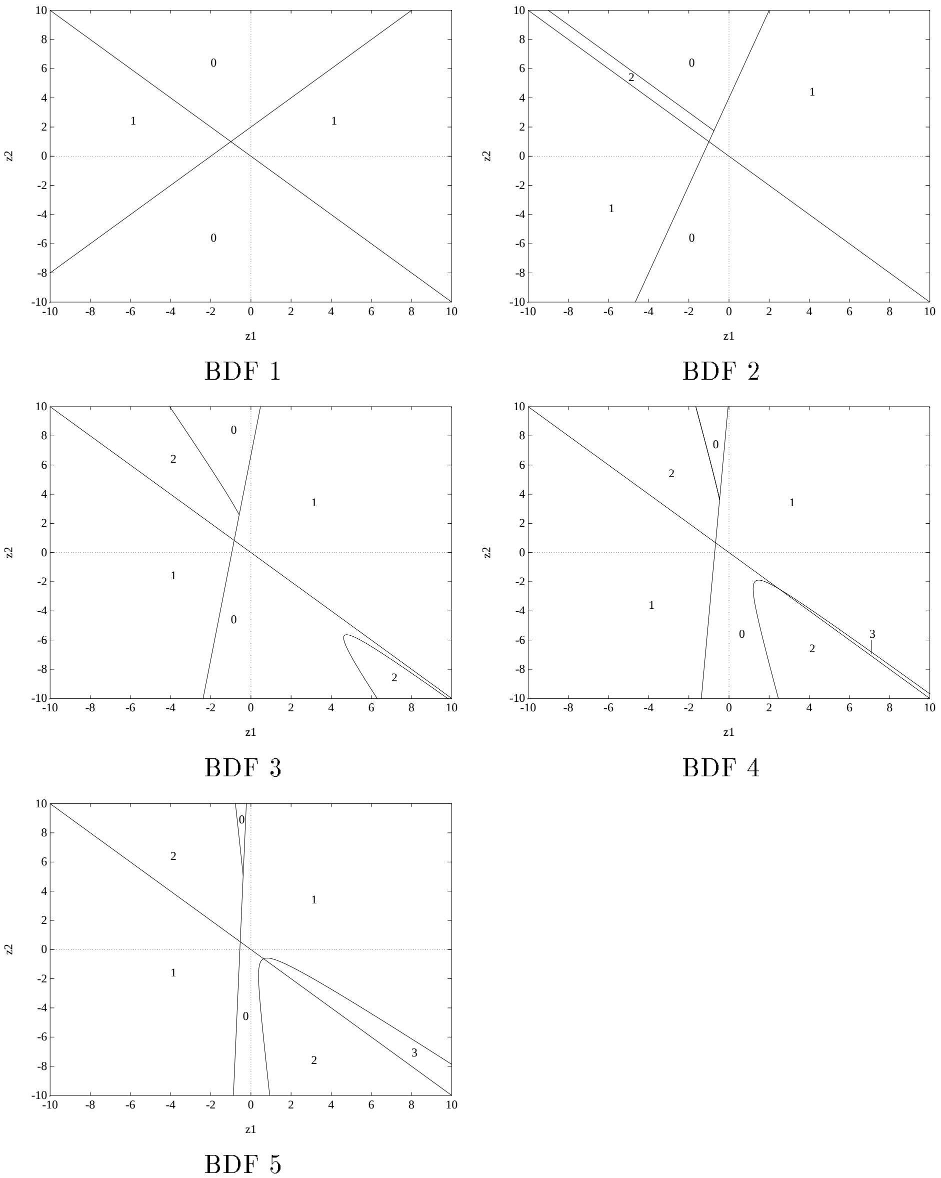


Figure 3. Extended Stability Domains III

In opposition to the Runge-Kutta methods no BDF method has an unbounded stability region on the real z_1-axis. With increasing order the stability region gets smaller, so that only a small region around the real z_2-axis remains for the BDF methods of higher order.

8. Numerical Experiments

In this section we will illustrate, that stability in the process of integration of the EUODE is necessary for the stable integration of the original DAE.

Example 1: First of all we use a test example where we can find matrices S, R which satisfy (2), (11).

$$\begin{aligned} y' =& \lambda_2 y + \begin{pmatrix} \lambda_1 t \\ 1-\lambda_1 \end{pmatrix} z + \begin{pmatrix} ((\lambda_2-\lambda_1-1)t-1)e^t \\ (\lambda_1-\lambda_2)e^t \end{pmatrix} \\ 0 =& \begin{pmatrix} \lambda_1 - 1 \\ \lambda_1 t - 1 \end{pmatrix}^T y + (1-t)e^t \end{aligned}$$

The initial values $y_1 = 0, y_2 = 1, z = 1$ give the exact solution

$$y_1 = -te^t, y_2 = e^t, z = e^t,$$

which is independent of (λ_1, λ_2). The choice

$$R = \begin{pmatrix} \lambda_1 - 1 \\ \lambda_1 t \end{pmatrix}^T, S = \begin{pmatrix} \frac{\lambda_1 t - 1}{1-\lambda_1} \\ 1 \end{pmatrix}$$

satisfies (2),(11) with $R'S = \lambda_1, RBS = \lambda_2$. The corresponding EUODE is

$$v' = (\lambda_1 + \lambda_2)v + (-(\lambda_1+\lambda_2)t + t + 1)e^t, \quad v(0) = 0$$

with the solution

$$v = te^t.$$

We computed numerical solutions from $t_0 = 0$ to $t_e = 1$ for

$$\begin{aligned} &\text{case } 1{:}\lambda_1 = -1000, \lambda_2 = -100 \\ &\text{case } 2{:}\lambda_1 = -400, \lambda_2 = -1000 \end{aligned}$$

with constant stepsizes. The Runge-Kutta equations were solved with an accuracy of 10^{-15}. The results are displayed in table 2. As expected, the methods work quite well inside the stability regions but run into trouble outside. A stable resp. unstable integration of the EUODE implies in general a stable resp. unstable integration of the DAE although there may be a strongly time-varying transformation between the solution (y, z) of the DAE and the solution v of the EUODE. This has consequences for the implementation with stepsize control. A method that is A-stable for ODE's may be unstable for large stepsizes if applied to a DAE as given here. A simple stepsize-controller will work quite inefficient if the method works on the boundary of the extended stability region. We will illustrate this.

We compute the solution of example 1 with $\lambda_1 = -1000, \lambda_2 = -700$ from $t_0 = 0$ to $t_e = 1$. The simple stepsize controller is based on an $O(h^p)$ error model. The Runge-Kutta equations are solved with residuals less than 10^{-13}. The results are displayed in figure 4. The Gauss and the Radau method give oscillating step size sequences because their extended stability domain is bounded in the direction (λ_1, λ_2). The collocation method AV gives an almost constant stepsize sequence. It has a larger extended stability region than the other two methods for this choice

Method	steps	Example 1		Example 2
		erry (case 1)	erry (case 2)	erry
AV	25	7.5E-06	3.1E+02	1.9E-06
	50	4.8E-07	6.9E-02	1.3E-07
	100	3.0E-08	1.9E-08	4.0E-09
	200	1.9E-09	1.1E-09	8.4E-11
	400	1.7E-10	2.9E-11	1.6E-12
	800	4.7E-12	3.2E-13	2.1E-14
Lobatto IIIC	25	2.5E-04	9.5E-04	5.4E-06
	50	5.9E-05	2.0E-04	6.7E-07
	100	1.3E-05	3.1E-05	8.1E-08
	200	2.8E-06	3.3E-06	9.4E-09
	400	8.7E-07	2.7E-07	1.0E-09
	800	1.5E-07	1.9E-08	1.1E-10
Radau IIA	25	6.8E-04	4.0E+06	4.2E-03
	50	3.8E-06	9.7E+06	6.3E-01
	100	6.8E-08	2.0E-08	2.2E+02
	200	4.3E-09	1.6E-09	3.0E-02
	400	1.3E-09	6.8E-11	6.8E-11
	800	9.4E-12	2.2E-12	1.2E-13
Gauss	25	7.0E+23	2.1E+36	5.1E+045
	50	1.3E+31	8.3E+48	1.4E+081
	100	6.4E+12	6.1E+25	1.3E+134
	200	2.1E-09	1.7E-09	1.2E+196
	400	4.7E-10	2.4E-11	3.5E+205
	800	9.6E-13	3.5E-13	2.2E+026

Table 2. Error in y-components at $t = 1$, computed with constant stepsize for projected implicit Runge-Kutta methods with 3 stages.

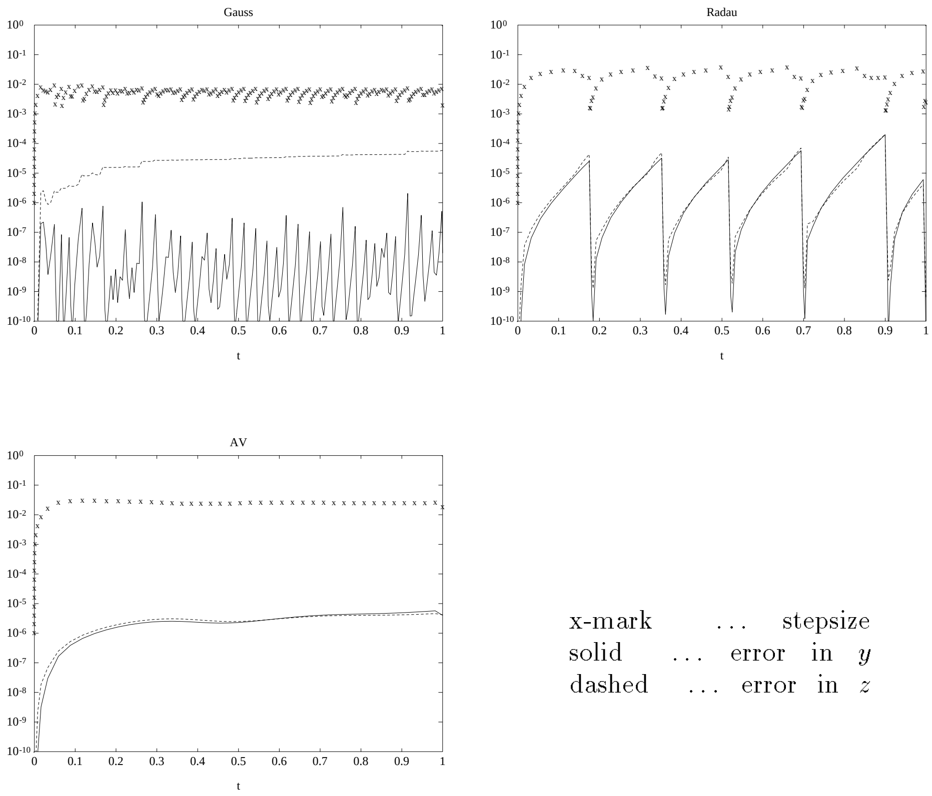


Figure 4. Numerical results, stepsize control

of λ-values. The stepsize is therefore determined by the desired accuracy and not by stability.

Example 2: This system comes from [1].

$$\begin{aligned} y_1' =& (2-t)\nu z + (1+\nu)e^t \\ y_2' =& (\nu - 1)z + \left(1 + \frac{\nu - 1}{2-t}\right) e^t \\ 0 =& (t+2)y_1 + (t^2-4)y_2 - (t^2+t-2)e^t \end{aligned}$$

with initial values

$$y_1(0) = 1, \quad y_2(0) = 1, \quad z(0) = -1/2.$$

The exact solution is given by

$$y_1 = e^t, \quad y_2 = e^t, \quad z = -\frac{1}{2-t}e^t.$$

With

$$R = (1 - \nu, (2 - t)\nu)^T$$

we get

$$v' = -\frac{\nu}{2 - t} v$$

as the homogenous part of the EUODE. The coefficients of the EUODE are timedepending now, i. e., this system does not coincide with our model-problem anymore. The corresponding value λ_1 varies from $-\nu/2$ to $-\nu$, and λ_2 vanishes.

We solved this system in $(0, 1)$ with the same set of methods as in example one. The stiffness-parameter ν was set to 10^4. The results are given in table 2.

The Gauss-method runs into trouble because its extended stability region is bounded by ≈ 3, i. e., $h\nu \leq 3$. Although the stability region of the Radau-method is bounded too, this method performs quite well. There are two reasons for that: The extended stability region of the Radau-method is bounded by a value of ≈ 32 and the value of the extended stability function is $3/2$ at infinity, so that for a small number of steps only a mild error accumulation is recognized.

The other methods perform well. Their stability region is not bounded for this choice of λ-values.

We want to make some remarks on the solution of the Runge-Kutta equations. In our tests the Runge-Kutta equations were solved by Newtons method (not simplified). The reason is, that the Jacobian changes rapidly. If the same Jacobian is used for the equations of all stages, then the convergence rate is really weak or vanishes with increasing stepsizes. The stability restrictions to the stepsize are replaced by convergence restrictions. We computed example 2 with the RADAU5 code of Hairer and Wanner [5]. This results in $1.8 \cdot 10^5$ accepted steps and $2.7 \cdot 10^5$ rejections because of bad convergence in the simplified Newton iteration by a tolerance of 10^{-4} and $\nu = 10^4$. The code kept the value of $h\nu$ on approximately 0.05 for different values of v. That is clear inside the stability region (the stability region would allow values up to 32).

Summary: The numerical treatment of DAE's of index two with stiff EUODE is complicated, especially if the term $R'S$ is large. By scaling these DAE's they may be interpreted as 'almost' index three DAE's, because $\frac{\|(DC)\|}{\|D\|\|C\|} << 1$, i.e., the index two condition is 'almost' violated. Methods, that are A-stable in common sense, do not allow arbitrary stepsizes if applied to those equations.

In the special case $RBS = 0$ the AV-method gives best performance. For other choices of $R'S, RBS$ may be constructed special methods with unbounded stability regions that work quite well, but this is still an open question. Especially the case $R'S \approx RBS < 0$ seems to be a problem to every method.

References

1. U. Ascher and L. Petzold, *Projected Implicit Runge-Kutta Methods for Differential-Algebraic Equations.* Siam J. Numer. Anal., 28 (1991), pp. 1097-1120
2. U. Ascher and L. Petzold, *Stability of Computational methods for constrained dynamics systems.* Siam J. Sci. Comput., 14(1993), pp. 95-120
3. K. Brenan, S. Campbell and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* North-Holland, 1989
4. E. Griepentrog and R. März, *Differential-Algebraic Equations and Their Numerical Treatment.* Leipzig, 1986

5. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II.* (1991), Berlin, Heidelberg, New York, 1991
6. E. Hairer, C. Lubich and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods.* Springer LNM 1409, 1989
7. C. Lubich, *On Projected Runge-Kutta methods for diferential algebraic equations.* BIT 31(1991)
8. J. Wensch *Stabilität von Runge-Kutta Methoden für Algebrodifferentialgleichungen vom Index 2.* Master thesis, Halle 1992