Monografías del Semin. Matem. García de Galdeano. **27:** 345–352, (2003).

ESDIRK methods for index-2 DAE: starting algorithms

Inmaculada Higueras and T. Roldán

Departamento de Matemática e Informática, Universidad Pública de Navarra

higueras@unavarra.es, teo@unavarra.es

Abstract

When semi-explicit differential-algebraic equations are solved with implicit Runge-Kutta methods, the computational effort is dominated by the cost of solving the non-linear systems and therefore it is important to have good starting values to begin the iterations. In this paper we show how to construct an efficient starting algorithm, without additional computational cost, for a class of Runge-Kutta methods in the case of index-2 DAEs.

Keywords: Starting algorithms, differential-algebraic equations, ESDIRK methods
AMS Classification: 65L05, 65L06, 65L80, 65H10.

1 Introduction.

We consider semi-explicit index-2 differential algebraic systems of the form

$$\begin{cases} y' = f(y,z) & y(x_0) = y_0 \\ 0 \ = g(y) & z(x_0) = z_0 \ , \end{cases} \tag{1}$$

where $f : \mathbb{R}^l \times \mathbb{R}^m \longrightarrow \mathbb{R}^l$ and $g : \mathbb{R}^l \longrightarrow \mathbb{R}^m$ are sufficiently smooth functions, and the matrix $g_y f_z$ is invertible in a neighbourhood of the solution of (1). Furthermore, we assume that the initial values are consistent, i. e., they satisfy the algebraic equation $g(y_0) = 0$ and the hidden constrain $g_y(y_0) f(y_0, z_0) = 0$.

If we consider an s-stage Runge-Kutta method $(\mathcal{A}, b)$ to solve (1), a standard assumption is the matrix $\mathcal{A}$ to be regular. However we can also use methods with singular matrices of the form

$$\begin{array}{c|cc} 0 & 0 & 0^t \\ \bar{c} & a & \bar{\mathcal{A}} \\ \hline & b_1 & \bar{b}^t \end{array} \tag{2}$$

where $a \in \mathbb{R}^{s-1}$ and $\bar{\mathcal{A}}$ is an $(s-1)\times(s-1)$ regular matrix [6]. In particular ESDIRK (Explicit Single Implicit RK) methods belong to this class. On these methods we assume that conditions $C(1)$ and $B(1)$ hold, i.e.

$$a + \bar{\mathcal{A}}\bar{e} = \bar{c} \qquad\qquad b_1 + \bar{b}^t\bar{e} = 1 \tag{3}$$

where $\bar{e} = (1,\ldots,1)^t \in \mathbb{R}^{s-1}$, and that $R(\infty)$ is bounded, with $R(z)$ the stability function of the method, i.e. we impose

$$b_1 - \bar{b}^t\bar{\mathcal{A}}^{-1}a = 0\,. \tag{4}$$

In this way (3) imply

$$\bar{b}^t\bar{\mathcal{A}}^{-1}\bar{c} = 1\,.$$

For these methods, the first internal stages are $Y_{n+1,1} = y_n$, $Z_{n+1,1} = z_n$, and the rest of the stages are given by the non-linear system

$$\begin{aligned} \bar{Y}_{n+1} &= \bar{e}\otimes y_n + h\,a\otimes f(y_n,z_n) + h(\bar{\mathcal{A}}\otimes I_l)f(\bar{Y}_{n+1},\bar{Z}_{n+1})\,, &(5)\\ 0 &= g(\bar{Y}_{n+1})\,. &(6)\end{aligned}$$

We have denoted $\bar{Y}_{n+1} = (Y^t_{n+1,2},\ldots,Y^t_{n+1,s})^t \in \mathbb{R}^{l(s-1)}$, $\bar{Z}_{n+1} = (Z^t_{n+1,2},\ldots,Z^t_{n+1,s})^t \in \mathbb{R}^{m(s-1)}$; $f(\bar{Y}_{n+1},\bar{Z}_{n+1}) \in \mathbb{R}^{l(s-1)}$ is the vector $[f(Y_{n+1,2},Z_{n+1,2})^t,\ldots,f(Y_{n+1,s},Z_{n+1,s})^t]^t$ and in an analogous way for $g(\bar{Y}_{n+1})$. The symbol $\otimes$ denotes the Kronecker product.

As the matrix $\bar{\mathcal{A}}$ is regular, system (5)-(6) can be solved for $\bar{Y}_{n+1}$ and $\bar{Z}_{n+1}$. Once these values have been computed, with condition (4), we obtain

$$y_{n+1} = R(\infty)y_n + (\bar{b}^t\bar{\mathcal{A}}^{-1}\otimes I_l)\,\bar{Y}_{n+1}\,,$$

and similarly we can compute

$$z_{n+1} = R(\infty)z_n + (\bar{b}^t\bar{\mathcal{A}}^{-1}\otimes I_m)\,\bar{Z}_{n+1}\,.$$

If the method is stiffly accurate, i.e. $a_{si} = b_i$, $i = 1,\ldots,s$, we simply obtain

$$y_{n+1} = \bar{Y}_{n+1,s} \qquad z_{n+1} = \bar{Z}_{n+1,s}\,.$$

Observe that in this case the numerical solution satisfies $g(y_{n+1}) = 0$. If the method is not stiffly accurate, the numerical solution must be projected onto the constraint $g(y) = 0$ (see [1], [6]). Examples of stiffly accurate methods of the form (2) are Lobatto IIIA methods and SDIRK methods considered in [2] and [9].

In each step, we have to obtain the internal stage vectors $\bar{Y}_{n+1}$ and $\bar{Z}_{n+1}$ through the resolution of the non-linear system (5)-(6). In the iterative scheme to solve the nonlinear system, we need starting values $(\bar{Y}^{(0)}_{n+1},\bar{Z}^{(0)}_{n+1})$ as accurate as possible, because in other case,

the number of iterations in each step may be too high or even worse, the convergence may fail.

In [7] and [8] a type of initializers for index-1 DAEs was studied, and in [5], [7] they were extended to the case of index-2 and index-3 DAEs. For the index-1 case, the coefficient matrix $\mathcal{A}$ is not assumed to be regular and thus the study made covers methods of the type (2). For example in [7], [8], the coefficients of the starting algorithms for the Lobatto IIIA methods with two and three stages were given. For index-2 and index-3 DAEs, the order conditions given in [5], [7] involve the inverse of $\mathcal{A}$ and consequently those results are not longer valid.

In this paper we study initializers to obtain starting values for the internal stages when methods of the form (2) are used. In each step these starting values will be obtained using the information from the previous step. We are going to assume that we have just given a step $x_{n-1} \xrightarrow{h} x_n$ from the consistent initial values (y_{n-1}, z_{n-1}), we have calculated the numerical solution (y_n, z_n) at x_n, as well as the internal stages $(\bar{Y}_n, \bar{Z}_n)$, and we are about to give another step $x_n \xrightarrow{rh} x_{n+1}$ to compute the numerical solution (y_{n+1}, z_{n+1}). To achieve this, we have to solve the non-linear system (5)-(6) but now with step hr instead of h in order to consider the most general case of variable step.

The rest of the paper is organized as follows. Section 2 begins with a review of the work done in [5], [7] for index-2 DAEs. In previous papers ([3], [5], [8]) it has been proved that the use of a high order starting algorithm improves the implementation of the method. The results obtained in [5], [7] are transferred to (2) embedding this method into one with regular coefficient matrix. Finally in Section 4 we test the algorithm proposed in some numerical experiments.

2 Starting algorithms

Given an s-stages Runge-Kutta method $(\mathcal{A}, b)$ with $\mathcal{A}$ regular, the starting algorithms considered in [5] are of the form

$$Y_{n+1}^{(0)} = b_0 \otimes y_{n-1} + (B \otimes I_l) Y_n \,, \tag{7}$$

$$Z_{n+1}^{(0)} = c_0 \otimes z_{n-1} + (C \otimes I_m) Z_n \,, \tag{8}$$

where b_0, $c_0 \in \mathbb{R}^s$, and B and C are $s \times s$ matrices which have to be determined.

We say that the starting formula (7)-(8) has order (r_y, r_z) if these are the largest integers which satisfy

$$||Y_{n+1}^{(0)} - Y_{n+1}|| = \mathcal{O}(h^{r_y+1}) \,, \qquad ||Z_{n+1}^{(0)} - Z_{n+1}|| = \mathcal{O}(h^{r_z+1}) \,.$$

The vectors b_0, c_0 and the matrices B, C are determined so that these algorithms achieve the maximum possible order in each variable. To obtain this, we need the series both for the initializers $(Y_{n+1}^{(0)}, Z_{n+1}^{(0)})$ and for the internal stages (Y_{n+1}, Z_{n+1}). With the help of rooted trees, in [5] these series in powers of h are given in terms of certain functions $\Phi_y(t)$, $\Phi_z(u)$, $\bar{\Phi}_y(t)$, $\bar{\Phi}_z(t)$, which are defined recursively. Comparing these series, it is possible to determine the order conditions for the starting algorithm. In particular, in [5] it is proved that (7) reaches order r_y for the differential variable if this is the largest integer which satisfies

$$b_0 + Be = e\,, \tag{9}$$

$$B\Phi_y(t) = \bar{\Phi}_y(t) \qquad \forall t \in DAT2_y \quad \text{with} \quad 1 \le \rho(t) \le r_y\,, \tag{10}$$

and (8) reaches order r_z for the algebraic variable if this is the largest integer which satisfies

$$c_0 + Ce = e\,, \tag{11}$$

$$C\Phi_z(u) = \bar{\Phi}_z(u) \qquad \forall u \in DAT2_z \quad \text{with} \quad 1 \le \rho(u) \le r_z\,. \tag{12}$$

Recall that the study done in [5] is valid only for regular matrices $\mathcal{A}$ and thus it is not directly applicable to the methods considered in this paper (2). In order to make use of those results, we embed the method (2) into the ε-method

$$\begin{array}{c|cc} \varepsilon & \varepsilon & 0^t \\ \bar{c} & a & \bar{\mathcal{A}} \\ \hline & b_1 & \bar{b}^t \end{array} \qquad = \qquad \begin{array}{c|c} c_\varepsilon & \mathcal{A}_\varepsilon \\ \hline & b^t \end{array}\,.$$

If $\varepsilon \neq 0$, the coefficient matrix is regular and thus we can try to apply the results in [5], [7].

The internal stages $Y_{n,\varepsilon}$, $Z_{n,\varepsilon}$ for this numerical method converge to $(y_{n-1}, \bar{Y}_n)$ and $(z_{n-1}, \bar{Z}_n)$ when ε tends to zero. The idea is to construct starting algorithms like (7-8) for the ε-method, and take the limit when ε tends to zero.

The starting algorithm for the ε-method will be constructed imposing the order conditions obtained in [5], i.e.

$$b_0 + Be = e\,, \tag{13}$$

$$B\Phi_{y,\varepsilon}(t) = \bar{\Phi}_{y,\varepsilon}(t) \qquad \forall t \in DAT2_y \quad \text{with} \quad 1 \le \rho(t) \le r_y\,, \tag{14}$$

for the differential variable and

$$c_0 + Ce = e\,, \tag{15}$$

$$C\Phi_{z,\varepsilon}(u) = \bar{\Phi}_{z,\varepsilon}(u) \qquad \forall u \in DAT2_z \quad \text{with} \quad 1 \le \rho(u) \le r_z\,. \tag{16}$$

for the algebraic one. The expressions involved in the order conditions (14)-(16), namely $\Phi_{y,\varepsilon}$, $\Phi_{z,\varepsilon}$, $\bar{\Phi}_{y,\varepsilon}$ and $\bar{\Phi}_{z,\varepsilon}$, are bounded when ε tends to zero. Recall that the matrix $\mathcal{A}_\varepsilon^{-1}$,

$$\mathcal{A}_\varepsilon^{-1} = \begin{pmatrix} \varepsilon & 0 \\ a & \bar{A} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{\varepsilon} & 0 \\ -\frac{1}{\varepsilon}\bar{A}^{-1}a & \bar{A}^{-1} \end{pmatrix},$$

which contains the term $1/\varepsilon$, is involved in the definition of these functions. For example, for the tree $u_{q+1,1} =$  , the function $\bar{\Phi}_{z,\varepsilon}(u_{q+1,1})$ is given by

$$\bar{\Phi}_{z,\varepsilon}(u_{q+1,1}) = \frac{1}{q+2}\,\frac{1}{r}\,\mathcal{A}_\varepsilon^{-1}[-eb^t\mathcal{A}_\varepsilon^{-1}c_\varepsilon^{q+2} + (e + rc_\varepsilon)^{q+2}]$$

The important point is to observe that we do not have to deal with $\mathcal{A}_\varepsilon^{-1}$ but with this matrix multiplied by certain vectors. For example, for the tree $u_{q+1,1}$ with $q = 1$, the expression that must be bounded when ε tends to zero is

$$\mathcal{A}_\varepsilon^{-1}c_\varepsilon^2 = \begin{pmatrix} \varepsilon \\ -\varepsilon\bar{A}^{-1}a + \bar{A}^{-1}\bar{c}^2 \end{pmatrix}.$$

In [4] a detailed study on the boundedness of the functions Φ_ε and $\bar{\Phi}_\varepsilon$ is done.

3 Example

In this section we show how to constuct a starting algorithms for a concrete method. This process can be easily done with a symbolic manipulator like Mathematica.

We consider the stiffly accurate methods considered in [9] satisfying $B(3)$ and $C(2)$,

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 2\lambda & \lambda & \lambda & 0 & 0 \\ c_3 & \dfrac{6c_3\lambda - 4\lambda^2 - c_3^2}{4\lambda} & \dfrac{c_3u_1}{4\lambda} & \lambda & 0 \\ 1 & \dfrac{12u_2\lambda^2 + 6u_3\lambda - u_3}{12c_3\lambda} & \dfrac{6\lambda u_2 + u_3}{12\lambda u_1} & \dfrac{6\lambda^2 - 6\lambda + 1}{3c_3u_1} & \lambda \\ \hline \end{array} \tag{17}$$

where

$$u_1 = c_3 - 2\,\lambda \qquad u_2 = 1 - c_3\,, \qquad u_3 = 3c_3 - 2\,,$$

and $c_3 \neq 0, 2\lambda$. With

$$c_3 = \frac{2\lambda(\lambda - 1/4)(\lambda - 1)}{(\lambda - 1/2)^2 - 1/12}$$

the method has order 3 for the differential variable and 2 for the algebraic one for index-2 DAEs. For $\lambda \approx 0.43586652$, the method is L-stable.

The order conditions up to 3 for the differential variable are

$$\begin{aligned} &Be = e - b_0\,, && Bc = e + rc\,,\\ &Bc^2 = (e + rc)^2\,, && Bc^3 = (e + rc)^3\,,\\ &B\mathcal{A}\,c^2 = eb^t c^2 + r\mathcal{A}(e + rc)^2\,. \end{aligned}$$

whereas the order conditions up to order 2 for the differential variable are

$$\begin{aligned} &Ce = e - c_0\,, && Cc = e + rc\,,\\ &Cc^2 = (e + rc)^2\,, && C\mathcal{A}^{-1}c^3 = \tfrac{1}{r}\,\mathcal{A}^{-1}[-eb^t\mathcal{A}^{-1}c^3 + (e + rc)^3]\,. \end{aligned}$$

For the differential variable, from the conditions

$$\begin{aligned} B_\varepsilon e &= e - b_0\,,\\ B_\varepsilon c_\varepsilon &= e + rc_\varepsilon\,,\\ B_\varepsilon c_\varepsilon^2 &= (e + rc_\varepsilon)^2\,, \end{aligned}$$

we obtain a family of predictors with order 2. Imposing one of the two conditions of order 3 we can obtain a unique predictor

$$B_\varepsilon = \bar{V}_\varepsilon\ V_\varepsilon^{-1}$$

where $V_\varepsilon = (e, c_\varepsilon, c_\varepsilon^2, c_\varepsilon^3)$ and $\bar{V}_\varepsilon = (e, e + rc_\varepsilon, (e + rc_\varepsilon)^2, (e + rc_\varepsilon)^3)$. Taking the limit when ε tends to zero, we get

$$B = \lim_{\varepsilon\to 0} B_\varepsilon = \begin{pmatrix} 0 & 0 & 0 & 1\\ b_{21}(r) & b_{22}(r) & b_{23}(r) & b_{24}(r)\\ b_{31}(r) & b_{32}(r) & b_{33}(r) & b_{34}(r)\\ b_{41}(r) & b_{42}(r) & b_{43}(r) & b_{44}(r) \end{pmatrix}. \tag{18}$$

where the $b_{ij}(r)$ are polynomials of degree three.

For the algebraic variable, up to order 2, the four order conditions can be written as

$$C_\varepsilon W_\varepsilon = \bar{W}_\varepsilon\,,$$

where $W_\varepsilon = (e, c_\varepsilon, c_\varepsilon^2, {\mathcal{A}_\varepsilon}^{-1}c_\varepsilon^3)$ and

$$\bar{W}_\varepsilon = (e, e + rc_\varepsilon, (e + rc_\varepsilon)^2, \frac{1}{r}\,{\mathcal{A}_\varepsilon}^{-1}[-eb^t{\mathcal{A}_\varepsilon}^{-1}c_\varepsilon^3 + (e + rc_\varepsilon)^3])\,.$$

The matrix W_ε is regular if and only if $\alpha = \lim_{\varepsilon\to 0} e_4^t V_\varepsilon^{-1}{\mathcal{A}_\varepsilon}^{-1}c_\varepsilon^3 \neq 0$, where $e_4 = (0, 0, 0, 1)^t$. For the method considered $\alpha \approx -46.48$, and thus there exists a unique predictor of order 2. The matrix $C = \lim C_\varepsilon$ is given by

$$C = \begin{pmatrix} 0 & 0 & 0 & 1\\ 0.0015r + 0.7450r^2 & -3.7551r - 2.7551r^2 & 2.5463(r + r^2) & 1 + 1.2072r - 0.5363r^2\\ 0.0020r + 1.3097r^2 & -4.9701(r + r^2) & 3.3703r + 4.3703r^2 & 1 + 1.5978r - 0.7098r^2\\ 0.0018r + 1.0012r^2 & -4.3076(r + r^2) & 2.9210(r + r^2) & 1 + 1.3848r + 0.3848r^2 \end{pmatrix} \tag{19}$$

4 Numerical experiments

For the numerical experiments we have considered the ESDIRK method (17) with the predictor (19) for both variables. This predictor achieves order 2 for the algebraic variable but also order 2 for the differential one. We have also considered the trivial predictor used with this ESDIRK method in [9]. In order to test these two predictors we have considered the following index-2 problems. We have run the codes with different steps and different tolerances. The results have been summarized in tables 1 and 2 .

Problem 1. First we have considered a simple scalar index-2 problem

$$\begin{cases} y' = 2\,y/z\,, & y(0) = 1\,, \\ 0 = y^2 - 1 - \sin t\,, & z(0) = 4\,, \end{cases} \qquad t \in [0,1]\,. \tag{20}$$

The exact solution is $y(t) = \sqrt{1+\sin t}\,e^{-t}$, $z(t) = \frac{4\,(1+\sin t)}{\cos t} - 3\,e^{-t}$. As we know the exact solution, we have used the criteria

$$||y(t_n) - y_n|| < C * h^3 \quad \text{and} \quad ||z(t_n) - z_n|| < C * h^2$$

to stop the iterations. Remember that the ESDIRK method considered achieves order 3 for the differential variable and order 2 for the algebraic one. In Table 1, in the last two columns, we have put the number of iterations per stage needed to achieve the tolerance required for the predictors considered.

h	steps	Trivial predictor	Predictor (19)
0.01	10	3.2100	1.0133
0.001	1000	3.2117	1.0013
0.0001	10000	3.2117	1.0001

Table 1: Results for problem 1

Problem 2. This problem is a mechanical system formulated in the form (1)

$$\begin{cases} y_1' = y_3\,, & y_1(0) = 1\,, \\ y_2' = y_4\,, & y_2(0) = 0\,, \\ y_3' = y_3^2 - y_1 - 2\,\lambda\,y_1\,, & y_3(0) = 0\,, \\ y_4' = 10\,y_4^2 - 20\,y_2 - 2\,\lambda\,y_2\,, & y_4(0) = 0.3\,, \\ 0 = y_1\,y_3 + y_2\,y_4\,, & \lambda(0) = \lambda_0\,, \end{cases} \qquad t \in [0, 0.5] \tag{21}$$

The value of λ_0 has been taken such that the initial conditions are consistent. In this case, we have used the criteria $||\,Y_n^{(k)} - Y_n^{(k-1)}|| < TOL$ to stop the iterations, where Y_n denotes the internal stages.

As it occurred in the previous problem, we obtain better results with predictor (19) . Observe that Newton iterations are reduced to the minimum possible value with this

h	steps	TOL	Trivial predictor	Predictor (19)
0.01	50	1.0E-3	3.3067	1.02667
0.001	500	1.0E-6	2.3780	1.0040
0.0001	5000	1.0E-8	2.3015	1.0004

Table 2: Results for problem 2

predictor. This reduction of the Newton iterations implies a reduction in the CPU-time, in this case the gain is about 50%.

References

[1] Ascher, U. and L. Petzoldt: 1991, 'Projected implicit Runge-Kutta methods for differential-algebraic equations'. *SIAM J. Numer. Anal.* **28**, 1097–1120.

[2] Cameron, F.: 1999, 'A class of low order DIRK methods for a class of DAEs'. *Applied Numerical Mathematics* **31**, 1–16.

[3] Higueras, I. and T. Roldán: 2000, 'Starting algorithms for some DIRK methods'. *Numerical Algorithms* **23**, 357–369.

[4] Higueras I. and T. Roldán: 2001 'Starting algorithms for a class of RK methods for index-2 DAE'. Preprint, Universidad Pública de Navarra, n. 4, 2001.

[5] Higueras, I. and T. Roldán: 2003, 'IRK Methods for index 2 and 3 DAEs: Starting algorithms'. *BIT* **43-1**, 65–90.

[6] Jay, L.: 1993, 'Convergence of a class of Runge-Kutta methods for differential-algebraic systems of index 2'. *BIT* **33**, 137–150.

[7] Roldán, T.: 2000, 'Implicit Runge-Kutta methods for DAEs: starting algorithms'. Ph.D. thesis, Departamento de Matemática e Informática, Universidad Pública de Navarra, Spain.

[8] Roldán, T. and I. Higueras: 1999, 'IRK methods for DAE: Starting algorithms'. *J. Comput. Appl. Math.* **111**, 77–92.

[9] Williams, R., K. Burrage, I. Cameron, and M. Kerr: 1999, 'A Four-Stage Index 2 Diagonally Implicit Runge-Kutta Method'. *Applied Numerical Mathematics* **40-3**, 415-432.