The Numerical Solution of Delay-Differential-Algebraic Equations of Retarded and Neutral Type

by
Uri M. Ascher and
Linda R. Petzold

Technical Report 92-19
December 1992

Department of Computer Science
University of British Columbia
Rm 333 - 6356 Agricultural Road
Vancouver, B.C.
CANADA V6T 1Z2

Telephone: (604) 822-3061
Fax: (604) 822-5485

The Numerical Solution of Delay-Differential-Algebraic Equations of Retarded and Neutral Type

Uri M. Ascher
Department of Computer Science
University of British Columbia
Vancouver, British Columbia
Canada V6T 1Z2 *
and
Linda R. Petzold
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455 †

December 7, 1992

Abstract

In this paper we consider the numerical solution of initial value delay-differential-algebraic equations (DDAEs) of retarded and neutral types, with a structure corresponding to that of Hessenberg DAEs. We give conditions under which the DDAE is well-conditioned, and show how the DDAE is related to an underlying retarded or neutral delay-ODE (DODE). We present convergence results for linear multistep and Runge-Kutta methods applied to DDAEs of index 1 and 2, and show how higher-index Hessenberg DDAEs can be formulated in a stable way as index-2 Hessenberg DDAEs.

*The work of this author was partially supported under NSERC Canada Grant OGP 0004306.
†The work of this author was partially supported by ARO, contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center, and by ARO contract number DAAL03-92-G-0247, DOE contract number DE-FG02-92ER25130 and NIST contract number 60NANB2D1272.

1 Introduction

Recently there has been much work on the numerical solution of systems of differential-algebraic equations (DAEs) [9], [16]. These systems, which are given most generally as $F(t, y, y') = 0$, arise in a wide variety of scientific and engineering applications including circuit analysis, computer-aided design and real-time simulation of mechanical (multibody) systems, power systems, chemical process simulation, optimal control, etc. In some situations, for example in real-time simulation, where time delays can be introduced by the computer time needed to compute an output after the input has been sampled, and where additional delays can be introduced by the operator-in-the loop [13], differential equations with delays must be included in the model. Delays arise also in circuit simulation and power systems, due for example to interconnects for computer chips [18] and transmission lines [20], and in chemical process simulation when modeling pipe flow [21]. Although there is an extensive literature on the mathematical structure of delay-ODEs (see [8] for an introduction) and on numerical methods for some of these systems (a brief introduction is given in [15]), we are aware of very little work on the structure of singular (DAE) systems with delays [10, 11, 12], and of virtually no work on the numerical solution of these systems. Delay-DAE (DDAE) systems arise when DAE systems from circuits or power systems or mechanical or chemical systems are subject to delays. It is the purpose of this work to study the conditioning of some of these systems and their numerical solution.

The *index* of a DAE is a measure of the degree of singularity of the system and is widely regarded also as an indication of certain difficulties for numerical ODE systems [9]. DAEs of higher-index (index > 1) are in a sense ill-posed. Fortunately, most DAEs arising in applications are in semi-explicit form, which allows more opportunity for developing general-purpose methods, and many are in the further restricted *Hessenberg* form[1] [9]. Still, even in this restricted form DAEs of index ≥ 2 present many challenges to designers of numerical methods [9]. The index-one semi-explicit DAE is given by

$$x' = f(x, y) \tag{1.1a}$$
$$0 = g(x, y) \tag{1.1b}$$

where $\frac{\partial g}{\partial y}$ is nonsingular.

The index-2 Hessenberg DAE is given by

$$x' = f(x, y) \tag{1.2a}$$

[1] An alternative for the name Hessenberg form is a *pure* form of a certain index: such a DAE contains no subsystems of a lower index with respect to the algebraic variables.

$$0 = g(x) \tag{1.2b}$$

where $\frac{\partial g}{\partial x}\frac{\partial f}{\partial y}$ is nonsingular.

The Hessenberg index-3 DAE is given by

$$y' = f(x,y,z) \tag{1.3a}$$
$$x' = g(x,y) \tag{1.3b}$$
$$0 = h(x) \tag{1.3c}$$

where $\frac{\partial h}{\partial x}\frac{\partial g}{\partial y}\frac{\partial f}{\partial z}$ is nonsingular.

Semi-explicit index-one systems arise in a wide variety of applications including most circuit analysis and power systems problems. Some examples of Hessenberg index-two systems are modeling of incompressible fluids (following spatial discretization), and some index-2 formulations of mechanical systems [3]. Hessenberg index-3 DAEs arise in the simulation of mechanical systems and in optimal control. For a variety of reasons, systems of index-3 and higher have proven to be very difficult to solve numerically [9], and much recent work has focused instead on reformulating these systems as index-2 or lower. Hence for our numerical results we will focus on delay-DAEs of index one and two in pure (Hessenberg) form.

A great deal is known about the structure of delay-ODEs [8], [17]. These systems are classified by their *type*. For a scalar delay-ODE (DODE)

$$ax'(t-1) + bx'(t) + cx(t-1) + dx(t) = f(t)$$

the system is of retarded type if $a = 0,\ b \neq 0$, of neutral type if $a \neq 0,\ b \neq 0$, and of advanced type if $a \neq 0, b = 0$ and $d \neq 0$. One of the important attributes of the type is that it classifies how DODEs propagate discontinuities to future delay-intervals (assuming an initial value problem). Discontinuities in retarded systems become smoother in each successive interval, whereas discontinuities in advanced systems become less smooth in each successive interval. Discontinuities in neutral systems are carried into successive delay intervals with the same degree of smoothness. Hence, we wish to study separately DDAEs which are equivalent to retarded and neutral DODEs, but to avoid altogether those which lead to DODEs of advanced type.

In this paper we study delay-DAEs (DDAE) of retarded type which are extensions of Hessenberg form. These DDAE systems are given by [2]

$$x' = f(x, x(t-1), y, y(t-1)) \tag{1.4a}$$

[2]We use an autonomous form for the nonlinear systems considered, without loss of generality, simply to keep the notation concise.

$$0 = g(x, x(t-1), y), \tag{1.4b}$$

(where $\frac{\partial g}{\partial y}$ is nonsingular) for index-one,

$$x' = f(x, x(t-1), y) \tag{1.5a}$$
$$0 = g(x), \tag{1.5b}$$

(where $\frac{\partial g}{\partial x}\frac{\partial f}{\partial y}$ is nonsingular) for index-two and

$$y' = f(x, x(t-1), y, y(t-1), z) \tag{1.6a}$$
$$x' = f(x, x(t-1), y) \tag{1.6b}$$
$$0 = h(x), \tag{1.6c}$$

(where $\frac{\partial h}{\partial x}\frac{\partial g}{\partial y}\frac{\partial f}{\partial z}$ is nonsingular) for index-three. The delays are allowed only in certain variables as described above, because allowing delays in the other variables/equations leads to equations of neutral or advanced type (see Appendix A). For some interesting examples of DDAEs, and how some DDAEs which "look like" they should be of retarded type but are actually neutral or advanced-type, see Campbell [11].

We further consider cases where g in (1.4b) is allowed to depend on $y(t-1)$ and where g in (1.5b) is allowed to depend on $x(t-1)$. These extensions lead to equations of a neutral type, as explained in Appendix A.

In this paper we investigate the conditioning and convergence of numerical methods for initial-value problems for retarded Hessenberg DDAEs (1.4), (1.5), (1.6) and their extensions to neutral cases. In Section 2 we define a delay-essential-underlying-ODE (DEUODE) for the DDAE and show that the DDAE is well-conditioned when the DEUODE is stable. In Section 3 we investigate the convergence and order for numerical methods such as backward differentiation formulae (BDF) and projected implicit Runge-Kutta (PIRK) [2] applied to index-one and index-two retarded and neutral Hessenberg DDAEs. In Section 4 we show how to reformulate a higher-index Hessenberg DDAE so that the numerical method is stable for well-conditioned problems.

2 Conditioning for higher-index delay-DAE

In this section we first consider the DDAE of order m

$$x^{(m)} = f(z(x(t)), z(x(t-1)), y) \tag{2.1a}$$

$$0 = g(x) \tag{2.1b}$$

where $f : U_1 \to V_1$, $g : U_2 \to V_2$, $U_1 \subseteq \mathcal{R}^{mn_x} \times \mathcal{R}^{mn_x} \times \mathcal{R}^{n_y}$, $V_1 \subseteq \mathcal{R}^{n_x}$, $U_2 \subseteq \mathcal{R}^{n_x}$, $V_2 \subseteq \mathcal{R}^{n_y}$,

$$z_j(t) = x^{(j-1)}(t) = \frac{d^{j-1}x(t)}{dt^{j-1}} \qquad (1 \le j \le m) \tag{2.2a}$$

$$z(x) = (x^T, x'^T, \ldots, x^{(m-1)^T})^T \tag{2.2b}$$

and $g_x f_y$ is assumed to be nonsingular for all t, $0 \le t \le t_f$. This system has index-$(m+1)$ (ignoring the delay-terms) and includes the Hessenberg index-2 and an important subset of the higher-index Hessenberg delay-DAEs from Section 1. The delay, or lag, has been normalized to 1, which can be done without loss of generality for any constant (positive) delay. We assume that the functions f and g are sufficiently smooth, and that the initial values for x on $[-1, 0]$ are given such that $x^{(m)}$ exists on $[0, t_f]$.

Standard arguments using Newton's method and the Newton-Kantorovich Theorem apply here as in [2], so we concentrate on the linear (or linearized) case

$$x^{(m)} = \sum_{j=1}^{m} A_j z_j + \sum_{j=1}^{m} D_j z_j(t-1) + By + q \tag{2.3a}$$

$$0 = Cx + r \tag{2.3b}$$

where A_j, B and C are smooth functions of t, $0 \le t \le t_f$, $A_j(t) \in R^{n_x \times n_x}$, $B(t) \in R^{n_x \times n_y}$, $C(t) \in R^{n_y \times n_y}$, $n_y \le n_x$, $D_j(t) \in R^{n_x \times n_x}$ and CB is nonsingular for each t. All of these matrix functions, together with their derivatives up to order m, are assumed to be uniformly bounded in norm by a constant of moderate size. The inhomogeneities $q(t) \in R^{n_x}$ and $r(t) \in R^{n_y}$ are assumed to be m-times differentiable. Above and henceforth, when we omit the argument of a function it is understood to be t (so delay arguments are always specified).

Now, to derive a stability result for this system note that, as in [2], there exists a smooth, bounded matrix function $R(t) \in R^{(n_x - n_y) \times n_x}$ whose linearly independent, normalized rows form a basis for the null space of B^T (R can be taken to be orthonormal). Thus, for each t, $0 \le t \le t_f$,

$$RB = 0 \tag{2.4}$$

We assume that there exists a constant K_1 of moderate size such that

$$||(CB)^{-1}|| \leq K_1 \tag{2.5}$$

uniformly in t, and obtain (Lemma 2.1 in [2]) that there is a constant K_2 of moderate size such that

$$\|(S\ F)\| \equiv \left\| \begin{pmatrix} R \\ C \end{pmatrix}^{-1} \right\| \leq K_2. \tag{2.6}$$

The constant K_2 depends, in addition to K_1, also on $||B||$, $||C||$ and $||R||$. Let K_3 be such that

$$||B^{(j)}||,\ ||C^{(j)}||,\ ||R^{(j)}|| \leq K_3, \qquad j = 0, 1, \ldots, m \tag{2.7}$$

Define new variables

$$u = Rx, \quad 0 \leq t \leq t_f \tag{2.8}$$

Then, using (2.3b), the inverse transformation is given by

$$x = \begin{pmatrix} R \\ C \end{pmatrix}^{-1} \begin{pmatrix} u \\ -r \end{pmatrix} \equiv Su - Fr \tag{2.9}$$

where $S(t) \in \mathcal{R}^{n_x \times (n_x - n_y)}$ satisfies

$$RS = I, \qquad CS = 0 \tag{2.10}$$

and

$$F = B(CB)^{-1}. \tag{2.11}$$

By our assumptions and (2.6) this mapping is well-conditioned. Both S and F are smooth and bounded. The first m derivatives of S and F are bounded by a constant involving K_2 and K_3. Taking m derivatives of (2.8) and multiplying (2.3) by R yields

$$u^{(m)} = (Rx)^{(m)} = \sum_{j=1}^{m} \left[RA_j + \binom{m}{j-1} R^{(m-j+1)} \right] z_j + \sum_{j=1}^{m} RD_j z_j(t-1) + Rq \tag{2.12}$$

Further, using $m-1$ derivatives of (2.9) we obtain the *delay-essential underlying ODE* (DEUODE)

$$\begin{aligned} u^{(m)} &= \sum_{j=1}^{m} \left[RA_j + \binom{m}{j-1} R^{(m-j+1)} \right] \left[(Su)^{(j-1)} - (Fr)^{(j-1)} \right] \\ &+ \sum_{j=1}^{m} RD_j \left[(Su)^{(j-1)}(t-1) - (Fr)^{(j-1)}(t-1) \right] + Rq \end{aligned} \tag{2.13}$$

For a unique solution of (2.3) one needs to impose $m(n_x - n_y)$ initial conditions on u and its derivatives, on the interval $[-1, 0]$. Assuming that B, C and r can be

defined on $[-1,0]$ and that the original initial conditions on z, $z(t) = \beta(t)$ on $[-1,0]$, satisfies the constraint

$$0 = C(t)x(t) + r(t)$$

and its $m-1$ derivatives on $[-1,0]$ (this is a *consistency* requirement on the initial conditions), we can obtain u and its derivatives by differentiating (2.8). If the delay-ODE (2.13) is stable [3] then a similar conclusion holds for the DDAE. We obtain the following theorem:

Theorem 2.1 *Let the DDAE (2.3) have smooth, bounded coefficients, and assume that (2.5) holds and that the underlying problem for (2.13) is stable. Then there is a constant K of moderate size such that* [4]

$$||z|| \leq K \left(||q|| + \sum_{j=0}^{m-1} ||r^{(j)}|| + ||\beta|| \right) \tag{2.14a}$$

$$||y|| \leq K \left(||q|| + \sum_{j=0}^{m} ||r^{(j)}|| + ||\beta|| \right) \tag{2.14b}$$

Proof: The proof is similar to that of Theorem 2.1 in [3], and can therefore be omitted here.

Remark

The DEUODE (2.13) is non-unique. For any nonsingular, smooth, bounded transformation $T(t) \in \mathcal{R}^{(n_x-n_y)\times(n_x-n_y)}$, the transformed $R(t)$ given by

$$R \leftarrow TR \tag{2.15}$$

still satisfies (2.4), (2.6) and (2.7). Hence R is unique only up to such a transformation and, correspondingly, so is the DEUODE. However, a transformation of the variables u in (2.8) corresponding to (2.15) does not alter the boundedness (or lack thereof)

[3] For these purposes, the DODE is said to be stable, or well-conditioned, if the solution can be bounded by a constant of moderate size times the norm of the right-hand side. For ODEs, the means for making this bound is the Green's function. For DODEs, the analogous bound is obtained by representing the solution in terms of an integral of a matrix function which satisfies an adjoint equation, times the right hand side (see e.g. [8], Chapter 10). If the adjoint function can be bounded by a constant of moderate size, then the DODE is well conditioned.

[4] Throughout this paper we use the following notation: Let $|\cdot|$ be the Euclidean vector norm. For a matrix A we denote the induced matrix norm by $||A||$. For a function $u(t)$, $0 \leq t \leq t_f$, we denote the corresponding max function norm by $||u|| := \max\{|u(t)|, 0 \leq t \leq t_f\}$.

of the adjoint function, and hence the stability properties are properly reflected in Theorem 2.1.

Turning to the delay-index-one system

$$x' = f(x, x(t-1), y, y(t-1)) \tag{2.16a}$$
$$0 = g(x, x(t-1), y, y(t-1)), \tag{2.16b}$$

the assumption that g_y is nonsingular allows one to solve the constraint equations (2.16b) for $y(t)$ (using the implicit function theorem), yielding

$$y(t) = \tilde{g}(x(t), x(t-1), y(t-1)) \tag{2.17}$$

If $y(t-1)$ does not appear in (2.16b), and therefore not in (2.17) either, then substituting (2.17) into (2.16a) we obtain the delay-ODE

$$x' = f\left(x, x(t-1), \tilde{g}\left(x, x(t-1)\right), \tilde{g}\left(x(t-1), x(t-2)\right)\right) \tag{2.18}$$

Thus, the DDAE is stable if the DODE (2.18) is stable. Note that if all the delay terms are present in this retarded DODE, then the initial conditions need to be defined for x on $[-2, 0]$.

In the more general case, (2.17) represents a recursion for y. Solving this recursion and substituting into (2.16a) we obtain a delay-ODE which now has the delays $1, 2, \ldots, j$ for $j \leq t < j+1$. Again the DDAE is stable if the DODE is stable, but now no smoothing of boundary discontinuities occurs — the DODE is of a neutral type. The well-conditioning of the DDAE in this case depends on the stability of the recursion (2.17) and on the number of delay-intervals. For additional details, see Appendix A.

Finally, consider the following extension of (2.3) to neutral cases for $m = 1$,

$$x' = Ax + Dx(t-1) + By + q \tag{2.19a}$$
$$0 = Cx + Ex(t-1) + r \tag{2.19b}$$

(see Appendix A). We still have (2.4) - (2.8) holding, but now the inverse transformation (2.9) is replaced by

$$x = Su - FEx(t-1) - Fr \tag{2.20}$$

This is a recursion for x in terms of u, much as (2.17) was a recursion for y in terms of x. (Note that in both of these neutral cases, n_y additional initial interval conditions

are needed: in case of (2.17) on y, and here on all of x.) The obtained DEUODE for u is a delay ODE of neutral type, involving many retarded delays. If it is stable, and if the back-transformation recursion (2.20) is well-conditioned, then Theorem 2.1 still holds.

3 Convergence of numerical methods for retarded and neutral DDAEs

In this section we investigate numerical methods such as BDF and (projected) implicit Runge-Kutta applied to Hessenberg index-1 and index-2 DDAEs of retarded and neutral type.

3.1 BDF

3.1.1 Index-one

Consider the index-1 semi-explicit retarded DDAE,

$$x' = f(x, x(t-1), y, y(t-1)) \tag{3.1a}$$
$$0 = g(x, x(t-1), y) \tag{3.1b}$$

where $\frac{\partial g}{\partial y}$ is nonsingular. We assume that the system (3.1) is well-conditioned. Recall that the underlying DODE is given by (2.18). We wish to discretize (3.1) using a BDF scheme of order k, $1 \leq k \leq 6$, denoted BDF(k). The approximate solution thus obtained at a sequence of mesh points with a maximum step size h is denoted x_h, y_h. For the retarded values of x and perhaps y which may not fall on previous mesh points, we use local interpolants φ^x and φ^y. Assume that these interpolants are of order k_i, i.e., $||\varphi^x v - v|| = O(h^{k_i})$, $||\varphi^y v - v|| = O(h^{k_i})$ for any sufficiently smooth $v(t)$, and use the shorthand φ^x for $\varphi^x x_h$ and φ^y for $\varphi^y y_h$. We suppose that $x \in C^p[0, t_f]$, $y \in C^p[0, t_f]$ and $x^{(p+1)}$ exists and is bounded on $[0, t_f]$.

Theorem 3.1 *Consider the k^{th}-order BDF method applied to the index-1 semi-explicit retarded DDAE (3.1), where $x(t_n - 1), y(t_n - 1)$ are approximated by k_i^{th}-order local interpolants of x_h, y_h satisfying $k_i \geq k$, and using k starting values accurate to $O(h^{\min(p,k)})$. Then this method converges to $O(h^{\min(p,k)})$. Furthermore, if the delayed values of y are computed, instead of by an interpolant through y, by solving the constraint equation (3.1b) for delayed values of y in terms of delayed values of x and its*

interpolant, then the numerical solution of the DDAE coincides with the solution by BDF of the underlying DODE (2.18).

Proof: The solution to (3.1) by BDF(k) satisfies

$$\frac{\rho x_n}{h} = f\left(x_n, \varphi^x(t_n-1), y_n, \varphi^y(t_n-1)\right) \tag{3.2a}$$

$$0 = g\left(x_n, \varphi^x(t_n-1), y_n\right) \tag{3.2b}$$

It is obvious that the approximate solution exists, for h sufficiently small, for both options of dealing with y.

The 'furthermore' part of the theorem is immediate: if $y(t_n-1)$ is approximated, instead of directly by an interpolant, by requiring that the constraint be satisfied,

$$0 = g\left(\varphi^x(t_n-1), \varphi^x(t_n-2), \overline{y}(t_n-1)\right) \tag{3.3}$$

then solving for $\overline{y}(t_n-1)$ in (3.3) and substituting into (3.2), we obtain exactly BDF(k) applied to the underlying DODE (2.18).

If we run a separate interpolant through y, the true solution satisfies

$$\begin{aligned}\frac{\rho x(t_n)}{h} = {} & f\Big(x(t_n), \varphi^x x(t_n-1) + O(h^{\min(p,k)}), \\ & y(t_n), \varphi^y y(t_n-1) + O(h^{\min(p,k)})\Big) + O(h^{\min(p,k)})\end{aligned} \tag{3.4a}$$

$$0 = g\left(x(t_n), \varphi^x x(t_n-1) + O(h^{\min(p,k)}), y(t_n)\right) \tag{3.4b}$$

Subtracting (3.4) from (3.2) and letting $e_n^x = x_n - x(t_n)$, $e_n^y = y_n - y(t_n)$, we obtain

$$\begin{aligned}\frac{\rho e_n^x}{h} = {} & F_0^x e_n^x + F_1^x \varphi^x e^x(t_n-1) + F_0^y e_n^y \\ & + F_1^y \varphi^y e^y(t_n-1) + O(h^{\min(p,k)}) + \eta_1\end{aligned} \tag{3.5a}$$

$$0 = G_0^x e_n^x + G_1^x \varphi^x e^x(t_n-1) + G_0^y e_n^y + O(h^{\min(p,k)}) + \eta_2 \tag{3.5b}$$

where $F_0^x = \frac{\partial f}{\partial x(t)}$, $F_1^x = \frac{\partial f}{\partial x(t-1)}$, $F_0^y = \frac{\partial f}{\partial y(t)}$, $F_1^y = \frac{\partial f}{\partial y(t-1)}$, etc., and η_1, η_2 are higher order terms in e_n^x, e_n^y, etc. Solving in (3.5b) for e_n^y, we obtain

$$e_n^y = -(G_0^y)^{-1}\left(G_0^x e_n^x + G_1^x \varphi^x e^x(t_n-1)\right) + O(h^{\min(p,k)}) + O(\eta_2) \tag{3.6}$$

At the delayed time $t_n - 1$, the interpolant of e^y satisfies

$$\varphi^y e^y(t_n - 1) = \vartheta(e^x(t_n - 1), e^x(t_n - 2)) + O(h^{\min(p,k)}) + O(\eta_2) \tag{3.7}$$

where ϑ is a local approximation operator, accurate to $O(h^{\min(p,k)})$ at least. Note that ϑ first passes an interpolant through values of e^y at mesh points close to $t_n - 1$. For each of these values, the expression (3.6) at previous mesh points is further used.

Substituting (3.6) and (3.7) into (3.5a), we obtain

$$\begin{aligned} \frac{\rho e_n^x}{h} &= F_0^x e_n^x + F_1^x \varphi^x e^x(t_n - 1) \\ &\quad -F_0^y (G_0^y)^{-1} \left(G_0^x e_n^x + G_1^x \varphi^x e^x(t_n - 1)\right) \\ &\quad -F_1^y \vartheta(e^x(t_n - 1), e^x(t_n - 2)) \\ &\quad +O(h^{\min(p,k)}) + O(\eta_1) + O(\eta_2) \end{aligned} \tag{3.8}$$

Thus the method approximates x locally to $O(h^{\min(p+1,k+1)})$. By zero-stability of BDF ($k \leq 6$), (3.8) approximates x globally to $O(h^{\min(p,k)})$. Using (3.6) gives also the desired result for y. □

Remark

The proof applies also to any zero-stable linear multistep method of order k using local k^{th} order interpolants, where the constraints are enforced at every step, using (3.2b).

We can prove a similar result for semi-explicit neutral index-one systems,

$$x' = f(x, x(t-1), y, y(t-1)) \tag{3.9a}$$
$$0 = g(x, x(t-1), y, y(t-1)) \tag{3.9b}$$

We assume that the system (3.9) is well-conditioned.

Corollary 3.1 *Consider the k^{th}-order BDF method applied to the index-1 semi-explicit neutral DDAE (3.9), where $x(t_n - 1), y(t_n - 1)$ are approximated by k_i^{th}-order local interpolants of x_h, y_h satisfying $k_i \geq k$, and using k starting values accurate to $O(h^{\min(p,k)})$. Then this method converges to $O(h^{\min(p,k)})$.*

Proof: The proof follows almost exactly the proof of Theorem 3.1. In place of (3.6) we have a recursion for e_n^y which is stable whenever the DDAE is stable. Solving this recursion for e_n^y and substituting into the equivalent of (3.5a) yields a recursion which is similar to (3.8) except that it involves past values of e^x at all the previous

delay-intervals. Since this recursion is the solution by BDF of the linearized DODE, it is stable and the result follows. □

Remarks

- As in the retarded case, the proof applies also to any zero-stable multistep method of order k using local k^{th} order interpolants, where the constraints are enforced at every step, as in (3.9b).
- We have assumed here that the number of delay-intervals is kept fixed while the number of mesh points grows.

3.1.2 Index-two

We consider retarded Hessenberg index-two DDAEs,

$$x' = f(x, x(t-1), y) \tag{3.10a}$$
$$0 = g(x) \tag{3.10b}$$

where $\frac{\partial g}{\partial x}\frac{\partial f}{\partial y}$ is nonsingular. We discretize this using a BDF(k) scheme with a local interpolant φ for the delay values in x. As before we assume that the interpolant order satisfies $k_i \geq k$, and denote φx_h by φ^x. We assume that $x \in C^p[0, t_f]$ and that $x^{(p+1)}$ exists and is bounded on $[0, t_f]$.

Theorem 3.2 *The BDF(k) method applied to retarded Hessenberg index-2 DAE systems (3.10), with the interpolant φ used to approximate the delayed values of x, converges to $O(h^{\min(p,k)})$.*

Proof: The BDF(k) method applied to (3.10) reads

$$\frac{\rho x_n}{h} = f(x_n, \varphi^x(t_n - 1), y_n) \tag{3.11a}$$
$$0 = g(x_n) \tag{3.11b}$$

Assuming that the initial conditions are consistent and that the starting values for BDF are accurate, the approximate solution clearly exists for $h > 0$ sufficiently small. The true solution satisfies

$$\frac{\rho x(t_n)}{h} = f\left(x(t_n), \varphi x(t_n - 1) + O(h^{\min(p,k)}), y(t_n)\right) + O(h^{\min(p,k)}) \tag{3.12a}$$
$$0 = g(x(t_n)) \tag{3.12b}$$

Subtracting (3.12) from (3.11) and letting $e_n^x = x_n - x(t_n)$, $e_n^y = y_n - y(t_n)$, we find

$$\frac{\rho e_n^x}{h} = F_0^x e_n^x + F_1^x \varphi e^x(t_n - 1) + F_0^y e_n^y + O(h^{\min(p,k)}) + \eta_1 \quad (3.13a)$$

$$0 = G_0^x e_n^x + \eta_2 \quad (3.13b)$$

where η_1, η_2 are higher order terms in e^x, e^y. Analogously to the derivation of the DEUODE, let R_n be such that $R_n F_0^y\ (x(t_n),\ y(t_n)) = 0$. Define $u_n = R_n x_n$, $u(t_n) = R_n x(t_n)$. Thus $e_n^u = R_n e_n^x$. Using (3.13b), we have also $e_n^x = S_n e_n^u + O(\eta_2)$, for $\begin{pmatrix} R \\ G_0 \end{pmatrix}^{-1} = (S\ F)$.

Multiplying (3.13a) by R_n and changing variables to e_n^u, we obtain

$$\begin{aligned} \frac{\rho e_n^u}{h} = {} & R_n A_n S_n e_n^u + (R_n' S_n + O(h)) \sum_{i=1}^{k} i\alpha_i e_{n-i}^u \\ & + (R_n D_n S(t_n - 1) + O(h))\, \varphi e^u(t_n - 1) \\ & + O(h^{\min(p,k)}) + O(\eta_1) + O(\eta_2) \end{aligned} \quad (3.14)$$

Noting that $\sum_{i=1}^{k} i\alpha_i x(t_{n-i}) = x(t_n) + O(h^k)$ because α_i are the BDF(k)-coefficients (in contrast to Theorem 3.1, here we are using the fact that the formula is BDF), we see that (3.14) is a zero-stable k^{th}-order discretization of the DODE

$$(e^u)' = (RAS + R'S)e^u + RDS(t-1)e^u(t-1) + O(h^{\min(p,k)}) \quad (3.15)$$

which is the same as the error equation which is obtained by solving the DEUODE directly by BDF.

We have assumed that this delay-EUODE is well-conditioned and that its values on the initial delay-interval are $O(h^{\min(p,k)})$. Hence its true solution is $O(h^{\min(p,k)})$. Thus, its numerical solution by (3.14) is $O(h^{\min(p,k)})$. Note that for moderate stepsizes, stability depends on the size of the term $R'S$, as in the (non-delay) DAE case [3]. We will see in Section 4 how to formulate the DDAE system so that this term is not large. Finally, nonlinear convergence follows by arguments similar to the BDF analysis in [9]. □

Now, consider the class of neutral index-two systems given by

$$x' = f(x, x(t-1), y) \quad (3.16a)$$

$$0 = g(x, x(t-1)) \quad (3.16b)$$

Corollary 3.2 *The BDF method applied to the class of neutral Hessenberg index-2 DDAE systems (3.16), with the interpolant φ used to approximate the delayed values of x, converges to $O(h^{\min(p,k)})$.*

Proof: The proof follows exactly along the lines of the proof of Theorem 3.2. In place of the back-transformation $e_n^x = S_n e_n^u + O(\eta_2)$ we have instead the recursion corresponding to (2.20), which is solved for x in terms of u. □

3.2 Runge-Kutta methods

3.2.1 Index-one

We are again considering the index-one retarded DDAE

$$\begin{aligned} x' &= f\left(x, x(t-1), y, y(t-1)\right) && \text{(3.17a)} \\ 0 &= g\left(x, x(t-1), y\right) && \text{(3.17b)} \end{aligned}$$

where $\partial g/\partial y$ is nonsingular, and we again assume that $x \in C^p[0, t_f]$ and $x^{(p+1)}$ exists and is bounded on $[0, t_f]$.

Define the s-stage implicit Runge-Kutta method as in [9], applied to (3.17) by

$$\begin{aligned} X_i' &= f\left(X_i, \varphi^x(t_i-1), Y_i, \varphi^y(t_i-1)\right) && \text{(3.18a)} \\ 0 &= g\left(X_i, \varphi^x(t_i-1), Y_i\right) \qquad i = 1, 2, \ldots, s && \text{(3.18b)} \end{aligned}$$

where

$$X_i = x_{n-1} + h \sum_{j=1}^{s} a_{ij} X_j' \tag{3.19}$$

and the interpolants φ^x and φ^y have the properties as described in Section 3.1.1. They are given, for example, by continuous embedded formulas of order k_i (see for instance [15], Section II.5) at the past times, and depend on past intermediate solution and derivative approximations for x and past intermediate solution approximations for y. The numerical solution is advanced by

$$\begin{aligned} x_n &= x_{n-1} + h \sum_{i=1}^{s} b_i X_i' && \text{(3.20a)} \\ 0 &= g\left(x_n, \varphi^x(t_n-1), y_n\right) && \text{(3.20b)} \end{aligned}$$

We note that in the (non-delay) DAE case, these methods are equivalent to solving the underlying ODE directly. Hence, they retain for the DAE all the properties they possess for ODEs such as order, stability, etc.

However, for DODEs the order of these methods often reduces (even when the solution is smooth, say $p \geq k_d$) from the (nonstiff, superconvergence) ODE order $O(h^{k_d})$ to $O(h^{k_s+1})$, where k_s is the stage order of the method, $k_s \leq k_d$. A proof is given in Appendix B (cf. [7, 6]). This is in contrast to BDF schemes, which have no extra accuracy to lose so no order reduction occurs. For instance, in case of a Radau formula, which corresponds to Radau collocation, the order may in general drop from $k_d = 2s - 1$ to $k_s + 1 = s + 1$ (except for the backward Euler case, $s = 1$, for which the order remains $k_d = k_s = 1$).

For the special case of piecewise polynomial collocation (cf. [2]), it is natural to use the same piecewise polynomial solution as the interpolant φ at retarded arguments. This in itself does not improve the loss of accuracy due to order reduction, unless the step sizes are chosen as follows: Assuming that $t_f = J$ is an integer (i.e. there is an integral number of delay intervals), use the same sequence of steps in each delay interval $(j-1, j]$, $j = 1, \ldots, J$, with the last step ending at the point j. (Making the delay interval ends part of the global mesh is a good idea anyway, because there is a possibility for a lower solution discontinuity there.) We call a mesh so constructed π^*.

Theorem 3.3 *Given an s-stage Runge-Kutta method (3.18)-(3.20) applied to the index-1 semi-explicit retarded DDAE (3.17), with a stage order k_s, an ODE order $k_d \geq k_s$ and an interpolation order $k_i \geq k_s$, the following hold:*

1. *The method is convergent and globally accurate to order* $\min(p, k_s + 1, k_d)$.
2. *If the delayed-values of y are computed by solving the constraint equation (3.17b) for y in terms of delayed-values of x and its delayed-approximation, then the numerical solution coincides with the solution by the delay-Runge-Kutta method of the underlying DODE (2.18).*
3. *Furthermore, if a mesh π^* is used, if $x^{(k_d)}$ exists on the subintervals of π^*, and if the delayed x-values are computed using corresponding values of X_i at the appropriate lagged mesh subinterval, then the method converges to order k_d.*

Proof: For the first claim, we outline the proof which is a straightforward extension to the delay case of known DAE results (see for example [2]). Subtract from (3.18) and (3.19) the corresponding expression for $E_i^{x'}$ in terms of e_{n-1}^x. Substitute this into

the error equation corresponding to (3.20), to obtain the recurrence which propagates the error in x. Note that this recurrence agrees with the recurrence which propagates the errors for the delay-RK method applied to the delay-underlying-ODE (2.18), up to terms of order $O(h^{\min(p+1,k_s+2,k_d+1)}) + O(h^{\min(p+1,k_i+1)})$.

The second claim follows directly, as in the BDF case.

The last claim is obtained from a corresponding result for DODEs (see Appendix B). The DODE for x is written as an ODE for the variables $\mathbf{x} = (x_1, x_2, \ldots, x_J)$, where

$$\begin{aligned} x_1(\tau) &= x(\tau) \\ x_2(\tau) &= x(\tau+1) \\ x_3(\tau) &= x(\tau+2) \\ &\vdots \\ x_J(\tau) &= x(\tau+J-1) \end{aligned} \tag{3.21}$$

$0 \le \tau \le 1$. Given the mesh π^*, the application of the Runge-Kutta method to the DODE involves no interpolation and coincides with the same scheme applied to the ODE for (3.21). The convergence results for the ODE are therefore inherited by the method for the DDAE. □

The results extend immediately to the neutral index-1 systems (3.9).

Corollary 3.3 *Given an s-stage Runge-Kutta method (3.18)-(3.20) applied to the index-1 semi-explicit neutral DDAE (3.9), with a stage order k_s, an ODE order $k_d \ge k_s$ and an interpolation order $k_i \ge k_s$, the following hold:*

1. *The method is convergent and globally accurate to order* $\min(p, k_s+1, k_d)$.
2. *If a mesh π^* is used, if $x^{(k_d)}$ exists on the subintervals of π^*, and if the delayed x-values are computed using corresponding values of X_i at the appropriate lagged mesh subinterval, then the method converges to order k_d.*

Proof: The proof follows directly along the lines of the proof of Theorem 3.3, with the extra delay terms handled similarly to the proof of Corollary 3.1. □

3.2.2 Index-two

Here we consider the projected implicit RK methods (PIRK) [2] applied to the retarded index-two Hessenberg DDAE

$$x' = f(x, x(t-1), y) \tag{3.22a}$$

$$0 = g(x) \tag{3.22b}$$

where $\frac{\partial g}{\partial x}\frac{\partial f}{\partial y}$ is nonsingular. As before, we assume that $x \in C^p[0, t_f]$ and $x^{(p+1)}$ exists and is bounded on $[0, t_f]$.

The PIRK method applied to (3.22) is given by

$$X_i' = f(X_i, \varphi^x(t_i - 1), Y_i) \tag{3.23a}$$
$$0 = g(X_i), \qquad i = 1, 2, \ldots, s \tag{3.23b}$$

with the intermediate values X_i defined by

$$X_i = x_{n-1} + h \sum_{j=1}^{s} a_{ij} X_j' \tag{3.24}$$

and the solution advanced by

$$x_n = x_{n-1} + h \sum_{i=1}^{s} b_i X_i' + G(x_n)\lambda_n \tag{3.25a}$$
$$0 = g(x_n) \tag{3.25b}$$

Again, we assume that $\varphi x(t_i - 1)$ is a k_i-th order approximation to $x(t_i - 1)$, usually a continuous embedded formula which makes use of intermediate solution and derivative approximations which were computed near $t_i - 1$, and use the shorthand φ^x for φx_h. Then we have

Theorem 3.4 *Given a well-conditioned, retarded Hessenberg index-2 system (3.22) to be solved by the delay-PIRK method (3.23)-(3.25) where the delay-values in x are approximated locally to $O(h^{k_i})$, $k_i \geq k_s$, then*

1. *The method converges with global order* $\min(p, k_s + 1, k_d)$.
2. *The method is stable, with a moderate stability constant, provided that the DEUODE has a moderate stability constant, and the inverse transformation (2.20) is well-conditioned.*
3. *Suppose the DDAE (3.22) is linear in y, the Runge-Kutta matrix $(a_{ij})_{i,j=1}^{s}$ is invertible, and the method coefficients satisfy the conditions $B(k_d) : \sum_{i=1}^{s} b_i c_i^{k-1} =$*

$\frac{1}{k}$ *for* $k = 1, \ldots, k_d$, $C(q) : \sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k}$ *for* $k = 1, \ldots, q$ *and* $i = 1, \ldots, s$, *and* $D(r) : \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{b_j}{k}(1 - c_j^k)$ *for* $k = 1, \ldots, r$ *and* $j = 1, \ldots, s$. *Let* $k_d \leq 2q + 1$ *and* $k_d \leq q + r + 1$. *If a mesh* π^* *is used, if* $x^{(k_d)}$ *exists on the subintervals of* π^*, *and if the delayed* x*-values are computed using corresponding values of* X_i *at the appropriate lagged mesh subinterval, then the method converges to order* k_d. *(We note that this gives superconvergence order for many methods, including collocation methods).*

Proof: The proof follows exactly along the lines of the proof of Theorem 3.1 in [2]. The delay forms are handled similarly to Theorem 3.2 of this paper. The extra accuracy with the special mesh is shown as in Theorem 3.3, with the DAE order-k_d (superconvergence) results imported from [2] and [19]. □

Remarks

- Note that under the conditions when the construction of the mesh π^* is possible there is an option of transforming the delay-DAE to a boundary value DAE, along the lines of (3.21). The order reduction arising for DODEs can then be avoided. (For a complete set of order conditions for the DAE, see [16].) But the size of the obtained DAE system grows with J and can be very large. This is particularly detrimental for IRK methods, the very ones whose high order is restored by this technique. The special mesh construction described above is often preferable.

- Recall that a DODE, and therefore also a DDAE, can often have discontinuous derivatives at multiples j of the delay, even if the initial data and the functions in the DDAE definition are all smooth. For the DDAEs considered here which lead to retarded DODEs, a discontinuity in $x'(0)$ leads to no worse than a discontinuity in $x^{(j)}(j-1)$ (or in $x^{(j)}(2j-2)$, in case that $x(t-2)$ appears in (2.18)). Still, the global smoothness assumption suggests a possibly very low convergence rate p in Theorems 3.1–3.4.

  Fortunately, the situation can be improved, at least for the Runge-Kutta schemes, if a mesh which includes all of the first p (or $2p$) delay interval ends is used. That is so because if $x \in C[0, t_f]$ then the degree p in the global smoothness assumption may refer to all points *other than* mesh points. (This is clearly true for a mesh π^*, following the argument of conversion to ODE presented in the proof of Theorem 3.3 and in Appendix B. For the more general case, a standard finite-element-type argument is applied.) For a BDF(k) scheme the situation is somewhat more complicated, and a restart may be necessary for each of the first k (or $2k$) delay-intervals. Alternatively for BDF(k), a lower order BDF (with smaller stepsize) may be used until the solution is sufficiently continuous.

The results of Theorem 3.4 can be extended for the class of neutral Hessenberg index-2 systems (3.16). The PIRK method is extended in an obvious way, where (3.23b) and (3.25b) are replaced by $0 = g(X_i, \varphi^x(t_i - 1))$ and $0 = g(x_n, \varphi^x(t_n - 1))$, respectively.

Corollary 3.4 *Given a well-conditioned, neutral Hessenberg index-2 system (3.16) to be solved by the delay-PIRK method (3.23)-(3.25) extended as described above, where the delay-values in x are approximated locally to $O(h^{k_i})$, $k_i \geq k_s$, then*

1. *The method converges with global order* $\min(p, k_s + 1, k_d)$.
2. *The method is stable, with a moderate stability constant, provided that the DEUODE has a moderate stability constant.*
3. *Under the conditions of Theorem 3.4, part 3, if a mesh π^* is used, if $x^{(k_d)}$ exists on the subintervals of π^*, and if the delayed x-values are computed using corresponding values of X_i at the appropriate lagged mesh subinterval, then the method converges to order k_d.*

Proof: The proof follows exactly along the lines of the proof of Theorem 3.4. The back-transformation is handled similarly to Theorem 3.2. □

We close this section with a numerical example.

Example

The following is a nonlinear, semi-explicit DDAE of index at most 2

$$\begin{aligned} x_1' &= (1 + x_2 - \sin t)y + \cos t - (x_2(t-\delta) - \sin(t-\delta))^2 \\ x_2' &= \cos t + x_2(t-\delta) - \sin(t-\delta) \\ x_3' &= y + (x_2(t-\delta) - \sin(t-\delta))^2 \\ 0 &= (x_1 - \sin t)(y - e^t) \end{aligned}$$

where δ is a (positive, possibly time-dependent) given delay. For the initial data

$$x_1(0) = 0, x_2(0) = 0, x_3(0) = 1, \qquad x_2(t) = \sin t, \ (t \leq 0)$$

there are two isolated, smooth solutions.

- One solution is

  $$x_1 = \sin t + (e^t - 1), \ x_2 = \sin t, \ x_3 = e^t, \ y = e^t$$

  The linearized problem about the exact solution has index 1, so this is an instance of (3.17).

- The other solution is

$$x_1 = \sin t,\ x_2 = \sin t,\ x_3 = 1,\ y = 0$$

  The linearized problem about the exact solution has index 2, as in (3.22). But elsewhere, the index may still be 1, so we implemented a program which adaptively decides whether to project as in (3.25) or not ("not" means taking $\lambda_n = 0$ in (3.25a)).

In Table 3.1 we record maximum errors over $0 \le t \le 1$ when running with Gauss-Legendre Runge-Kutta (i.e. collocation at Gaussian points). From Theorems 3.3 and 3.4 we expect the errors to be $O(h^{s+1})$, unless a mesh π^* (and in the index-2 case, a projected method) are used, in which case the order improves to $O(h^{2s})$ errors in x at mesh points.

The notation used in Table 3.1 is as follows: δ denotes the delay: all results are for a uniform step size h chosen so that with $\delta = .2$ we have a mesh π^* whereas with $\delta = .21$ we do not; 'sln' denotes the exact solution being approximated (this depends on the chosen value for approximating $y(0)$); err_x denotes the maximum error over all components of x at mesh points jh, $0 \le j \le 1/h$; erg_x is likewise the "global" error on $[0, 1]$ obtained using the collocation interpolant; erg_y is the "global" error in y (the error in y at mesh points is not significantly different, unless an a-posteriori improvement is used for the index-1 case).

δ	sln	h	s	projected?	err_x	erg_x	erg_y
.2	1	.025	1		.82e-4	.16e-3	.34e-1
.21	1	.025	1		.72e-4	.16e-3	.34e-1
.2	1	.1	3		.78e-11	.87e-8	.28e-5
.21	1	.1	3		.78e-10	.87e-8	.28e-5
.2	2	.025	1	no	.66e-4	.66e-4	.52e-4
.21	2	.025	1	no	.66e-4	.66e-4	.52e-2
.2	2	.025	1	yes	.22e-4	.13e-3	.52e-2
.21	2	.025	1	yes	.22e-4	.13e-3	.52e-2
.2	2	.1	3	no	.55e-8	.55e-8	.67e-6
.21	2	.1	3	no	.55e-8	.55e-8	.67e-6
.2	2	.1	3	yes	.78e-11	.11e-7	.19e-5
.21	2	.1	3	yes	.78e-10	.11e-7	.19e-5

Table 3.1: Maximum solution errors

The results recorded in Table 3.1 tend to confirm the higher order convergence estimates claimed in Part 3 of Theorems 3.3 and 3.4. The error at mesh points is larger for the case $\delta = .21$ (also for $\delta = .2t^2$ which we tried as well), than for $\delta = .2$. For the general case ($\delta = .21$ and other values of δ which we tried) it appears that the estimates in those theorems are somewhat pessimistic for this example. Note,

however, that we have chosen a smooth exact solution: in cases when the solution is less smooth at points $j\delta$ (for a constant δ), it makes a big difference if such points are part of the mesh or not. □

4 Higher-index DDAEs

Although there are convergence theorems for some higher-index non-delay DAE using methods like BDF [9] and certain Runge-Kutta [16], in the context of application to a wide variety of practical problems we cannot in general recommend the use of numerical ODE methods for solving higher-index (index ≥ 3) DAEs directly. This is true even without the introduction of delays. For non-delay DAEs, much recent work has therefore been directed at lower-index formulations of the problem which preserve the stability and which lead to a robust and efficient numerical solution.

In [3], a wide variety of formulations were investigated for the (non-delay) high-index Hessenberg DAEs, and a class of promising formulations called *projected invariants* were proposed. Recall the matrix $R'S$ which appears in the error recurrence for BDF in Theorem 3.2, equation (3.14) (it also appears in the Runge-Kutta error recurrences in the proof of Theorem 3.4). This matrix multiplies explicit terms in the error recurrence. Thus, there can be a problem with numerical stability (i.e. where the stepsize needs to be restricted to maintain stability) if the matrix $R'S$ is large in norm. The projected invariants methods were introduced to overcome that problem by projecting orthogonally onto the constraint manifold to control the size of $R'S$. Essentially, the problem is not only formulated into one of index-2 but also the resulting formulation is nicely conditioned in cases where the ODE is stable on the manifold even if it is not very stable nearby. Here we show how to formulate higher-index, higher-order Hessenberg DDAEs via projected invariants to index-2 systems for which good numerical stability can be attained.

Starting with the index-$(m+1)$, order m retarded DDAE (2.1) which we rewrite here,

$$x^{(m)} = f\left(z(x(t)), z(x(t-1)), y\right) \tag{4.1a}$$

$$0 = g(x) \tag{4.1b}$$

if preservation of the higher-order form (4.1) is not a consideration, the projected invariants form can be obtained by first differentiating the constraint (4.1b) m times. Together with (4.1a), this gives y as a function of $z(x(t))$ and $z(x(t-1))$. Plugging y back into (4.1a) yields the DODE

$$x^{(m)} = \hat{f}\left(z(x(t)), z(x(t-1))\right) \tag{4.2}$$

for which (4.1b) and its first $m-1$ derivatives form an invariant manifold. Now the original constraint can be reintroduced via an additional Langrange multiplier μ. Rewriting (4.2) in first order form leads to a retarded Hessenberg index-2 system

$$\begin{aligned} x_1' &= x_2 + G^T\mu \\ x_2' &= x_3 \\ &\vdots \\ x_m' &= \hat{f}\left(z(x(t)),\ z\left(x(t-1)\right)\right) \\ 0 &= g(x_1) \end{aligned} \tag{4.3}$$

This system is equivalent to (it has the same analytic and also numerical solutions when using compatible discretizations), and is usually written as,

$$\begin{aligned} x_1' &= x_2 + G^T\mu \\ x_2' &= x_3 \\ &\vdots \\ x_m' &= f\left(z(x(t)), z\left(x(t-1)\right), y\right) \\ 0 &= g^{(m)}(x_1, \ldots, x_m) \\ 0 &= g(x_1) \end{aligned} \tag{4.4}$$

Additional derivatives of the original constraint can be enforced similarly, see [3], [4]. If it is important to preserve the higher-order structure, then a trick introduced in [4] can be used to produce such a stable formulation, which is given by

$$\begin{aligned} x^{(m)} &= f\left(x+\phi, x', \ldots, x^{(m-1)}, (x+\phi)(t-1), x', \ldots, x^{(m-1)}(t-1), y\right) \\ \phi' &= -G^T\mu \\ 0 &= g^{(m)}(x+\phi, x', \ldots, x^{(m)}) \\ 0 &= g(x+\phi) \end{aligned} \tag{4.5}$$

with $\phi \equiv 0$ an [-1,0]. This system has the true solution $\phi \equiv 0$, $\mu \equiv 0$.

Stable index-2 formulations for neutral Hessenberg systems which are the higher-index generalizations of the form (3.16) are defined similarly.

Finally, we note that there are a wide variety of formulations which have been proposed for handling high-index DAEs [3], and most extend easily to retarded Hessenberg DDAEs. In particular, any DAE stabilization method which can be viewed as a stabilization of an invariant manifold for an ODE [5] can be immediately extended to a stabilization method of the invariant manifold based on (4.1b) and its derivatives for the DODE (4.2).

Acknowledgements

The authors would like to thank Andy Lumsdaine, Bruno Meyer and Kishore Singhal for pointing out the need to solve delay DAEs and making us aware of applications in circuit analysis and power systems.

References

[1] U. Ascher, *Collocation for two-point boundary value problems revisited*, SINUM, 23 (1986), pp. 596-609.

[2] U. Ascher and L. Petzold, *Projected implicit Runge–Kutta methods for differential–algebraic equations*, SINUM, 28 (1991), pp. 1097-1120.

[3] U. Ascher and L. Petzold, *Stability of computational methods for constrained dynamics systems*, to appear, SISSC.

[4] U. Ascher and L. Petzold, *Projected collocation for higher–order higher–index differential–algebraic equations*, to appear, Appl. Comp. Math.

[5] U. Ascher, H. Qin and S. Reich, *Stabilization of DAE and invariant manifolds*, Tech. Rep. 92-17, Comp. Sci., UBC, Vancouver, 1992.

[6] G. Bader, *Solving boundary value problems for functional differential equations by collocation*, in Numerical Boundary Value ODEs, Ed. U. Ascher and R. Russell, Birkhauser Boston, 1985.

[7] A. Bellen, *One-step collocation for delay differential equations*, J. Comp. Appl. Math 10 (1984), pp. 275-283.

[8] R. Bellman and K.L. Cooke, *Differential-Difference Equations*, Academic Press, New York, 1963.

[9] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Elsevier, 1989.

[10] S.L. Campbell, *Singular linear systems of differential equations with delays*, Applicable Analysis II (1980), pp.129-136.

[11] S.L. Campbell, *2-D (Differential-delay) implicit systems*, Proc. 13th IMACS World Congress on Computation and Applied Mathematics, 1991.

[12] S.L. Campbell, *Nonregular descriptor systems with delays*, to appear, Proceedings of Symposium on Implicit and Nonlinear Systems, Automation & Robotics Research Institute, University of Texas-Arlington, Dec. 14-15, 1992.

[13] C. W. Gear, *Simulation: Conflicts between real-time and software*, Mathematical Software III, Academic Press, 1977.

[14] C.W. Gear and Dianhan Wang, *Real-time integration formulas with off-step inputs and their stability*, University of Illinois Dept. of Computer Science Report No. UIUCDCS-R-86-1277.

[15] E. Hairer, S.P. Norsett and G. Wanner, *Solving Ordinary Differential Equations I, Nonstiff Problems*, Springer-Verlag,New York, 1987.

[16] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems,* Spring, 1992.

[17] J. Hale, *Theory of Functional Differential Equations*, Springer Press, New York, 1977.

[18] H. Heeb and A. Ruehli, *Retarded models for PC board interconnects - or how the speed of light affects your SPICE circuit simulation*, Proc. ICCAD, 1991.

[19] Ch. Lubich, *On projected Runge-Kutta mehtods for differential-algebraic equations*, BIT 31 (1991), pp. 545-550.

[20] K. Singhal, AT&T Bell Laboratories, Allentown, Pennsylvania, personal communication.

[21] A. Skjellum, Lawrence Livermore National Laboratory, Livermore, California, personal communication.

Appendices

A DDAE classification

Campbell [10, 11] notes that retarded-looking DDAEs may in fact be "hiding" DODEs of advanced, or neutral type. Here we consider classification of such DDAEs in Hessenberg form, arriving at the class restrictions utilized in Sections 1 and 2.

Consider first the index-1 system (1.4), reproduced here, for $0 \le t \le J$, with J an integer:

$$x' = f(x, x(t-1), y, y(t-1)) \tag{1a}$$
$$0 = g(x, x(t-1), y), \tag{1b}$$

where $\frac{\partial g}{\partial y}$ is nonsingular. From (1b) we can write, in principle,

$$y(t) = \tilde{g}(x(t), x(t-1)) \tag{2}$$

i.e., $y(t)$ can be expressed as a function of x at t and $t-1$. Applying (2) also for $y(t-1)$ and substituting into (1a), we obtain the retarded DODE

$$x' = f(x, x(t-1), \tilde{g}(x, x(t-1)), \tilde{g}(x(t-1), x(t-2))) \tag{3}$$

which has two delay arguments (these may be more genuinely different if the delay size is a function of t), due to the appearance of $y(t-1)$ in (1a). We may now apply the theory for retarded DODEs to (3) and expect, under certain reasonable conditions, that the initial value DDAE problem for x be well-posed and that a discontinuity in $x'(0)$, say, be propagated into a discontinuity in $x^{(j+1)}(j)$, $0 \le j \le J$, in case of only one delay. (The smoothing is twice as slow in case of two delays.)

Next, consider including dependence on $y(t-1)$ in (1b). For notational simplicity, assume that some linearization has been applied, and consider

$$y = \hat{g}(x, x(t-1)) + Dy(t-1) \tag{4}$$

(i.e. $D = -g_y^{-1} g_{y(t-1)}$). Differentiating (4) we see that, in fact, we have to deal with a DODE of a neutral type for y. Indeed, to remove y, as was done when arriving at (3), we now have to propagate the recursion in (4) back from t to $t-\nu-1$, for ν the integral part of t. This gives

$$y(t) = [\Pi_{j=0}^{\nu} D(t-j)]y(t-\nu-1) + \sum_{l=0}^{\nu} [\Pi_{j=0}^{l-1} D(t-j)]\hat{g}(x(t-l), x(t-l-1)) \tag{5}$$

When this expression is substituted into (1a), the obtained retarded DODE has j delays for $j-1 \le t < j$, and this corresponds to a DODE of a neutral type. In particular, a discontinuity in $x'(0)$ propagates as a discontinuity in $x'(j)$, $0 \le j \le J$, so there is neither smoothing not anti-smoothing by the inverse DODE operator. Note also that initial values must be given here (on an interval in $t \le 0$) both for x and for y. The well-conditioning of the problem depends on the sum in (5). If $\|D\| < 1$ (uniformly) then that sum may not explode even as the number J of delay intervals increases.

Consider now an index-2 DDAE in Hessenberg form. We may view (4) with y replaced by εy, and let $\varepsilon \to 0$. From the limit expression in (5) it is then clear that we must require $D = 0$. Otherwise, a DODE of advanced type is obtained. In order to avoid DODEs of advanced type we therefore restrict ourselves to a DDAE of the form

$$\begin{aligned} x' &= f(x, x(t-1), y, y(t-1)) && \text{(6a)} \\ 0 &= g(x, x(t-1)) && \text{(6b)} \end{aligned}$$

where $\frac{\partial g}{\partial x}\frac{\partial f}{\partial y}$ is nonsingular. Differentiating (6b) and substituting (6a) for $x'(t)$ and $x'(t-1)$, we obtain

$$0 = g_x f(x, x(t-1), y, y(t-1)) + g_{x(t-1)} f(x(t-1), x(t-2), y(t-1), y(t-2)) \quad (7)$$

This allows us to express

$$y(t) = \tilde{g}(x(t), x(t-1), x(t-2), y(t-1), y(t-2))) \quad (8)$$

and propagate this back in time, essentially as in (5). The underlying DODE is of a neutral type. Note that if $y(t-1)$ does not appear in (6a) then $y(t-2)$ does not appear in (8).

To obtain a truely retarded index-2 DDAE in Hessenberg form, we must therefore restrict the form of the DDAE under consideration to

$$\begin{aligned} x' &= f(x, x(t-1), y) && \text{(9a)} \\ 0 &= g(x) && \text{(9b)} \end{aligned}$$

Now a differentiation and substitution as in (7), (8), yields the simpler expression (2), and when this gets substituted into (9a) we obtain a retarded DODE.

B Collocation result for delay-ODEs

Here we prove a convergence result similar to those in Theorems 3.3 and 3.4 for a DODE of retarded or neutral type. The statement of the problem is somewhat more general and the proof is different from those in [7, 6]. We consider an s-stage piecewise polynomial collocation method ($k_s = s$) with ODE order k_d, applied to the linear DODE of retarded or neutral type

$$x' = \sum_{j=0}^{J} A_j x(t-j) + q, \qquad 0 < t < J \tag{1a}$$

$$x(t) = \beta(t) \qquad -1 < t \leq 0 \tag{1b}$$

where the matrices $A_j(t) \in \mathcal{R}^{n_x \times n_x}$ satisfy $A_j(t) \equiv 0$ for $t < j-1$, and $q = q(t)$.

An extension of the analysis to nonlinear problems follows standard lines. Similarly, an extension to non-collocation Runge-Kutta methods is possible (cf. [16]). An extension to boundary value DODEs also follows immediately from the arguments below. For simplicity of exposition, we assume a uniform step size $h = J/N$, although a mesh π with no relative stepsize restrictions whatsoever may be used (cf. [1]). If there is an integer μ such that $h\mu = 1$ then the mesh includes the delay interval ends, and is denoted π^*. We also assume for now that the problem (1) has a sufficiently smooth solution, because the modification of our results for a lower smoothness is standard. Let $t_{nj} = t_{n-1} + hc_j$, $1 \leq j \leq s$, be the collocation points in $[t_{n-1}, t_n]$ ($c_j = \sum_{l=1}^{s} a_{jl}$, to recall). The collocation solution $x_\pi(t)$ is a continuous function on $[-1, J]$ which reduces on each element $[t_{n-1}, t_n]$ to a polynomial of degree at most s, and satisfies (the initial conditions and) (1) at the collocation points. Therefore, the error

$$e(t) = x_\pi(t) - x(t) \tag{2}$$

satisfies homogeneous initial conditions and

$$e' = \sum_{j=0}^{J} A_j e(t-j) + d, \qquad 0 < t < J \tag{3a}$$

$$d(t_{nj}) = 0, \qquad 1 \leq j \leq s, 1 \leq n \leq N \tag{3b}$$

The assumption of well-conditioning of (1) implies, for h sufficiently small, stability of the collocation approximation and the basic error estimate

$$\|e\|_{L_\infty[0,J]} = O(h^s)$$

Using this in (3) then yields at collocation points $e'(t_{nj}) = O(h^s)$, and since x_π is a piecewise polynomial of degree $< s$, we have at all points other than mesh points (cf. [1]),

$$\|e^{(j)}\|_{L_\infty[0,J]} = O(h^{s+1-j}), \qquad 1 \leq j \leq s \tag{4}$$

The quest is now to obtain a sharper estimate on $e(t)$.

Consider the conversion of (1) to an ODE system for $\mathbf{x} = (x_1, x_2, \ldots, x_J)$, where

$$\begin{aligned} x_1(\tau) &= x(\tau) \\ x_2(\tau) &= x(\tau+1) \\ x_3(\tau) &= x(\tau+2) \\ &\vdots \\ x_J(\tau) &= x(\tau+J-1) \end{aligned} \tag{5}$$

$0 \le \tau \le 1$. Similarly, let $q_j(\tau) = q(\tau+j-1)$ and $A_{lj} = A_l(\tau+j-1)$, $j = 1, \ldots, J$. We have from (1), for $1 \le j \le J$,

$$x_j' = \sum_{l=0}^{j-1} A_{lj} x_{j-l} + q_j + A_{jj}\beta(\tau-1), \qquad 0 < \tau < 1 \tag{6}$$

This is an ODE system of size Jn_x, for which we have the boundary conditions $x_j(0) = x_{j-1}(1)$, $j = 2, \ldots, J$ and $x_1(0) = \beta(0)$. We obtain a well-conditioned boundary value ODE according to our assumptions, and thus there exists a nicely bounded Green's function $G(\tau, \sigma) \in \mathcal{R}^{Jn_x \times Jn_x}$. If we now define $\mathbf{x}_\pi$, $\mathbf{d}$ and $\mathbf{e}$ as relating to the collocation solution x_π and the errors d and e, respectively, in precisely the same way as $\mathbf{x}$ relates to x, we obtain

$$\mathbf{e}(\tau) = \int_0^1 G(\tau, \sigma)\mathbf{d}(\sigma)d\sigma, \qquad 0 \le \tau \le 1 \tag{7}$$

Now, if the mesh has the special structure π^* then $\mathbf{d}(\tau_{nj}) = 0$, where $\tau_{nj} = t_{nj}$, $n = 1, \ldots, \mu$, are the collocation points in τ. In this case the ODE collocation theory immediately applies, and we obtain (cf. [1])

$$\|e\|_{L_\infty[0,J]} = O(h^{\min(s+1, k_d)}) \tag{8a}$$

$$|e(t_n)| = O(h^{k_d}), \qquad 0 \le n \le N \tag{8b}$$

For a general mesh, let us write

$$G = (\, G_1 \quad \ldots G_J \,)$$

where each G_j is a block of n_x columns of the Green's function G. We can then write (7) as

$$\mathbf{e}(\tau) = \int_0^1 \sum_{j=1}^{J} G_j(\tau, \sigma) d_j(\sigma) d\sigma \tag{9}$$

For each j in this expression (9) we now write the integral as a sum of its components according to the mesh,

$$\int_0^1 G_j(\tau,\sigma)d_j(\sigma)d\sigma = \left\{\int_{j-1}^{t_1^j} + \int_{t_1^j}^{t_2^j} + \ldots + \int_{t_{N_j}^j}^{j}\right\} G_j(\tau,\rho-j+1)d(\rho)d\rho \qquad (10)$$

where $t_1^j, t_2^j, \ldots, t_{N_j}^j$ are part of the given mesh π in the $j-th$ delay interval.

The integrals in (10) are of two types. The first group includes possibly the first, possibly the last, and the interval where $\tau - j + 1$ is. For each of these we cannot use orthogonality (the first and last integrals are not over a full mesh element, and where $\tau - j + 1$ is there is no smoothness in G_j), so each contributes an error $O(h^{s+1})$. For the other integrals in (10), we have that $d(t_{nl}) = 0$, $l = 1, \ldots, s$, and we may use orthogonality. Thus, these integrals are each $O(h^{k_d+1})$, as in the ODE collocation theory. Summing up, we have

$$\int_0^1 G_j(\tau,\sigma)d_j(\sigma)d\sigma = O(h^{\min(s+1,k_d)})$$

Substituting this in (9), we obtain

$$\|e\|_{L_\infty[0,J]} = O(h^{\min(s+1,k_d)}) \qquad (11)$$

Thus, the general order of convergence for the ODE case is recovered, as in (8a), even for a general mesh, but the superconvergence result (8b) is not.

Note that we have assumed here that the number of delay intervals J is kept fixed while the number of mesh points grows. In case that $J \geq N$, say, a different, additional analysis is needed. The first question is then how G depends on J, but we do not pursue this question further here.