

## THE NUMERICAL SOLUTION OF DELAY-DIFFERENTIAL-ALGEBRAIC EQUATIONS OF RETARDED AND NEUTRAL TYPE\*

URI M. ASCHER<sup>†</sup> AND LINDA R. PETZOLD<sup>‡</sup>

**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-ordinary differential equation (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 as stable a way as index-2 Hessenberg DDAEs. We also comment on some practical aspects of the numerical solution of these problems.

**Key words.** differential-algebraic equations, delays, higher index

**AMS subject classification.** 65L05

**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, and optimal control. 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 to, for example, interconnects for computer chips [19] and transmission lines [23], and in chemical process simulation when modeling pipe flow [25]. Although there is extensive literature on the mathematical structure of delay-ordinary differential equations (DODEs) (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]–[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 also widely regarded 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

---

\* Received by the editors December 22, 1992; accepted for publication (in revised form) March 9, 1994.

<sup>†</sup> Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z4 (ascher@cs.ubc.ca). The work of this author was partially supported by Natural Sciences and Engineering Research Council of Canada grant OGP 0004306.

<sup>‡</sup> Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455 (petzold@cs.umn.edu). The work of this author was partially supported by Army Research Office, contract DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center, and by Army Research Office contract DAAL03-92-G-0247, Department of Energy contract DE-FG02-92ER25130 and National Institute of Standards and Technology contract 60NANB2D1272.

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<sup>1</sup> [9]. Still, even in this restricted form, DAEs of index  $\geq 2$  present many challenges to designers of numerical methods [9]. The index-1 semi-explicit DAE is given by

$$(1.1a) \quad x' = f(x, y),$$

$$(1.1b) \quad 0 = g(x, y),$$

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

The index-2 Hessenberg DAE is given by

$$(1.2a) \quad x' = f(x, y),$$

$$(1.2b) \quad 0 = g(x),$$

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

The Hessenberg index-3 DAE is given by

$$(1.3a) \quad y' = f(x, y, z),$$

$$(1.3b) \quad x' = g(x, y),$$

$$(1.3c) \quad 0 = h(x),$$

where  $\frac{\partial h}{\partial x} \frac{\partial g}{\partial y} \frac{\partial f}{\partial z}$  is nonsingular.<sup>2</sup>

Semi-explicit index-1 systems arise in a wide variety of applications including most circuit analysis and power systems problems. Some examples of Hessenberg index-2 systems are modeling of incompressible fluids (following spatial discretization) and some index-2 formulations of mechanical systems [4]. 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 DDAEs of index 1 and 2 in pure (Hessenberg) form.

A great deal is known about the structure of DODEs [8], [17]. These systems are classified by their *type*. For a scalar 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.

<sup>1</sup> 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.

<sup>2</sup> These nonsingularity conditions ensure, using the implicit function theorem, that the algebraic variables can be eliminated after  $m$  constraint differentiations for the index- $(m+1)$  DAE.

In this paper we study DDAEs of retarded type which are extensions of Hessenberg form. These DDAE systems are given by<sup>3</sup>

$$(1.4a) \quad x' = f(x, x(t - 1), y, y(t - 1)),$$

$$(1.4b) \quad 0 = g(x, x(t - 1), y)$$

(where  $\frac{\partial g}{\partial y}$  is nonsingular) for index-1,

$$(1.5a) \quad x' = f(x, x(t - 1), y),$$

$$(1.5b) \quad 0 = g(x)$$

(where  $\frac{\partial g}{\partial x} \frac{\partial f}{\partial y}$  is nonsingular) for index-2 and

$$(1.6a) \quad y' = f(x, x(t - 1), y, y(t - 1), z),$$

$$(1.6b) \quad x' = f(x, x(t - 1), y),$$

$$(1.6c) \quad 0 = h(x)$$

(where  $\frac{\partial h}{\partial x} \frac{\partial g}{\partial y} \frac{\partial f}{\partial z}$  is nonsingular) for index-3. 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 some DDAEs which “look like” they should be of retarded type but are actually neutral or advanced type, see Campbell [11].

We also 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.

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 §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 §3 we investigate the convergence and order for numerical methods such as backward differentiation formulae (BDF) and projected implicit Runge–Kutta (PIRK) method [3] applied to index-1 and index-2 retarded and neutral Hessenberg DDAEs. We also comment on some practical aspects of the numerical solution of these problems. In §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 DDAEs.** In this section we first consider the DDAE of order  $m$

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

$$(2.1b) \quad 0 = g(x),$$

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

$$(2.2a) \quad z_j(t) = x^{(j-1)}(t) = \frac{d^{j-1}x(t)}{dt^{j-1}}, \quad (1 \leq j \leq m),$$

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

---

<sup>3</sup> We use an autonomous form for the nonlinear systems considered, without loss of generality, simply to keep the notation concise.

and  $g_x f_y$  is assumed to be nonsingular for all  $t$ ,  $0 \leq t \leq t_f$ . 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 DDAEs from §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 [3], so we concentrate on the linear (or linearized) case

$$(2.3a) \quad x^{(m)} = \sum_{j=1}^m A_j z_j + \sum_{j=1}^m D_j z_j(t - 1) + B y + q,$$

$$(2.3b) \quad 0 = C x + r,$$

where  $A_j, B$ , and  $C$  are smooth functions of  $t$ ,  $0 \leq t \leq t_f$ ,  $A_j(t) \in R^{n_x \times n_x}$ ,  $B(t) \in R^{n_x \times n_y}$ ,  $C(t) \in R^{n_y \times n_y}$ ,  $n_y \leq n_x$ ,  $D_j(t) \in R^{n_x \times n_x}$ , 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 (see, e.g., [2]). 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 [3], 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  $t$ ,  $0 \leq t \leq t_f$ ,

$$(2.4) \quad RB = 0.$$

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

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

uniformly in  $t$ , and obtain (Lemma 2.1 in [3]) that there is a constant  $K_2$  of moderate size such that

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

In addition to  $K_1$ , the constant  $K_2$  depends on  $\|B\|$ ,  $\|C\|$ , and  $\|R\|$ . Let  $K_3$  be such that

$$(2.7) \quad \|B^{(j)}\|, \|C^{(j)}\|, \|R^{(j)}\| \leq K_3, \quad j = 0, 1, \dots, m.$$

Define new variables

$$(2.8) \quad u = R x, \quad 0 \leq t \leq t_f.$$

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

$$(2.9) \quad x = \begin{pmatrix} R \\ C \end{pmatrix}^{-1} \begin{pmatrix} u \\ -r \end{pmatrix} \equiv S u - F r,$$

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

$$(2.10) \quad RS = I, \quad CS = 0,$$

and

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

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

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

Further, using  $m - 1$  derivatives of (2.9) we obtain the DEUODE

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

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  $[-1, 0]$  satisfy 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 DODE (2.13) is stable<sup>4</sup> 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<sup>5</sup>*

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

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

<sup>4</sup> 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 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, Chap. 10]). If the adjoint function can be bounded by a constant of moderate size, then the DODE is well conditioned.

<sup>5</sup> 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  $\|u\| := \max\{|u(t)|, 0 \leq t \leq t_f\}$ .

*Proof.* The proof is similar to that of Theorem 2.1 in [4], and can therefore be omitted here.  $\square$

*Remark.* The DEUODE (2.13) is nonunique. 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

$$(2.15) \quad R \leftarrow TR$$

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 variable  $u$  in (2.8) corresponding to (2.15) does not alter the boundedness (or lack thereof) of the adjoint function, and hence the stability properties are properly reflected in Theorem 2.1.

Turning to the delay-index-1 system

$$(2.16a) \quad x' = f(x, x(t-1), y, y(t-1)),$$

$$(2.16b) \quad 0 = g(x, x(t-1), y, y(t-1)),$$

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

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

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

$$(2.18) \quad x' = f(x, x(t-1), \tilde{g}(x, x(t-1)), \tilde{g}(x(t-1), x(t-2))).$$

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 DODE which now has the delays  $1, 2, \dots, j$  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 the neutral cases for  $m = 1$ :

$$(2.19a) \quad x' = Ax + Dx(t-1) + By + q,$$

$$(2.19b) \quad 0 = Cx + Ex(t-1) + r$$

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

$$(2.20) \quad x = Su - FEx(t-1) - Fr.$$

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 DODE 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 PIRK applied to Hessenberg index-1 and index-2 DDAEs of retarded and neutral type.

**3.1. BDF.**

**3.1.1. Index-1.** Consider the index-1 semi-explicit retarded DDAE

$$(3.1a) \quad x' = f(x, x(t-1), y, y(t-1)),$$

$$(3.1b) \quad 0 = g(x, x(t-1), y),$$

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  $\varphi^x$  and  $\varphi^y$ . 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  $v(t)$ . 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 by solving the constraint equation (3.1b) for delayed values of  $y$  in terms of delayed values of  $x$  and its interpolant, instead of by an interpolant through  $y$ , 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

$$(3.2a) \quad \frac{\rho x_n}{h} = f(x_n, \varphi^x x_h(t_n-1), y_n, \varphi^y y_h(t_n-1)),$$

$$(3.2b) \quad 0 = g(x_n, \varphi^x x_h(t_n-1), y_n).$$

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

$$(3.3) \quad 0 = g(\varphi^x x_h(t_n-1), \varphi^x x_h(t_n-2), \bar{y}(t_n-1))$$

be satisfied, then solving for  $\bar{y}(t_n-1)$  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

$$(3.4a) \quad \frac{\rho x(t_n)}{h} = f\left(x(t_n), \varphi^x x(t_n-1) + O(h^{\min(p,k)}), y(t_n), \varphi^y y(t_n-1) + O(h^{\min(p,k)})\right) + O(h^{\min(p,k)}),$$

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

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

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

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

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  $\eta_1, \eta_2$  are higher-order terms in  $e_n^x, e_n^y$ , etc. Solving in (3.5b) for  $e_n^y$ , we obtain

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

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

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

where  $\vartheta$  is a local approximation operator, accurate to  $O(h^{\min(p,k)})$  at least. Note that  $\vartheta$  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 used.

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

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

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) also gives the desired result for  $y$ .  $\square$

*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-1 systems,

$$(3.9a) \quad x' = f(x, x(t-1), y, y(t-1)),$$

$$(3.9b) \quad 0 = g(x, x(t-1), y, y(t-1)).$$

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.  $\square$

*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-2.** We consider retarded Hessenberg index-2 DDAEs,

$$(3.10a) \quad x' = f(x, x(t-1), y),$$

$$(3.10b) \quad 0 = g(x),$$

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  $\varphi$  for the delay values in  $x$ . As before we assume that the interpolant order satisfies  $k_i \geq k$ . 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  $\varphi$  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

$$(3.11a) \quad \frac{\rho x_n}{h} = f(x_n, \varphi x_h(t_n-1), y_n),$$

$$(3.11b) \quad 0 = g(x_n).$$

Assuming that the initial conditions are consistent [21] and that the starting values for BDF are accurate, the approximate solution clearly exists for  $h > 0$  sufficiently small. The true solution satisfies

$$(3.12a) \quad \frac{\rho x(t_n)}{h} = f(x(t_n), \varphi x(t_n-1) + O(h^{\min(p,k)}), y(t_n)) + O(h^{\min(p,k)}),$$

$$(3.12b) \quad 0 = g(x(t_n)).$$

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

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

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

where  $\eta_1, \eta_2$  are higher-order terms in  $e^x, e^y$ . Analogous 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

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

Noting that  $\sum_{i=1}^k (-i) \alpha_i x(t_{n-i}) = x(t_n) + O(h^k)$  because  $\alpha_i$  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

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

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

We have assumed that this DEUODE 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 (nondelay) DAE case [4]. We will see in §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].  $\square$

Now, consider the class of neutral index-2 systems given by

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

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

**COROLLARY 3.2.** *The BDF method applied to the class of neutral Hessenberg index-2 DDAE systems (3.16), with the interpolant  $\varphi$  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 instead have the recursion corresponding to (2.20), which is solved for  $x$  in terms of  $u$ .  $\square$

### 3.2. Runge–Kutta methods.

**3.2.1. Index-1.** We are again considering the index-1 retarded DDAE

$$(3.17a) \quad x' = f(x, x(t-1), y, y(t-1)),$$

$$(3.17b) \quad 0 = g(x, x(t-1), y),$$

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

$$(3.18a) \quad X'_i = f(X_i, \varphi^x x_h(t_i-1), Y_i, \varphi^y y_h(t_i-1)),$$

$$(3.18b) \quad 0 = g(X_i, \varphi^x x_h(t_i-1), Y_i), \quad i = 1, 2, \dots, s,$$

where

$$(3.19) \quad X_i = x_{n-1} + h \sum_{j=1}^s a_{ij} X'_j$$

and the interpolants  $\varphi^x$  and  $\varphi^y$  have the properties as described in §3.1.1. They are given, for example, by continuous embedded formulas of order  $k_i$  (see for instance [15, §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

$$(3.20a) \quad x_n = x_{n-1} + h \sum_{i=1}^s b_i X'_i,$$

$$(3.20b) \quad 0 = g(x_n, \varphi^x x_h(t_n-1), y_n).$$

We note that in the (nondelay) 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 and thus no order reduction occurs. For instance, in case of a Radau formula, which corresponds to Radau collocation, in general the order may 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. [3]), it is natural to use the same piecewise polynomial solution as the interpolant  $\varphi$  at retarded arguments. This in itself does not improve the loss of accuracy due to order reduction, unless the stepsizes are chosen as follows. Assume that  $t_f = J$  is an integer (i.e., there is an integral number of delay intervals) and use the same sequence of steps in each delay interval  $(j - 1, j]$ ,  $j = 1, \dots, J$ , 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  $\pi^*$ .

**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  $\pi^*$  is used, if  $x^{(k_d)}$  exists on the subintervals of  $\pi^*$ , 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 [3]). 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-Runge-Kutta 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 directly from a corresponding result for DODEs. For the latter see Theorem 15.1 of [15] or Appendix B.  $\square$

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  $k_d \geq k_s$ , 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 a mesh  $\pi^*$  is used, if  $x^{(k_d)}$  exists on the subintervals of  $\pi^*$ , 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.  $\square$

**3.2.2. Index-2.** Here we consider the projected implicit RK methods (PIRK) methods [3] applied to the retarded index-2 Hessenberg DDAE

$$(3.21a) \quad x' = f(x, x(t - 1), y),$$

$$(3.21b) \quad 0 = g(x),$$

where  $\frac{\partial g}{\partial x} \frac{\partial f}{\partial y}$  is nonsingular. Denote  $G = \frac{\partial g}{\partial x}$ . 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 (3.21) is given by

$$(3.22a) \quad X'_i = f(X_i, \varphi x_h(t_i - 1), Y_i),$$

$$(3.22b) \quad 0 = g(X_i), \quad i = 1, 2, \dots, s,$$

with the intermediate values  $X_i$  defined by

$$(3.23) \quad X_i = x_{n-1} + h \sum_{j=1}^s a_{ij} X'_j$$

and the solution advanced by

$$(3.24a) \quad x_n = x_{n-1} + h \sum_{i=1}^s b_i X'_i + G(x_n) \lambda_n,$$

$$(3.24b) \quad 0 = g(x_n)$$

(i.e.,  $\lambda_n$  in (3.24a) is chosen such that the resulting  $x_n$  satisfies the constraint (3.24b) at the right end of the integration step). 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  $t_i - 1$ . Then we have the following theorem.

**THEOREM 3.4.** *Given a well-conditioned, retarded Hessenberg index-2 system (3.21) to be solved by the delay-PIRK method (3.22)–(3.24) 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 (3.21) 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  $k = 1, \dots, k_d$ ,  $C(q) : \sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{c_i^k}{k}$  for  $k = 1, \dots, q$  and  $i = 1, \dots, s$ , and  $D(r) : \sum_{i=1}^s b_i c_i^{k-1} a_{ij} = \frac{b_j}{k} (1 - c_j^k)$  for  $k = 1, \dots, r$  and  $j = 1, \dots, s$ . Let  $k_d \leq 2q + 1$  and  $k_d \leq q + r + 1$ . If a mesh  $\pi^*$  is used, if  $x^{(k_d)}$  exists on the subintervals of  $\pi^*$ , 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 [3]. 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 [3] and [22].  $\square$

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 (3.22b) and (3.24b) are replaced by  $0 = g(X_i, \varphi x_h(t_i - 1))$  and  $0 = g(x_n, \varphi x_h(t_n - 1))$ , respectively.

**COROLLARY 3.4.** *Given a well-conditioned, neutral Hessenberg index-2 system (3.16) to be solved by the delay-PIRK method (3.22)–(3.24) 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  $\pi^*$  is used, if  $x^{(k_d)}$  exists on the subintervals of  $\pi^*$ , 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. □

**3.3. Further remarks and an example.** Below we make further remarks on various aspects of the numerical solution of DDAEs.

- The problem of numerically solving DDAEs inherits, of course, the difficulties and the usual considerations associated with the numerical solution of DODEs and DAEs. Thus note, for instance, that our BDF results are proved for a constant stepsize and that the usual considerations must be made when contemplating a variable stepsize implementation. Also, the choice of consistent initial conditions is not always straightforward (see, e.g., [21]). Note also that no projection is needed for Runge–Kutta methods which satisfy (3.24b) anyway (because this requirement is part of (3.22b)), so Theorem 3.4 and Corollary 3.4 hold, for instance, for collocation at Radau points.
- Note that under the conditions when the construction of the mesh  $\pi^*$  is possible there is an option of transforming the DDAE into a boundary value DAE, along the lines of (5) in Appendix B. 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 implicit Runge–Kutta (IRK) methods, the very ones whose high order is restored by this technique. The special mesh construction described above is often preferable. Still, the global nature of the mesh  $\pi^*$  makes it of a limited practical value.
- 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 that 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  $x(t - 2)$  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  $\pi^*$ , following the argument of conversion to an ODE presented 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.

In the case of a DDAE of a neutral type, in general all delay interval ends within the integration interval  $[0, t_f]$  must be part of the mesh in order to obtain high-order accuracy.

- We have not developed a theory for delays of variable size in the present work. However, the numerical methods we use can certainly be applied to such problems. The task of locating delay interval ends becomes nontrivial, but it is often possible and worthwhile to find these points and make sure that sufficiently many of them become part of the discretization mesh.
- R. Hauber [18] has pointed out to us the following. Given the added difficulty with using multistep methods in the DODE context, it is particularly desirable to find ways to restore the full superconvergence order of IRK methods on a general mesh containing the delay interval ends. From Appendix B this would be possible if a way is found to obtain the lagged solution values accurate to order  $k_d$ . This in turn is possible if one uses for a given lagged argument a sufficiently accurate Lagrange interpolation of approximate solution values at nearby mesh points, provided that the smallest interval containing all of these mesh points does not contain any low-continuity delay interval end. The latter condition is possible to arrange with special mesh restrictions and nonsymmetric interpolations, and then the superconvergence order can be recovered, but it does appear that the simplicity and elegance of the one-step method is somewhat lost with this process.
- Consider the initial value problem for the general DDAE

$$(3.25a) \quad x' = f(x, x(t-1), y, y(t-1)),$$

$$(3.25b) \quad 0 = g(x, x(t-1), y, y(t-1)),$$

where we assume  $t_f = J$  for notational simplicity. The usual approach to integrating this problem numerically is to sequentially integrate over delay intervals; i.e., for  $j = 0, 1, \dots, J-1$ , having integrated up to  $t = j$ , proceed to integrate the DAE over  $[j, j+1]$  with the lagged values known. This can be written as a functional iteration: with given functions  $x^0(t), y^0(t)$  which agree with the exact solution  $x(t), y(t)$  for  $t \leq 0$ , solve iteratively

$$(3.26a) \quad x'^{j+1} = f(x^{j+1}, x^j(t-1), y^{j+1}, y^j(t-1)),$$

$$(3.26b) \quad 0 = g(x^{j+1}, x^j(t-1), y^{j+1}, y^j(t-1)),$$

for  $j \leq t \leq j+1, j = 0, \dots, J-1$ . Next we can extend the definition of  $x^{j+1}(t)$  to the entire integration interval  $[-1, t_f]$  by letting  $x^{j+1}(t) = x^j(t), t \leq j$ ,  $x^{j+1}(t)$  arbitrary for  $t > j+1$ , and obtain a waveform method which yields the exact solution in  $J$  iterations.

One may now consider applying a waveform-type approximation to the DAE (3.26), see, e.g., [24]. For instance, if (3.26b) can be written as

$$y^{j+1} = \tilde{g}(x^{j+1}, x^j(t-1), y^j(t-1)),$$

then this can be substituted into (3.26a), yielding

$$(3.27) \quad x'^{j+1} = \tilde{f}(x^{j+1}, x^j(t-1), y^j(t-1)),$$

(for a higher-index DAE a process of constraint differentiation, elimination of  $y^{j+1}$ , and stabilization, e.g., as in [1], yields a similar expression), and a modified Picard iteration can be applied to (3.27):

$$(3.28) \quad x^{l+1} = Ax^{l+1} + \tilde{f}(x^l, x^l(t-1), y^l(t-1)) - Ax^l,$$

$l = 0, 1, \dots$ , where  $A$  is a suitable negative definite matrix. Choosing  $A$  to be diagonal or triangular yields a waveform relaxation method and a possibility for parallelism across the method [20], [24]. The integration range in (3.28) is  $t_l \leq t \leq \tau_l$ , where  $t_l$  is chosen such that  $x^l$  is already sufficiently accurate for  $t \leq t_l$ , and  $\tau_l = t_l + 1$  or  $\tau_l = J$ . Convergence is now guaranteed under mild conditions using a standard analysis.

Finally, consider replacing (3.26) by a quasilinearized form

$$(3.29a) \quad \begin{aligned} x^{l+1} &= f(x^l, x^l(t-1), y^l, y^l(t-1)) + A_l(x^{l+1} - x^l) + \\ & \quad B_l(y^{l+1} - y^l), \end{aligned}$$

$$(3.29b) \quad \begin{aligned} 0 &= g(x^l, x^l(t-1), y^l, y^l(t-1)) + C_l(x^{l+1} - x^l) + \\ & \quad D_l(y^{l+1} - y^l), \end{aligned}$$

for  $0 \leq t \leq \tau_l$ ,  $l = 0, 1, \dots$ , with  $A_l(t) = \frac{\partial f}{\partial x}(x^l, x^l(t-1), y^l, y^l(t-1))$ , etc. (i.e., the quasilinearization is performed only with respect to the nonlagged arguments at  $t$ ). This yields a linear DAE to be discretized and solved for each  $l$ . Local quadratic convergence is obtained using the usual Newton–Kantorovich theorem for each delay interval  $[j, j + 1]$  in turn. Moreover, our experience indicates that often when the delay interval is small ( $J$  large) and  $\tau_l = t_f$  then much fewer than  $J$  iterations (3.29) are needed for convergence.

We close this section with a numerical example.

*Example.* The following is a nonlinear, semi-explicit DDAE of index at most 2:

$$\begin{aligned} x'_1 &= (1 + x_2 - \sin t)y + \cos t + \sin t - (x_2(t - \delta) - \sin(t - \delta))^2, \\ x'_2 &= \cos t + x_2(t - \delta) - \sin(t - \delta), \\ x'_3 &= y + (x_2(t - \delta) - \sin(t - \delta))^2, \\ 0 &= (x_1 - \sin t)(y - e^t), \end{aligned}$$

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

$$(3.30) \quad \begin{aligned} x_1(0) &= 0, \quad x_2(0) = 0, \quad x_3(0) = 1, \\ x_2(t) &= \sin t, \quad (t \leq 0) \end{aligned}$$

there are two isolated, smooth solutions:

- One solution is

$$(3.31) \quad x_1 = \sin t - \cos t + e^t, \quad x_2 = \sin t, \quad x_3 = e^t, \quad y = e^t.$$

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

- The other solution is

$$(3.32) \quad x_1 = \sin t, \quad x_2 = \sin t, \quad x_3 = \cos t, \quad y = -\sin t.$$

The linearized problem about the exact solution has index 2, as in (3.21). But elsewhere the index may still be 1, so we wrote a program based on Gauss-Legendre Runge-Kutta (i.e., collocation at Gaussian points), which adaptively decides whether to project as in (3.24) or not ("not" means taking  $\lambda_n = 0$  in (3.24a)).

In Table 3.1 we record maximum errors over  $0 \leq t \leq 2$ . From Theorems 3.3 and 3.4 we expect the errors to be  $O(h^{s+1})$ , unless a mesh  $\pi^*$  (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:  $\delta$  denotes the delay, i.e., all results are for a uniform step size  $h$  chosen so that with  $\delta = .25$  we have a mesh  $\pi^*$  whereas with  $\delta = .26$  we do not; "sln" denotes the exact solution being approximated, i.e., sln = 1 for (3.31), sln = 2 for (3.32) (so the linearized problem has index  $i$  for sln =  $i$ ,  $i = 1, 2$ ; which solution the method converges to depends on the chosen value for approximating  $y(0)$ );  $\text{err}_x$  denotes the maximum error over  $x_1$  in case that sln = 1 and over  $x_3$  in case that sln = 2 at mesh points  $jh$ ,  $0 \leq j \leq 1/h$ ;  $\text{erg}_x$  is likewise the "global" error on  $[0, 2]$  obtained using the collocation interpolant;  $\text{erg}_y$  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).

TABLE 3.1  
Maximum smooth solution errors.

$\delta$	sln	$h$	$s$	$\text{err}_x$	$\text{erg}_x$	$\text{erg}_y$
.25	1	.05	1	.26e-2	.26e-2	.18
.26	1	.05	1	.23e-2	.23e-2	.18
.25	1	.025	1	.65e-3	.65e-3	.92e-1
.26	1	.025	1	.40e-3	.40 -3	.92e-1
.25	1	.25	3	.19e-8	.16e-4	.88e-3
.26	1	.25	3	.27e-6	.16e-4	.88e-3
.25	1	.125	3	.28e-10	.10e-5	.11e-3
.26	1	.125	3	.68e-7	.11e-5	.11e-3
.25	2	.05	1	.67e-3	.67e-3	.26e-1
.26	2	.05	1	.63e-3	.63e-3	.26e-1
.25	2	.025	1	.17e-3	.17e-3	.13e-1
.26	2	.025	1	.13e-3	.13e-3	.13e-1
.25	2	.25	3	.42e-8	.78e-5	.30e-3
.26	2	.25	3	.55e-7	.78e-5	.30e-3
.25	2	.125	3	.79e-10	.47e-6	.38e-4
.26	2	.125	3	.13e-7	.49e-6	.38e-4

The results recorded in Table 3.1 tend to confirm the convergence estimates claimed in Theorems 3.3 and 3.4. The global error in  $x$  behaves like  $O(h^{s+1})$ , i.e.,  $O(h^2)$  for  $s = 1$  and  $O(h^4)$  for  $s = 3$ , while the error in  $y$  behaves like  $O(h^s)$ . In case there is a difference between  $s + 1$  and  $2s$ , i.e., for  $s = 3$ , the error at mesh points is larger for the case  $\delta = .26$  (also for  $\delta = .25t^2$  which we tried as well) than for  $\delta = .25$ . For other cases, where the error is expected to be  $O(h^{s+1})$  regardless of whether the mesh is  $\pi^*$  or not, the errors for the two values of  $\delta$  are remarkably similar.

Note, however, that in (3.31) and (3.32) we have chosen a smooth exact solution: in cases when the solution is less smooth at points  $j\delta$  (for a constant  $\delta$ ), it can make a

big difference if such points are part of the mesh or not. In Table 3.2 we list numerical results corresponding to those of Table 3.1, but with the initial condition

$$(3.33) \quad x_2(t) = 0, \quad (t \leq 0)$$

replacing (3.30). Depending on the initial value for  $y$  we again converge to two different solutions, as before. For the “exact” solution used to compute the errors listed in Table 3.2 we have used a collocation computation with  $s = 3$  and  $h = 0.01$ , which yields a “dense” mesh  $\pi^*$  for both values of  $\delta$ .

TABLE 3.2  
*Maximum nonsmooth solution errors.*

$\delta$	sln	$h$	$s$	err <sub><math>x</math></sub>	erg <sub><math>x</math></sub>	erg <sub><math>y</math></sub>
.25	1	.05	1	.31e-2	.31e-2	.18
.26	1	.05	1	.35e-2	.35e-2	.18
.25	1	.025	1	.77e-3	.77e-3	.92e-1
.26	1	.025	1	.12e-2	.12e-2	.92e-1
.25	1	.25	3	.25e-4	.25e-4	.88e-3
.26	1	.25	3	.81e-3	.81e-3	.88e-3
.25	1	.125	3	.17e-5	.17e-5	.11e-3
.26	1	.125	3	.81e-3	.81e-3	.11e-3
.25	2	.05	1	.90e-3	.90e-3	.36e-1
.26	2	.05	1	.10e-2	.10e-2	.37e-1
.25	2	.025	1	.22e-3	.22e-3	.18e-1
.26	2	.025	1	.32e-3	.32e-3	.19e-1
.25	2	.25	3	.67e-5	.67e-5	.35e-3
.26	2	.25	3	.19e-3	.19e-3	.39e-3
.25	2	.125	3	.44e-6	.44e-6	.48e-4
.26	2	.125	3	.19e-3	.19e-3	.20e-3

From the results in Table 3.2 it is clear that having delay interval ends included as mesh points can be rather advantageous, especially for higher-order Runge–Kutta methods.  $\square$

**4. Higher-index DDAEs.** Although there are convergence theorems for some higher-index nondelay DAEs using methods like BDF [9] and certain Runge–Kutta methods [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  $\geq 3$ ) DAEs directly. This is true even without the introduction of delays. For nondelay DAEs, much recent work has therefore been directed at lower-index formulations of the problem which preserve the stability and lead to a robust and efficient numerical solution.

In [4], a wide variety of formulations were investigated for the (nondelay) high-index Hessenberg DAEs, and a class of promising formulations called *projected invariants* was 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 as

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

$$(4.1b) \quad 0 = g(x),$$

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

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

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  $\mu$ . Rewriting (4.2) in first-order form leads to a retarded Hessenberg index-2 system

$$(4.3) \quad \begin{aligned} x'_1 &= x_2 + G^T \mu, \\ x'_2 &= x_3, \\ &\vdots \\ x'_m &= \hat{f}(z(x(t)), z(x(t - 1))), \\ 0 &= g(x_1). \end{aligned}$$

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

$$(4.4) \quad \begin{aligned} x'_1 &= x_2 + G^T \mu, \\ x'_2 &= x_3, \\ &\vdots \\ x'_m &= f(z(x(t)), z(x(t - 1)), y), \\ 0 &= g^{(m)}(x_1, \dots, x_m), \\ 0 &= g(x_1). \end{aligned}$$

Additional derivatives of the original constraint can be enforced similarly, see [4], [5]. If it is important to preserve the higher-order structure, then a trick introduced in [5] can be used to produce such a stable formulation, which is given by

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

with  $\phi \equiv 0$  on  $[-1,0]$ . 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 [4], 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 [1] can be immediately extended to a stabilization method of the invariant manifold based on (4.1b) and its derivatives for the DODE (4.2).

**Appendix 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 §§1 and 2.

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

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

$$(A.1b) \quad 0 = g(x, x(t-1), y),$$

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

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

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

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

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 (A.1a). We may now apply the theory for retarded DODEs to (A.3) and expect, under certain reasonable conditions, that the initial value DDAE problem for  $x$  will be well posed and that a discontinuity in  $x'(0)$ , for example, will be propagated into a discontinuity in  $x^{(j+1)}(j)$ ,  $0 \leq j \leq J$ , in case of only one delay. (The smoothing is twice as slow in the case of two delays.)

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

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

(i.e.,  $D = -g_y^{-1}g_{y(t-1)}$ ). Differentiating (A.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 (A.3), we now have to propagate the recursion in (A.4) back from  $t$  to  $t-\nu-1$ , for  $\nu$  the integral part of  $t$ . This gives

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

When this expression is substituted into (A.1a), the obtained retarded DODE has  $j$  delays for  $j-1 \leq t < j$ , and this corresponds to a DODE of a neutral type. In particular, a discontinuity in  $x'(0)$  propagates as a discontinuity in  $x'(j)$ ,  $0 \leq j \leq J$ , so there is neither smoothing nor antismoothing by the inverse DODE operator. Note

also that initial values must be given here (on an interval in  $t \leq 0$ ) both for  $x$  and for  $y$ . The well-conditioning of the problem depends on the sum in (A.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 (A.4) with  $y$  replaced by  $\varepsilon y$ , and let  $\varepsilon \rightarrow 0$ . From the limit expression in (A.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

$$(A.6a) \quad x' = f(x, x(t-1), y, y(t-1)),$$

$$(A.6b) \quad 0 = g(x, x(t-1)),$$

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

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

This allows us to express

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

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

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

$$(A.9a) \quad x' = f(x, x(t-1), y),$$

$$(A.9b) \quad 0 = g(x).$$

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

**Appendix B. Collocation result for DODEs.** 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

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

$$(B.1b) \quad x(t) = \beta(t), \quad -1 < t \leq 0,$$

where the matrices  $A_j(t) \in \mathcal{R}^{n_x \times n_r}$  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 noncollocation 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  $\pi$  with no relative stepsize restrictions whatsoever may be used (cf. [2]). If there is an integer  $\mu$  such that  $h\mu = 1$  then the mesh includes the delay interval

ends and is denoted  $\pi^*$ , as in §3.2. We also assume for now that the problem (B.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) (B.1) at the collocation points. Therefore, the error

$$(B.2) \quad e(t) = x_\pi(t) - x(t)$$

satisfies homogeneous initial conditions and

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

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

The assumption of well-conditioning of (B.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 (B.3) then yields at collocation points  $e'(t_{nj}) = O(h^s)$ , and since  $x_\pi$  is a piecewise polynomial of degree  $< s$ , we have at all points other than mesh points (cf. [2]),

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

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

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

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

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

$$(B.6) \quad x'_j = \sum_{l=0}^{j-1} A_{lj} x_{j-l} + q_j + A_{jj} \beta(\tau - 1), \quad 0 < \tau < 1.$$

This is an ODE system of size  $Jn_x$ , for which we have the boundary conditions  $x_j(0) = x_{j-1}(1)$ ,  $j = 2, \dots, J$  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  $x_\pi$  and the errors  $d$  and  $e$ , respectively, in precisely the same way as  $\mathbf{x}$  relates to  $x$ , we obtain

$$(B.7) \quad \mathbf{e}(\tau) = \int_0^1 G(\tau, \sigma) \mathbf{d}(\sigma) d\sigma, \quad 0 \leq \tau \leq 1.$$

Now, if the mesh has the special structure  $\pi^*$  then  $\mathbf{d}(\tau_{nj}) = 0$ , where  $\tau_{nj} = t_{nj}$ ,  $n = 1, \dots, \mu$ , are the collocation points in  $\tau$ . In this case the ODE collocation theory immediately applies, and we obtain (cf. [2])

$$\begin{aligned} \text{(B.8a)} \quad & \|e\|_{L_\infty[0,J]} = O(h^{\min(s+1,k_d)}), \\ \text{(B.8b)} \quad & |e(t_n)| = O(h^{k_d}), \quad 0 \leq n \leq N. \end{aligned}$$

For a general mesh, let us write

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

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

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

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

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

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

The integrals in (B.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 (B.10), we have that  $d(t_{nl}) = 0$ ,  $l = 1, \dots, s$ , 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 (B.9), we obtain

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

Thus, the general order of convergence for the ODE case is recovered, as in (B.8a), even for a general mesh, but the superconvergence result (B.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 how  $G$  depends on  $J$ , but we do not pursue this question here.

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

## REFERENCES

- [1] U. ASCHER, H. CHIN, AND S. REICH, *Stabilization of DAE and invariant manifolds*, Numer. Math., 67 (1994), pp. 131–149.
- [2] U. ASCHER, R. MATTHEIJ, AND R. RUSSELL, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equation*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [3] U. ASCHER AND L. PETZOLD, *Projected implicit Runge–Kutta methods for differential–algebraic equations*, SIAM J. Numer. Anal., 28 (1991), pp. 1097–1120.
- [4] ———, *Stability of computational methods for constrained dynamics systems*, SIAM J. Sci. Comput., 14 (1993), pp. 95–120.
- [5] ———, *Projected collocation for higher-order higher-index differential-algebraic equations*, J. Comput. Appl. Math., 43 (1992), pp. 243–259.
- [6] G. BADER, *Solving boundary value problems for functional differential equations by collocation*, in Numerical Boundary Value ODEs, U. Ascher and R. Russell, eds., Birkhauser, Boston, 1985.
- [7] A. BELLEN, *One-step collocation for delay differential equations*, J. Comput. 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. BRENNAN, S. L. CAMPBELL, AND L. R. PETZOLD, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Elsevier, New York, 1989.
- [10] S. L. CAMPBELL, *Singular linear systems of differential equations with delays*, Appl. Anal., II (1980), pp. 129–136.
- [11] ———, *2-D (differential-delay) implicit systems*, in Proc. 13th IMACS World Congress on Computation and Applied Mathematics, 1991, pp. 1828–1829.
- [12] ———, *Nonregular descriptor systems with delays*, Proceedings of Symposium on Implicit and Nonlinear Systems, Automation and Robotics Research Institute, University of Texas-Arlington, Dec. 14–15, 1992, pp. 275–281.
- [13] C. W. GEAR, *Simulation: Conflicts between real-time and software*, in Mathematical Software III, Academic Press, New York, 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, Non-stiff Problems*, Springer-Verlag, New York, 1987.
- [16] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer-Verlag, New York, 1992.
- [17] J. HALE, *Theory of Functional Differential Equations*, Springer-Verlag, New York, 1977.
- [18] R. HAUBER, University of Munich, personal communication, 1992.
- [19] H. HEEB AND A. RUEHLI, *Retarded models for PC board interconnects - or how the speed of light affects your SPICE circuit simulation*, in Proc. IEEE ICCAD, 1991.
- [20] E. LELARSMEE, A. E. RUEHLI, AND A. L. SANGIOVANNI-VINCENTELLI, *The wavefront relaxation method for time domain analysis of large integrated circuits*, IEEE Trans. CAD IC Syst., 1 (1983), pp. 131–141.
- [21] B. LEIMKUHNER, L. R. PETZOLD, AND C. W. GEAR, *Approximation methods for the consistent initialization of differential-algebraic equations*, SIAM J. Numer. Anal., 28 (1991), pp. 205–226.
- [22] CH. LUBICH, *On projected Runge-Kutta methods for differential-algebraic equations*, BIT, 31 (1991), pp. 545–550.
- [23] K. SINGHAL, AT&T Bell Laboratories, Allentown, PA, personal communication, 1992.
- [24] R. D. SKEEL, *Waveform iteration and the shifted Picard splitting*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 756–776.
- [25] A. SKJELLUM, Dept. of Computer Science, Mississippi State University, personal communication, 1992.