Differential Algebraic Equations with After-Effect


Numerical Analysis Report No. 367

Oct 2000
Differential Algebraic Equations with After-Effect

C.T.H. Baker∗†, C.A.H. Paul† and H. Tian‡

Oct 1, 2000

Abstract

We consider the numerical solution of delay differential algebraic equations – they are differential algebraic equations with after-effect, or constrained delay differential equations. The general semi-explicit form of the problem consists of a set of delay differential equations along with a set of constraints that may involve retarded arguments. Even simply-stated problems of this type can give rise to difficult analytical and numerical problems. The more tractable examples can be shown to be equivalent to systems of delay or neutral delay differential equations. We shall explore, in particular through examples, some of the complexities and obstacles that can arise when solving these problems.

Key words. Delay differential algebraic equations, delay and neutral delay differential equations

AMS subject classifications. 34K05, 34K20, 35B25, 65C20

1 Introduction

Delay differential-algebraic equations (DDAEs) may be viewed from two different perspectives; one that DDAEs are differential-algebraic equations (DAEs) formulated with delayed solution terms, the other that DDAEs are delay differential equations (DDEs) subject to constraints. The discussion of DDAEs inherits many of the ideas and much of the terminology of both DAEs and DDEs (which

∗Director, MCCM; Honorary Senior Research Fellow, Chester College.
†This work was supported in part by EPSRC grant GR/L35218 & NAG Ltd.
‡This author was supported by an ORS grant and by Manchester University.
are special cases). However the interaction of algebraic constraints with delayed solution terms gives rise to behaviour that is not seen with either DAEs or DDEs. Thus DDAEs merit a separate investigation in their own right. As Petzold [26] has commented for DAEs, “DAEs are not ODEs”, and it is a fortiori true that DDAEs are in general neither DAEs nor DDEs. However some DDAEs may be reduced to a DDE, such as

$$
u'(t) = F(t, u(t), \alpha(t)) \quad (t \geq t_0),$$

where \(\alpha(t) \leq t\) and \(u(t) = U(t)\) for \(t \in [\inf_{t \geq t_0} \alpha(t), t_0]\). More generally, DDAEs may be differentiated analytically to yield a neutral delay differential equation (NDDE), such as

$$u'(t) = F(t, u(t), \alpha(t), u'(\gamma(t))) \quad (t \geq t_0),$$

where \(\alpha(t) \leq t\), \(\gamma(t) < t\), \(u(t) = U(t)\) for \(t \in [\inf_{t \geq t_0} \alpha(t), t_0]\) and \(u'(t) = U(t)\) for \(t \in [\inf_{t \geq t_0} \gamma(t), t_0]\).

2 Delay differential-algebraic equations

In this paper, we investigate various forms of the following semi-explicit DDAE

$$
\begin{align*}
x'(t) &= F(t, x(\alpha_1(t)), \ldots, x(\alpha_n(t)), y(\beta_1(t)), \ldots, y(\beta_n(t))), \\
0 &= G(t, x(\pi_1(t)), \ldots, x(\pi_n(t)), y(\omega_1(t)), \ldots, y(\omega_n(t))),
\end{align*}
$$

where \(t \in [t_0, \infty)\), \(x(t) \in \mathbb{R}^r\), \(y(t) \in \mathbb{R}^s\) and the delay functions satisfy \(\alpha_i(t) \leq t\), \(\beta_i(t) \leq t\), \(\pi_i(t) \leq t\) and \(\omega_i(t) \leq t\) for all \(i\). The analysis of eqn (3) can be considerably simplified by assuming that \(\pi_1(t) = t\) and \(\omega_1(t) = t\) (see §3.3). Where a discontinuity, that is to say a jump or a jump in a derivative, occurs in a solution, derivatives are interpreted as right-hand derivatives. In order to solve eqn (3), it is necessary to specify initial functions for \(x(t)\) and \(y(t)\) over some initial intervals \([t_{x_0}^*, t_0]\) and \([t_{y_0}^*, t_0]\), respectively. Although the values of \(t_{x_0}^*\) and \(t_{y_0}^*\) clearly depend on \(\alpha_i(t)\) and \(\pi_i(t)\), and on \(\beta_i(t)\) and \(\omega_i(t)\), respectively, they can do so in a highly complex manner (see §3.3).

By putting \(\alpha_i(t) = t\), \(\beta_i(t) = t\), \(\pi_i(t) = t\) and \(\omega_i(t) = t\) for all \(i\), eqn (3) becomes a standard semi-explicit DAE

$$
\begin{align*}
x'(t) &= F(t, x(t), y(t)), \\
0 &= G(t, x(t), y(t)).
\end{align*}
$$
There is a substantial literature on the analysis and solution of DAEs, and we refer the reader to Petzold [27], Brenan, Campbell & Petzold [5] and Hairer & Wanner [16]. By removing $y(t)$ and the constraints, eqn (3) becomes a standard DDE. We refer to Baker, Paul & Willé [3], Bellen & Zennaro [4] and Hairer, Norsett & Wanner [13] for an introduction to DDEs.

Several authors have investigated various forms of eqn (3): Campbell [7, 9] considered DDAEs with a single delayed differential variable $x(\cdot)$ with constant lag. The asymptotic stability of Hessenberg DDAEs with one delayed differential variable and one delayed algebraic variable $y(\cdot)$, both with the same constant lag, has been analyzed by Zhu & Petzold [31]. Ascher & Petzold [2] had previously analyzed the convergence of numerical methods for this class of Hessenberg DDAEs. Hauber [17] and Liu [21] considered the numerical solution of semi-explicit DDAEs of a restricted class of problems of the form (3). These papers provide valuable insight into various features of DDAEs, but the restrictions imposed on the delay functions or the form of the DDAE have meant that other features of DDAEs have gone unremarked. The paper [20] also merits mention.

2.1 Discontinuities and piecewise continuous solutions

One of the most significant features of DDEs is the propagation of discontinuities in the solution [22, 29]. Even if the derivative function $F$ and the initial function $X(\cdot)$ are both smooth, a discontinuity generally occurs in every solution component at the initial point $t_0$ because $X^{(v_i)}(t_0) \neq x^{(v_i)}(t_0)$ for some integer $v_i \geq 0$. These discontinuities are propagated into the solution whenever the value of a delay function $\alpha(t)$ crosses the initial point. However, whenever a discontinuity is propagated by a delayed solution term in a DDE, it occurs in a higher derivative [22]. Thus if the delay function has a fading memory, that is to say $\alpha(t) \to \infty$ as $t \to \infty$ for all $i$, then the solution $x(t)$ becomes smoother as $t \to \infty$. Discontinuities are also propagated in DAEs. They can be propagated by the differential equations, as for DDEs, and by the constraints, in which case they are not smoothed. However the absence of any delayed arguments in a DAE means that discontinuities only propagate between solution components and not between different times. Thus the propagation of discontinuities in DAEs has been largely ignored because it appears to have little impact on their solution.

For a DDAE, the occurrence of a delayed solution term in a constraint means
that discontinuities can be propagated between different times and solution components without smoothing. For example, consider the apparently fading memory DDAE [25, p.152]

\[
\begin{aligned}
  x'(t) &= y(t-1) \\
  y(t) &= x(t) - y(t-1)
\end{aligned}
\text{ for } t \geq 0 \text{ with } \begin{cases} 
  X(t) = 1 \text{ for } t \leq 0, \\
  Y(t) = 0 \text{ for } t < 0, \quad Y(0) = 1,
\end{cases}
\tag{5}
\]

whose solution appears in Fig. 1a. The solution \( y(t) \) inherits the discontinuity in \( Y(t) \) at \( t = 0 \) via the constraint, and consequently \( y(t) \) has jumps at all positive integer points. (Similar discontinuities occur in \( y(t) \) if \( Y(t) \) has a discontinuity at some \( t \in (-1, 0) \).) Thus \( y(t) \) exhibits behaviour generally associated with persistent memory problems, namely that the initial dynamics of the solution are never “forgotten”.

\begin{figure}[h]
  \centering
  \includegraphics[width=\textwidth]{fig1}
  \caption{Solution of eqn (5) and the reduced index equation (6), respectively.}
\end{figure}

An indication of the impact of discontinuities on numerical methods for solving DDAEs can be gained from looking at the effect in DDEs [25, §2]. If a discontinuity occurs in a sufficiently low derivative of the solution, then the asymptotic order of the integration method will not be attained and the asymptotic error estimator will be incorrect. This typically leads to an increased error in the solution, a higher proportion of rejected steps and a greater number of (unnecessarily) small steps [24, §3].

Defect control [11] and discontinuity tracking [29] are two strategies that have been used for dealing with discontinuities in DDEs. Both strategies can be extended to DDAEs, with varying degrees of success [17, 25]. Defect control is most effective on reasonably smooth problems – specifically problems that have a continuous defect – with fading memory. However, as eqn (5) shows,
the solutions of DDAEs may only be piecewise continuous and can exhibit the properties of persistent memory problems even when all the delay functions have fading memories. Discontinuity tracking can be applied to a DDAE after constructing the corresponding network dependency graph [25, 29]. If all the delay functions have constant lags then the network dependency graph is the same for all values of \( t \), and discontinuity tracking can be used. If a delay function has a non-constant lag, then not only may the network dependency graph need modifying as \( t \) changes, but discontinuities that are not predicted by the discontinuity tracking theory may occur (see §3.2).

2.2 Index reduction and functional differential equations

One method of solving DDAEs is to replace the algebraic variables in the differential equations by expressions (obtained from the constraints) involving only the differential variables. Thus it is sometimes possible to obtain a simpler differential equation (usually with deviating arguments) to solve. For example, the index 1 DDAE

\[ x'(t) = x(t)y(t - 2), \quad y(t - 4) = x(t - \tau(t)) \]

with lag function \( \tau(t) \geq 0 \), gives the differential equation \( x'(t) = x(t)x(t + 2 - \tau(t)) \). Depending on the size of \( \tau(t) \), the differential equation may be a DDE (\( \tau(t) > 2 \)), an ordinary differential equation (ODE) (\( \tau(t) = 2 \)) or an advanced differential equation (ADE) (\( \tau(t) < 2 \)); and so the nature of the problem can vary with \( t \). This feature of DDAEs giving rise to ADEs was first noted for constant lag DDAEs by Campbell [8].

A consequence of the next result is that, when appropriate conditions hold, a DDAE may be reduced to a DDE. (DAEs may be reduced to ODEs in a simpler manner.)

**Lemma 1** For any non-vanishing continuous delay function \( \alpha(t) \) defined on an interval \([t_0, T]\), there exists for each \( t \in [t_0, T] \) a unique integer \( q \equiv q(\alpha; t) \) such that \( \alpha^q(t) \leq t < \alpha^{q-1}(t) \), where \( \alpha^q(t) := \alpha(\alpha^{q-1}(t)) \) and \( \alpha^1(t) := \alpha(t) \).

For the constraint \( 0 = G(x(t), y(t), y(\omega(t))) \), if the derivative of \( G \) with respect to the second argument is invertible then there exists (by the implicit function theorem) a function \( \mathcal{G} \) such that \( y(t) = \mathcal{G}(x(t), y(\omega(t))) \). However, \( y(t) \) is given by an expression that still depends on the solution \( y(\cdot) \). Repeated
substitution of the expression for $y(t)$ into the constraint eventually yields (using the value $q(\omega; t)$ given by Lemma 1),

$$y(t) = G(x(t), G(x(\omega(t))), G(x(\omega^2(t))), \ldots, G(x(\omega^q(t)), y(\omega^q(t))))).$$

Thus $y(t)$ can be expressed solely in terms of $x(t)$ and $Y(t)$, and so a DDE for $x(t)$ may be obtained. For DDAEs with $n$ delayed algebraic variables in the constraints, there can be as many as $1 + \sum_{j=1}^{\max q} n^j$ terms involving $G$ in the expression for $y(t)$.

Another standard approach for solving DAEs is to differentiate the constraints analytically [12]. For DDAEs, differentiating a constraint (where valid) may yield an NDDE. However care should be taken with this approach. For example, differentiating the constraint in eqn (5) gives, formally, a system of NDDEs:

$$x'(t) = y(t - 1), \quad y'(t) = x'(t) - y'(t - 1) \text{ for } t \geq 0.$$  \hspace{1cm} (6)

The solution to eqn (6) satisfying the initial conditions of eqn (5) appears in Fig. 1b. However, because the solution $y(t)$ to eqn (5) is discontinuous at all integer points, the constraint cannot be differentiated at these points and so the solutions $x(t)$ and $y(t)$ of the reduced index problem are incorrect as solutions of eqn (5).

Campbell [7] noted that continuous solutions of DDAEs may exist only on finite intervals. However, as shown by eqn (5), such DDAEs can have well-defined piecewise continuous solutions. Ideally a general DDAE solver should be able to cope with such problems.

3 Numerical solution of DDAEs

We shall present various strategies for treating DDAEs. The discussion of numerical methods for DDAEs is simplified by making the following assumption:

**Assumption 2** When solving a DDAE on the interval $[t_i, t_{i+1}]$, a (piecewise) continuous extension of the solution is available for all $t \leq t_i$.

3.1 Numerical methods

The choice of integration method for solving DDAEs is influenced by experience gained from solving DAEs and DDEs. It is well-known that DAEs can be
extremely “stiff”, and thus codes for solving DAEs are mostly based on stiff ODE methods, in particular BDF methods [5]. However the propagation of discontinuities in DDAEs, as shown in §2.1, can have a greater impact on the smoothness of solutions than in the DDE case; and for non-smooth DDEs with persistent memory, one-step numerical methods are generally favoured [3, §3.3]. Thus for a general DDAE, one-step stiff integration methods, such as implicit Runge-Kutta (IRK) and collocation methods, seem likely to provide the most suitable methods. The requirement for DAEs that the constraints are satisfied at the end of each step means that sufficiently accurate [16, p. 46] Runge-Kutta (RK) methods, such as the Radau IIA methods, have been preferred [14, 17].

Stiffness is a potentially controversial topic: Stiffness in DDEs requires a more sophisticated analysis than stiffness in ODEs. Stiffness is usually associated with the property that an explicit integration method uses small stepsizes even though the local truncation error is small. Features that are sometimes taken as characteristics of stiffness in ODEs appear not to be diagnostic for DDEs. For example, the scalar DDE \( u'(t) = au(t) + bu(t - 1) \) \( (b \neq 0) \) is infinite-dimensional and incorporates a countably infinite number of “different timescales”. (The characteristic equation \( \nu - a - b \exp(-\nu) \) has a countable set of zeros \( \{\nu_i\}_{i=0}^{\infty} \) and the general solution of the DDE can be expressed as \( \sum_i p_i(t) \exp(\nu_i t) \).) Furthermore, whereas \( cu'(t) = au(t) \) with \( a < 0 \) is sometimes said to become “infinitely stiff” as \( \epsilon \searrow 0 \) – the solution \( u_\epsilon(0) \exp(at/\epsilon) \) decays with increasing speed as \( \epsilon \searrow 0 \) – there is a limit to the rate of exponential decay and exponential stability of solutions of \( cu'(t) = au(t) + bu_\epsilon(t - 1) \) as \( \epsilon \searrow 0 \) with \( -a > |b| \). In the latter case, the dominant zero of the quasi-polynomial \( c\nu - a - b \exp(-\nu) \) has a finite limit as \( \epsilon \searrow 0 \).

The jury appears to be out on the question of the optimum design of numerical methods for DDAEs. For simplicity, consider a direct RK discretization of

\[
\begin{align*}
x'(t) &= f(t, x(t), y(t), x(\alpha(t)), y(\beta(t))), \\
0 &= g(t, x(t), y(t), x(\pi(t)), y(\omega(t)))
\end{align*}
\]

using a grid \( T := \{t_0 < t_1 < t_2 < \cdots \} \) where \( t_{i+1} = t_i + h_i \) and continuous RK parameters \((c, A, b(s))\). Here \( A \) is a matrix with elements \( a_{ij} \), \( b(s) \) is a vector of polynomials in \( s \) and \( c \) is a vector of RK abscissae \( c_i \). For convenience, we assume that the abscissae \( c_i \) are distinct and satisfy \( 0 < c_i \leq 1 \).
Remark 3 The continuous RK method can be associated with ‘quadrature’

$$\tilde{\varphi}(c_i) - \tilde{\varphi}(0) = \sum_j a_{ij} \tilde{\varphi}'(c_j), \quad \tilde{\varphi}(s) - \tilde{\varphi}(0) = \sum_j b_j(s) \tilde{\varphi}'(c_j) \text{ for } s \in [0,1],$$  \hspace{1cm} (7)

which are compatible formulae if (as we suppose here) $a_{ij} = b_j(c_i)$. If $A$ is invertible with $A^{-1} = [a^{-1}_{ij}]$ then $\tilde{\varphi}(s) - \tilde{\varphi}(0) = \sum_j b_j(s) \sum_k a^{-1}_{jk} \{ \tilde{\varphi}(c_k) - \tilde{\varphi}(0) \}$ and we obtain

$$\tilde{\varphi}(s) = A_0(s) \tilde{\varphi}(0) + \sum_j A_j(s) \tilde{\varphi}(c_j) \text{ for } s \in [0,1].$$  \hspace{1cm} (8)

Eqn (8) defines a polynomial approximation for $\tilde{\varphi}(s)$. If the polynomials $b(s)$ and $\{A_j(s)\}$ are used to define approximations for $x(t)$ and $y(t)$, respectively, then Assumption 2 can be satisfied.

The RK parameters above give rise to the formulae

$$\tilde{x}(t_{k,i}) = x_k + h_k \sum_j a_{ij} f(t_{k,j}, \tilde{x}(t_{k,j}), \tilde{y}(t_{k,j}), \tilde{x}(\alpha(t_{k,j})), \tilde{y}(\beta(t_{k,j})))$$

where $t_{k,i} = t_k + c_i h_k$. The delayed solution values can be expressed as either $\tilde{x}(t_{\mu+i})$ or $\tilde{y}(t_{\nu+i})$ for $\mu, \nu \in \mathbb{N}$, $c_i \in [0,1]$ and $s \in [0,1]$, where

$$\tilde{x}(t_{\mu+i}) = \tilde{x}_\mu + h_\mu \sum_j b_j(\zeta) f(t_{\mu,j}, \tilde{x}(t_{\mu,j}), \tilde{y}(t_{\mu,j}), \tilde{x}(\alpha(t_{\mu,j})), \tilde{y}(\beta(t_{\mu,j}))),$$

$$\tilde{y}(t_{\nu+i}) = A_0(s) \tilde{y}(t_\nu) + \sum_j A_j(s) \tilde{y}(t_\nu + c_j h_\nu).$$

The above discretization results from a direct treatment of the DDAE, but it can also be obtained by using the continuous RK method to discretize the singularly perturbed delay differential equation (SPDDE)

$$x'_\epsilon(t) = f(t, x_\epsilon(t), y_\epsilon(t), x_\epsilon(\alpha(t)), y_\epsilon(\beta(t))),$$

$$c y'_\epsilon(t) = g(t, x_\epsilon(t), y_\epsilon(t), x_\epsilon(\pi(t)), y_\epsilon(\omega(t)))$$  \hspace{1cm} (9)

then, formally, setting $\epsilon$ to zero – compare the treatment for index 1 DAEs in [16, pp. 402–404]. There are other approaches, for example, solving a related SPDDE such as eqn (9) (see [10, pp.168–172] and §3.5), or reducing the DDAE to an underlying DDE or NDDE.
3.2 Index reduction

When the constraint has multiple algebraic solution terms with time-varying lag functions, index reduction can become complicated. The additional difficulties that are associated with DDAEs with non-constant lag functions, namely

- constraint equations may only apply on limited intervals;
- the correct choice of expression for algebraic variables can depend on $t$;
- constraint equations may require non-linear time-shifts, which may themselves give rise to problems of existence and uniqueness,

have so far gone unremarked in the literature, and we illustrate them here.

Consider, for example, the DDAE

$$x'(t) = x(t)y(t), \quad y(\omega_1(t))y(\omega_2(t)) = x(t-2) \quad \text{for} \quad t \geq 0. \quad (10)$$

If $\omega_1(t) = \sin(t)$ and $\omega_2(t) = 1 - \cos(t)$ then the constraint is valid for all $t \geq 0$, however it only imposes a constraint on $y(t)$ for $t \in [0, 2]$. Thus, with these delay functions, eqn (10) is under-determined for $t > 2$ – and differentiating the constraint does not help. For state-independent delay functions that only depend on $t$, it is possible to determine the constraint range before solving the DDAE. However for state-dependent delay functions that depend on $x(t)$ and/or $y(t)$, the constraint range can only be determined after solving the DDAE.

If $\omega_1(t) = t \sin(t)$ and $\omega_2(t) = t \cos(t)$, then the constraint range is $[0, \infty)$ as $t \to \infty$. However $\omega_1(t) \geq \omega_2(t)$ for $t \in [(2i + \frac{1}{4})\pi, (2i + \frac{5}{4})\pi]$ and $\omega_1(t) \leq \omega_2(t)$ for $t \in [(2i - \frac{3}{4})\pi, (2i + \frac{1}{4})\pi]$ for integer $i$ (see Fig. 2). In order to avoid advanced solution terms arising, on some intervals $y(t)$ should be determined using $y(\omega_1(t)) = x(t-2)/y(\omega_2(t))$ and on other intervals using $y(\omega_2(t)) = x(t-2)/y(\omega_1(t))$. This change in the definition of $y(t)$ results in a discontinuity in $x'(t)$; however, this type of discontinuity is not predicted by the discontinuity tracking theory of Willé & Baker [29] because it does not arise from an existing discontinuity but from a change in the comparative sizes of two delay functions.

Obtaining an expression for $y(t)$ involves a non-linear time-shift of the constraint, so that

$$y(t) = \begin{cases} 
  x(\omega_1^{-1}(t) - 2)/y(\omega_2(\omega_1^{-1}(t))) \quad & \text{if} \quad \omega_1(t) \geq \omega_2(t), \\
  x(\omega_2^{-1}(t) - 2)/y(\omega_1(\omega_2^{-1}(t))) \quad & \text{if} \quad \omega_1(t) \leq \omega_2(t).
\end{cases} \quad (11)$$
However neither delay function has a closed-form inverse or even a uniquely defined inverse. Noting that every delay function satisfies $\omega_i(t) < t$, for consistency $\omega_i^{-1}(t)$ should satisfy $\omega_i^{-1}(t) > t$. Defining $\omega_i^{-1}(t)$ to be the smallest number $s$ such that $\omega_i(s) = t$ with $s > t$ yields uniquely defined inverse delay functions. However, substituting these inverse delay functions into eqn (11) yields an ADE at $t \approx 1.82$.

3.3 Initial functions and consistent initial conditions

For the DDE (1), $U(t)$ need only be specified on the interval

$$[t^*_\alpha, t_0], \text{ where } t^*_\alpha := \inf_{t \geq t_0} \alpha(t).$$

For the DAE (4), an initial consistency condition is $0 = G(t_0, x(t_0), y(t_0))$ – higher index DAE problems have additional “hidden” consistency conditions [27]. The problem of determining a set of consistent initial conditions for DAEs has attracted much interest, see [6] and its references, and is still an active research area.

For the DDAE (3), there are several difficulties associated with specifying a consistent set of initial conditions. For simplicity of presentation, we assume that $\mathbf{x}(t), \mathbf{y}(t) \in \mathbb{R}$, and that at time $t$ the delay functions are (re)ordered so that $\pi_i(t) \geq \pi_{i+1}(t)$, $\beta_i(t) \geq \beta_{i+1}(t)$ and $\omega_i(t) \geq \omega_{i+1}(t)$.

1. The first difficulty is determining over what interval the initial functions
are required. In order to substitute the value of \( y(\beta_i(t)) \) in the differential equation, it is necessary to shift the time-axis of the constraint so that the implicit function theorem yields

\[
y(\beta_1(t)) = G(s, x(\pi_1(s)), \ldots, x(\pi_{n_s}(s)), y(\omega_2(s)), \ldots, y(\omega_{n_w}(s))),
\]

where \( s = \omega_1^{-1}(\beta_1(t)) \). Thus the initial intervals for \( X(t) \) and \( Y(t) \) are

\[
[\min_i \{ t_{n_i}^s, \min_{i \geq t_0} \pi_i(\omega_1^{-1}(\beta_1(t))) \}, t_0], [\min_i \{ t_{n_i}^s, \min_{i \geq t_0} \omega_i(\omega_1^{-1}(\beta_1(t))) \}, t_0],
\]

respectively; note that both \( \beta_1(t) \) and \( \omega_1(t) \) may change with \( t \).

2. Just as with DAEs, the initial conditions for DDAEs should also be consistent with the constraints. Thus for eqn (3),

\[
0 = G(t_0, x(\pi_1(t_0)), \ldots, x(\pi_{n_s}(t_0)), y(\omega_1(t_0)), \ldots, y(\omega_{n_w}(t_0))). \tag{12}
\]

Note that eqn (12) does not necessarily impose conditions on \( x(t_0) \) and \( y(t_0) \).

By putting \( t = \pi_1^{-1}(t_0) \) and \( t = \omega_1^{-1}(t_0) \), respectively, consistency conditions for \( x(t_0) \) and \( y(t_0) \) may be obtained. However, if in doing so \( \pi_i(\pi_1^{-1}(t_0)) > t_0 \) or \( \omega_i(\omega_1^{-1}(t_0)) > t_0 \) for some \( i \), then a different ordering of the \( \pi_i(t) \) or \( \omega_i(t) \), respectively, must be considered. When putting \( t = \pi_1^{-1}(t_0) \), if \( \omega_i(\pi_1^{-1}(t_0)) > t_0 \) for some \( i \) then \( y(t_0) \) is not totally determined by the other initial conditions, and if \( \omega_i(\pi_1^{-1}(t_0)) > \pi_1^{-1}(t_0) \) for some \( i \) then eqn (3) gives rise to an ADE. A similar analysis applies for \( t = \omega_1^{-1}(t_0) \).

In some cases, the analysis is made considerably simpler:

- If \( \beta_1(t) = t \), then no reordering of the \( \beta_i(t) \) is necessary.
- If every \( \beta_i(t) \) has a constant lag, then no reordering of the \( \beta_i(t) \) is necessary.
- If \( \pi_1(t) = t \), then eqn (12) imposes a consistency condition on \( x(t_0) \).
- If every \( \pi_i(t) \) has a constant lag, then no reordering of the \( \pi_i(t) \) is necessary.
- If \( \omega_1(t) = t \), then \( \omega_1^{-1}(t) = t \), no reordering of the \( \omega_i(t) \) is necessary and eqn (12) imposes a consistency condition on \( y(t_0) \).
• If \( \omega_1(t) = t - \tau \), then \( \omega_1^{-1}(t) = t + \tau \).

• If every \( \omega_i(t) \) has a constant lag, then no reordering of the \( \omega_i(t) \) is necessary.

• If \( \omega_1(t) = \beta_1(t) \), then no time-axis shift is required.

Most of the papers on DDAEs are concerned with equations that fall into one or more of the above categories.

### 3.4 Neutral delay differential equations, continuous extensions and defect

The possibility of NDDEs arising in the solution of a DDAE imposes additional requirements on a numerical method, in particular on the continuous extension used for the solution. It is well-known for DDEs that a \( p \)-th order integration method requires a \( (p+1) \)-st local order continuous extension in order to obtain an asymptotically correct error estimator [1]. As Arndt pointed out, the error approximating a solution consists of two separate components: (i) the error corresponding to the integration error and, (ii) the error corresponding to the error in approximating the numerical solution.

When solving NDDEs, a continuous extension of the derivative of the solution is required. Whilst such an extension may be obtained by differentiating the continuous extension of the solution [30, p.121], the subsequent loss of order means that the error estimator may no longer be asymptotically correct. This loss of order arises from a reduction in the order of the approximation error, and so it can be avoided by using a higher degree continuous extension. The idea of using a non-minimal degree approximation for evaluating delayed solution values in DDEs was first suggested by Oberle & Pesch [23, §3.3] for multi-step Hermite interpolants.

Several classes of IRK method have superconvergent solutions at the mesh-points when applied to ODEs and DAEs. Superconvergence at meshpoints is generally lost when IRK methods are applied to DDEs unless delayed solution values can be evaluated to sufficient order. The construction of suitable continuous extensions for IRK methods applied to DDEs has already been widely considered [18, §1.2]. In particular, for \( A \)-stable IRK methods with distinct abscissae satisfying \( c_i < 1 \) for all \( i \), the “natural interpolation” scheme – the local
collocation polynomial – has been shown to have poor stability properties when applied to DDEs [19].

Hairer, Lubich, and Roche [15] (see also [14, 16]) defined the perturbation index $\nu$ along a solution of a DAE. This definition can be extended to define the perturbation index of a solution of a DDAE, for example:

**Definition 4** The DDAE (3) has perturbation index $\nu$ along a solution $u(t) = (x(t), y(t))$ on $[t_0, T]$, if $\nu$ is the smallest integer such that

$$||\tilde{u}(t) - u(t)|| \leq C \left( \sup_{t \leq t_0} ||\tilde{u}(t) - u(t)|| + \sup_{t_0 \leq s \leq t} ||\eta(s)|| + \cdots + \sup_{t_0 \leq s \leq t} ||\eta^{(\nu-1)}(s)|| \right)$$

for all approximate solutions $\tilde{u}(t) = (\tilde{x}(t), \tilde{y}(t))$ with defect $\eta(t)$, whenever the bound is sufficiently small. The defect $\eta(t) = (\eta_F(t), \eta_C(t))$, where

$$\eta_F(t) := x'(t) - F(t, x(\alpha_1(t)), \ldots, x(\alpha_{\alpha_n}(t)), y(\beta_1(t)), \ldots, y(\beta_{\beta_n}(t))),$$

$$\eta_C(t) := G(t, x(\pi_1(t)), \ldots, x(\pi_{\pi_n}(t)), y(\omega_1(t)), \ldots, y(\omega_{\omega_n}(t))).$$

Under appropriate conditions (but see the discussion of eqn (6) above), diffe rentiating the constraints in a DDAE $m$ times gives rise to an NDDE that is satisfied by $x(t)$ and $y(t)$. It follows that a theorem of the form given below provides, with suitable assumptions, a bound $\nu \leq m + 1$ on the perturbation index $\nu$, or the result $\nu = m$ (cf. [16, pp.478–481]):

**Theorem 5** Let $u(t)$ satisfy eqn (2) and suppose that

$$\tilde{u}'(t) = F(t, \tilde{u}(t), \tilde{u}(\alpha(t)), \tilde{u}'(\gamma(t))) + \delta(t)$$

for $t \in [t_0, T]$, $\tilde{u}(t) = U(t) + \delta U(t)$ for $t \in [t^\star_n, t_0]$, $\tilde{u}'(t) = U(t) + \delta U(t)$ for $t \in [t^\star_n, t_0]$. If $|| \cdot ||$ is a vector-norm and

$$||F(t, u, v, w) - F(t, \tilde{u}, \tilde{v}, \tilde{w})|| \leq L_1||u - \tilde{u}|| + L_2||v - \tilde{v}|| + L_3||w - \tilde{w}||$$

uniformly for $t \in [t_0, T]$ and positive constants $L_1, L_2, L_3$, then there exists $K(T) > 0$ such that

$$\sup_{t \in [t_0, T]} ||u(t) - \tilde{u}(t)|| \leq K(T) \left\{ \sup_{t \in [t^\star_n, t_0]} ||\delta U(t)|| + \sup_{t \in [t^\star_n, t_0]} ||\delta U(t)|| + \sup_{t \in [t_0, T]} ||\delta(t)|| \right\}.$$  

Lemma 1 and a Gronwall inequality allow a straightforward proof of this result.

The discussion indicates that controlling the error in the numerical solution of a DDAE with perturbation index $\nu > 1$ involves controlling the sizes of the defect and certain derivatives of the defect.
3.5 Singularly perturbed delay differential equations and large derivatives

A SPDDE has connections to both DDEs and DDAEs. Eqn (3) is the degenerate case of the SPDDE

\[
\begin{align*}
    x'_\epsilon(t) &= F(t, x_\epsilon(t), x_\epsilon(t-\tau), y_\epsilon(t), y_\epsilon(t-\tau)), \\
    \epsilon y'_\epsilon(t) &= G(t, x_\epsilon(t), x_\epsilon(t-\tau), y_\epsilon(t), y_\epsilon(t-\tau)),
\end{align*}
\]  

(13)

where \( t \in [0, \infty), \epsilon > 0, x_\epsilon(t) \in \mathbb{R}^r \) and \( y_\epsilon(t) \in \mathbb{R}^s \). By putting \( \epsilon = 1 \), the SPDDE becomes a standard \((r+s)\)-dimensional system of DDEs and, by putting \( \epsilon = 0 \), it becomes a semi-explicit DDAE with \( r \) differential variables and \( s \) algebraic variables.

When \( x_\epsilon(t) \to x(t) \) and \( y_\epsilon(t) \to y(t) \) as \( \epsilon \searrow 0 \), an alternative approach to solving a DDAE directly is to solve a sequence of SPDDEs for decreasing values of \( \epsilon \). For simplicity, we only consider the pair of scalar equations

\[
\begin{align*}
    x'_\epsilon(t) &= F(t, x_\epsilon(t), x_\epsilon(t-\tau), y_\epsilon(t), y_\epsilon(t-\tau)), \\
    \epsilon y'_\epsilon(t) &= G(t, x_\epsilon(t), x_\epsilon(t-\tau), y_\epsilon(t), y_\epsilon(t-\tau)),
\end{align*}
\]  

(14)

where \( \tau > 0 \), and \( F \) and \( G \) are real functions that satisfy suitable conditions (detailed in [28]). In particular, we require \( \frac{\partial}{\partial \tau} G(t, x, u, y, v) \leq -\kappa < 0 \) on a suitable region. By putting \( \epsilon = 0 \), we obtain the equations satisfied by \((x(t), y(t))\).

For any integer \( N \geq 0 \), there exist constants \( C_N \) and \( \epsilon_0 \) such that the solution \((x_\epsilon(t), y_\epsilon(t))\) of eqn (14) with given initial conditions satisfies

\[
\begin{align*}
    x_\epsilon(t) &= \sum_{j=0}^{N} X_j^{(k)}(t) \epsilon^j + \epsilon \sum_{j=0}^{N-1} \Xi_j^{(k)}(\frac{t-k\tau}{\epsilon}) \epsilon^j + S_N^{(k)}(t, \epsilon), \\
    y_\epsilon(t) &= \sum_{j=0}^{N} Y_j^{(k)}(t) \epsilon^j + \epsilon \sum_{j=0}^{N} \Pi_j^{(k)}(\frac{t-k\tau}{\epsilon}) \epsilon^j + R_N^{(k)}(t, \epsilon),
\end{align*}
\]

on each interval \([k\tau, (k+1)\tau]\) where \( |S_N^{(k)}(t, \epsilon)|, |R_N^{(k)}(t, \epsilon)| \leq C_N \epsilon^{N+1} \) for \( 0 < \epsilon \leq \epsilon_0 \). The functions \( X_j^{(k)}(t) \) and \( Y_j^{(k)}(t) \) and the exponentially decaying functions \( \Xi_j^{(k)}(\cdot) \) and \( \Pi_j^{(k)}(\cdot) \) (which define “boundary layers”) are solutions of a set of \( \epsilon \)-independent differential equations. We can deduce the following convergence result:

**Theorem 6** Given the above conditions, \( \lim_{\epsilon \searrow 0} x_\epsilon(t) = x(t) \) and \( \lim_{\epsilon \searrow 0} y_\epsilon(t) = y(t) \) for \( t \in [0, T] \) except at points \( t = \tau, 2\tau, 3\tau, \ldots \).
There are some possible advantages to solving a related SPDDE:

- SPDDEs have continuous solutions for all values of $t > t_0$.
- SPDDEs can never give rise to NDDEs or ADEs.
- SPDDEs have index 0 (but they can become increasingly difficult to solve as $\epsilon \searrow 0$).
- SPDDEs do not require a consistent set of initial conditions. However if $x_\epsilon(t)$ and $y_\epsilon(t)$ are to converge to $x(t)$ and $y(t)$, respectively, then the initial conditions for the SPPDE should satisfy (at least to within $O(\epsilon)$) the constraints of the DDAE.

For example, a SPDDE version of eqn (5) is

\[
\begin{align*}
x_\epsilon'(t) &= y_\epsilon(t - 1) \\
y_\epsilon'(t) &= x_\epsilon(t) - y_\epsilon(t - 1)
\end{align*}
\] for $t \geq 0$ with \[
\begin{align*}
X_\epsilon(t) &= 1 \text{ for } t \leq 0, \\
Y_\epsilon(t) &= 0 \text{ for } t < 0, \quad Y_\epsilon(0) = 1.
\end{align*}
\] (15)

Solutions of eqn (15) for $\epsilon = 0.02$ and $\epsilon = 0.002$ appear in Fig. 3. Both $y_{\epsilon=0.02}(t)$ and $y_{\epsilon=0.002}(t)$ exhibit typical behaviour of solutions of SPDDEs, namely boundary layers of width $O(\epsilon)$ where the corresponding degenerate solution $y_{\epsilon}=0(t)$ has a jump. Although the SPDDE has solutions that become smoother as $t \to \infty$, because the degenerate eqn (5) has a solution $y(t)$ that has jumps at all integer points, the boundary layers in $y_\epsilon(t)$ occur at all integer points. Thus whilst the solution of the SPDDE eventually becomes sufficiently smooth that tracking discontinuities is no longer necessary, the continuing occurrence of boundary layers in the solution $y_\epsilon(t)$ means that these remain computationally significant. As $\epsilon \searrow 0$, the size of $|y_\epsilon'(t)| \to \infty$ in the boundary layer. Thus tracking boundary layers in SPDDEs can be just as important as tracking discontinuities. Whilst a jump in $y_{\epsilon=0}(t)$ always gives rise to a boundary layer in $y_\epsilon(t)$, large derivatives in $y_\epsilon(t)$ do not always correspond to jumps in $y_{\epsilon=0}(t)$. Therefore it is not possible to use the DDE discontinuity tracking approach to predict where boundary layers are likely to occur.

4 Further comments

There are numerous difficulties that can arise when solving a general DDAE. Many of the additional difficulties arising in DDAEs are associated with multiple
non-constant lag functions, and thus they have so far gone unnoticed in the literature. Some difficulties are unavoidable, such as the problems that arise when one or more solution components are only piecewise continuous. For this class of problems, differentiating the constraints analytically to yield a lower index DDAE is not possible. If algebraic manipulation of the DDAE does not produce a DDE or NDDE, then direct piecewise-continuous solution methods might be used.

Solving a DDAE (3) by solving a related SPDDE (13) can also, where justified, be attractive. This approach avoids many of the problems identified with solving DDAEs, except that the initial conditions of the SPDDE must also be consistent initial conditions for the DDAE in some sense. It is possible to use existing DDE codes for solving SPDDEs, though SPDDEs often become increasingly difficult to solve numerically as $\epsilon \downarrow 0$. Non-stiff DDE solvers may be unsuitable for very small values of $\epsilon$.

The problem of the propagation of discontinuities in DDAEs is still not adequately dealt with by existing DDE strategies. Defect control can be very inefficient on problems with poor continuity. Discontinuity tracking in DDAEs is generally more complex (and computationally expensive) than in the DDE case, but more importantly it can fail to track some low-order discontinuities. Thus an improved strategy for dealing with discontinuities is required for the efficient solution of a general DDAE.

A major concern in the numerical solution of DAEs is that the solutions "drift off" the constraints. This problem is complicated for DDAEs by the fact that values of delayed solution terms are obtained from (piecewise) continuous

Figure 3: Solutions of the singularly perturbed version of eqn (5).
extensions that generally do not satisfy the constraints except at a limited number of points. For an $A$-stable IRK method applied to a DDE, (i) the use of non-minimal degree continuous extensions can improve numerical performance and (ii) there can be stability problems when using the local collocation polynomial. This observation may affect our choice of extension in the case of DDAEs; how it applies to approximating the solutions of constrained variables is unclear.

The equivalence of a DDAE to an ADE usually means that the DDAE cannot be solved numerically but, in some cases, it may still be possible. For example, consider the DDAE

$$
x'(t) = y(t-1)x(t-t^2), \quad y(t-2) = x(t-1)x(t-2). \quad (16)
$$

Most numerical methods solve differential equations synchronously, that is, both $x(t)$ and $y(t)$ are calculated at time $t$. For eqn (16), calculating $y(t)$ at time $t$ using the constraint produces the advanced problem $y(t) = x(t+1)x(t)$. However, at time $t$, in order to solve the differential equation for $x(t)$, only the value of $y(t-1)$ is required, and the constraint gives $y(t-1) = x(t)x(t-1)$. Thus by solving $x(t)$ and $y(t)$ asynchronously – to be precise, by solving eqn (16) for $x(t)$ and $y(t-1)$ at time $t$ – an advanced problem is avoided.

Acknowledgements The authors have benefitted from a number of discussions with colleagues, notably Ruth Thomas, Volker Wulf (who experimented with a Radau IIA code for DDAEs) and Neville Ford from MCCM, and members of the groups associated with Dirk Roose in Leuven, and Alfredo Bellen in Trieste. The authors are indebted to Nicola Guglielmi and Ernst Hairer for drawing the authors’ attention to their development of a Radau IIA code available at http://www.unige.ch/math/folks/hairer.

References


