Stepsize control and continuity consistency for state-dependent delay-differential equations

David R. Willé \(^1\), Christopher T.H. Baker *

Department of Mathematics, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom

Received 5 February 1992; revised 25 November 1992

Abstract

Derivative discontinuities occur frequently in the solutions of delay-differential equations, even when the functions defining them are all \(C^\infty\). Unless correctly treated, such discontinuities can undermine the continuity assumptions made by ODE software. In this paper we discuss some specific difficulties arising from this aspect in the treatment of state-dependent problems. A necessary property, continuity consistency, is introduced along with a class of methods for which it is satisfied.

Keywords: State-dependent delay-differential equations; Derivative discontinuities; Variable-step linear multistep methods; Continuity consistency

1. Introduction

In this paper we consider a class of numerical methods applied to the solution of delay-differential problems of the form

\[
y'(t) = f(t, y(t), y(\alpha(t, y(t)))) , \quad b \geq t \geq a ,
\]

\[
y(t) = \phi(t) , \quad t \leq a ,
\]

(1)

where \(\alpha^* \leq \alpha(t, y(t)) \leq t\), \(f\), \(\phi\) and \(\alpha\) are all assumed \(C^\infty\) and \(y \in \mathbb{R}\). \(\alpha^*\) is here a real constant. In particular, our attention is directed towards state-dependent problems where the lag-function \(\alpha(t, y)\) is allowed to vary not only as a function of \(t\) but also of \(y(t)\). For convenience, our discussion is divided into three separate stages:

- the use of ODE codes in DDE software (Section 2);

\(^*\) Corresponding author.

\(^1\) This author would like to thank IBM Semea srl. for the provision of facilities used in the preparation of this paper.

Present address: Mathematical Applications, Ciba-Geigy AG, CH-4002 Basel, Switzerland.
• the accommodation of derivative discontinuities and the notion of continuity consistency (Sections 4–7); and
• the application of continuity consistency to predictor–corrector methods (Section 8).

The notion of continuity consistency, which we introduce in Section 6, will turn out to be central to our argument. Not only does it appear aesthetically pleasing, but it also helps explain, and hence avoid, a number of internal contradictions that can otherwise arise for state-dependent problems. In Section 8, it is used to introduce a modified predictor–corrector scheme suitable for such cases. Note that although, for ease of discussion, a deliberately simple form (1) is adopted, more general forms can be treated without difficulty.

2. Extended ODE methods

Delay equations of type (1) are examples of evolutionary problems. Defining

\[ z(t,u) := y(\alpha(t,u)), \]

they may thus be reduced to the form

\[ y'(t) = f(t,y(t),z(t,y(t))), \quad b \geq t \geq a, \]

where \( y(t) = \phi(t) \) for \( t \leq a \). Given suitable methods to record and approximate the past solution \( z(t,y(t)) \), the class (1) can then, in principle, be solved using ODE techniques. We shall refer to methods which follow this approach, and which reduce delay equations to related ODE problems, as extended ODE methods. Such methods are, under various names, widely used in practice and appear extensively in the literature. See, for example, [1,4,7,9] (as discussed by [13]), [11,12,15] and particularly [10]. For a general introduction to ODE schemes, the reader is referred to [9]. In this paper, assuming a familiarity with the underlying themes introduced by the above references, we consider the application of extended ODE methods to state-dependent problems.

Clearly, owing to their dependence on ODE software, the provision of suitable and efficient ODE codes is essential to the development of extended ODE methods. As for ODE methods, the choice of suitable stepsize and order selection strategies is crucial to achieving this extension. For example, ODE stepsize control — which we term primary stepsize control — is required to keep local truncation (or discretisation) errors in scale with, that is approximately equal to, user-specified error tolerances.

3. Derivative discontinuities

As is well known [8], the solutions to equations of the form (1) need not lie in \( C^{\infty} \), even if the functions \( f, \alpha \) and \( \phi \) are continuous in all derivatives. A derivative discontinuity can arise at the initial point \( t_0 = a \), for instance, if \( \phi'(a) \neq f(a, y(a), y(\alpha(a, y(a)))) \), before being propagated by the delayed term \( y(\alpha(t, y(t))) \) to later times. The solution to the state-independent problem

\[ y'(t) = y(t - 1), \quad t \geq 0, \quad y(t) = 1, \quad t \leq 0, \]

for example, has an initial discontinuity in \( y' \) at \( t = 0 \) which is then propagated to discontinuities in \( y^{(k+1)} \) at the later times \( t = k > 0, \ k \in \mathbb{Z} \). For extended ODE methods, the presence of such
discontinuities is clearly potentially significant. Unless suitable action is taken, they may undermine the numerical schemes chosen to approximate (2) and (3).

4. Secondary stepsize control

One natural approach to accommodate derivative discontinuities in extended ODE methods is to attempt to modify their stepsize control strategies so that all derivative discontinuities occur at mesh points \(\{t_i\}\) computed by the numerical scheme. This ensures that the step interiors \(\{(t_i, t_{i+1})\}\) are all maintained continuity-free. This is our idealised continuity requirement. Given appropriate conditions—typically that \(y \in C^1(\mathbb{R}) \cap C^k([t_i, t_{i+1}])\) for some suitable \(k\), for all \(i\) —derivative discontinuities can then be shown not to invalidate the error estimates of the formulae used to approximate (2) and (3). A more general discussion of this appears in [15]. The implementation of such techniques, in general, is however nontrivial. An additional level of stepsize control is required over and above that used by the ODE code to control the local error—primary stepsize control—simply in order to meet continuity requirements. We refer to this as secondary stepsize control. This paper investigates the application of secondary stepsize control to state-dependent problems.

5. Locating derivative discontinuities

Let \(S_n\) be the set of all the positions of derivative discontinuities \(\xi \leq t_n\) of the true solution, where \(t_n\) is the starting point for the current step \([t_n, t_n + h]\). Then, by elementary analysis, it can be shown that for (1), a further derivative discontinuity in \(y\) at \(t = \xi' > t_n\) can only arise when the lag-point \(\alpha(t, y(t))\) crosses the position of some previous derivative discontinuity \(\xi\) in \(S_n\):

\[
\xi = \alpha(\xi', y(\xi')).
\]

This recurrence relation provides the basis of a numerical method to detect and locate derivative discontinuities as the integration advances. Secondary stepsize control on the step \([t_n, t_n + h]\) is then merely equivalent to the detection and location of solutions to equations of the form

\[
\xi = \alpha(\xi'_A, y_A(\xi'_A)), \quad \xi'_A \in [t_n, t_n + h],
\]

for \(\xi \in S_n^*,\) where \(S_n^*\) is a previously computed set of numerical approximations to \(S_n\). Here \(y_A\) is a numerical estimate of the true solution \(y\) and \(h\) is the initially attempted stepsize. For state-independent problems the stepsize is simply reduced, \(h \rightarrow \tilde{h} := \xi'_A - t_n\). If more than one solution \(\xi'_A\) exists in \([t_n, t_n + h]\), the minimum such \(\tilde{h}\) is taken. When applied to state-dependent problems, however, this approach is nontrivial. The location of \(\xi'_A\) requires an estimate of \(y\) over \([t_n, t_n + h]\), but this is typically not available before the step is advanced. Since a value for \(\xi'_A\) is required to determine the stepsize, the overall scheme thus becomes implicit in \(y\). Hence secondary stepsize control schemes for state-dependent problems not only require an in-interval approximant, but are also, in general, implicit in the \(y\)-values.

In summary, therefore, two additional problems arise in general for extended ODE methods applied to state-dependent problems.
Tracking the positions of derivative discontinuities, i.e., forming the set $\mathcal{E}_n$. This cannot be done exactly in general as only approximate solution values are known.

- Determining an appropriate stepsize for the current step; this becomes implicit for general state-dependent problems.

The above observations provide the starting point for our discussion.

6. Multistage methods: an example

In multistage methods, i.e., those which make more than one function evaluation per step, further problems can arise. We illustrate this by reference to Runge–Kutta and predictor–corrector linear multistep schemes. The formulae used in such methods to approximate $y\big|_{[t_n, t_{n+1}]}$ are typically updated after each new derivative function evaluation, and so estimates of $\xi'$, defined above, will in general vary between different stages. When this occurs, a decision must be made as to which value should be used by the secondary stepsize control to constrain the stepsize. This is the central issue considered in this paper. To illustrate the problem, consider the modified Euler or Heun’s method [6]

\begin{align}
\dot{y}_{n+1}^p &= y_n + h_n F(t_n, y_n), \\
\dot{y}_{n+1} &= y_n + \frac{1}{2} h_n \left[ F(t_n, y_n) + F(t_{n+1}, y_{n+1}^p) \right], \\
h_n = t_{n+1} - t_n, \quad t_0 = a, \quad y_0 = y(a),
\end{align}

(6)

defined for the ordinary differential equation

\begin{align}
y'(t) &= F(t, y(t)).
\end{align}

(7)

Defining, in the notation of (3), $F(t, y) := f(t, y, z(t, y))$, we then apply this to the state-dependent problem

\begin{align}
y'(t) &= \frac{1}{t} \exp(y(\alpha(t, y(t)))), \quad t \in [1, 3], \\
y(t) &= 0, \quad t \in [-\ln 2 + 1, 1],
\end{align}

(8)

where $\alpha(t, u) = u - \ln 2 + 1$. This is of interest because the analytic solution [10]

\begin{align}
y(t) &= \begin{cases} 
\ln t, & t \in [1, 2], \\
\frac{1}{2} t + \ln 2 - 1, & t \in [2, 3],
\end{cases}
\end{align}

is known, and has a discontinuity in its second derivative at $t = 2$. Over $[1, 2]$ the problem (8), however, reduces to an ODE since

\begin{align}
y(t) - \ln 2 + 1 \leq 1.
\end{align}

For $t \leq 2$. In particular for (7), whilst $t \in [1, 2]$, we can write

\begin{align}
F(t, y) = \frac{1}{t}.
\end{align}

(9)
The implementation of secondary stepsize control beyond \([1, 2]\) is, however, nontrivial. Suppose, for example, that we had wished to advance a step across \([t_n, t_{n+1}]\), but found that
\[
y_{n+1}^p - \ln 2 + 1 > 1,
\]
whilst
\[
y_n - \ln 2 + 1 < 1.
\]
Then, by the previous observations, \([t_n, t_{n+1}]\) may contain a derivative discontinuity at some interior point \(\xi' \in [t_n, t_{n+1}]\). Constructing an obvious interpolant
\[
y_p(t) = y_n + (t - t_n)F(t, y_n),
\]
the position of the discontinuity may be approximated by \(\xi_p'\):
\[
1 = \alpha(\xi_p', y_p(\xi_p')),
\]
and the resulting value used to restrict the stepsize, \(h_n \rightarrow \overline{h}_n := \xi_p' - t_n\). Equivalently we force \(t_{n+1} := \xi_p'\). This is secondary stepsize control. Clearly such a stepsize reduction is necessary to ensure that the continuity requirements for Heun’s method are met in the current step. Advancing the second stage of the step, however, the final lag-point estimate
\[
\alpha(t_{n+1}, y_{n+1}) = y_{n+1} - \ln 2 + 1
\]
still lies to the left of the discontinuity at \(t = 1\). This follows since, in the special case (9),
\[
\int_{t_n}^{t_{n+1}} F(t, y(t)) \, dt < \frac{1}{2} h_n \left[ F(t_n, y(t_n)) + F(t_{n+1}, y(t_{n+1})) \right] < h_n F(t_n, y_n),
\]
which thus implies
\[
y(t_{n+1}) < y_{n+1} < y_{n+1}^p.
\]
Hence the effect of the discontinuity is effectively postponed until the following step. By induction, the same problem will occur on \([t_{n+1}, t_{n+2}]\), and on a sequence of intervals of decreasing length\(^2\)
\[
([t_i, t_{i+1}])_{i \geq n},
\]
Here \(h_i := t_{i+1} - t_i \rightarrow 0\) and \(t_i \rightarrow 2\) as \(i \rightarrow \infty\), which is clearly unsatisfactory. This example demonstrates that, in the design of numerical schemes, we should ensure not only that any necessary continuity requirements are fulfilled, but also that secondary stepsize control returns end points \(\{t_m\}\) such that
\[
\alpha(t_m, y_m) \in \Xi_n^*, \quad m_i \leq n.
\]
Here, as before, \(\Xi_n^*\) denotes the set of all numerical estimates of previous derivative discontinuities. We refer to any method with this property as being continuity consistent. Clearly our above requirements imply that \(\Xi_n^* \subseteq \{t_i, i \leq n\}\).\(^2\)
\(^2\)In practical codes the sequence is always truncated after a few steps due to machine round-off or related effects.
7. Continuity consistency

The notion of continuity consistency is central to our discussion. When implementing algorithms for secondary stepsize control, the positions of derivative discontinuities should be taken as solutions of (5), rather than solutions of (4). At every step the numerical method effectively solves a perturbed problem—cf. (3)—defined in terms of the existing numerical solution. Where the continuity structure of the perturbed problem differs from that of the original problem, it is that of the perturbed problem which should be used in stepsize selection.

This observation is closely related to the notion underlying continuity consistency. Continuity consistency is required to ensure that the numerical formulae defined by the differential equation and the past numerical solution are not compromised by derivative discontinuities. This is necessary to implement secondary stepsize control correctly. If a method is not continuity consistent, then secondary stepsize controls may fail, as illustrated by the above example, and the ODE solver and associated primary stepsize controls may be compromised by a lack of solution continuity.

8. Multistepsize methods

To construct extended ODE methods which are continuity consistent, but which remain unaffected by derivative discontinuities, one interesting approach is to use predictor-corrector methods to solve the ODE subproblem (3). In a predictor-corrector method, the advanced point solution $y(t_{n+1})$ is first approximated by an explicit formula known as a predictor, before being refined, or corrected, by a corrector formula, usually based on an implicit expression. In such schemes it is, in principle, possible to adjust the current stepsize between individual stages of the same integration step to take account of changing estimates of discontinuity positions. We illustrate this below with an example based upon Heun’s method. More sophisticated examples of this approach can be found in the linear-multistep codes REBUS by Bock and Schlöder [4,5] and DELSOL by Willé [16]. For further details of [5], see [14].

Since (in all of these schemes) more than one stepsize can be used to advance across any given step, we refer to methods of this type as multistepsize methods. Consider once again Heun’s method applied to the model problem (8). As before, the problem reduces to an ODE over $[1, 2]$. Advancing the “predictor” $y_{n+1}^p$ as above, and constructing the corrector interpolant

$$y_c(t) - y_n = \int_{t_n}^t \left\{ f_n + \left( \frac{s-t_n}{h_n} \right) \left( f_{n+1}^p - f_n \right) \right\} \, ds$$

over $t \in [t_n, t_{n+1}]$, the method simply requires a second stepsize adjustment $\bar{h}_n \rightarrow \bar{h}_{n} := \xi'_{c} - t_n$, where

$$\alpha(\xi', y_{\xi'}) = \xi,$$

immediately prior to the second function evaluation. The integration can then be continued as before, but upon completion yields the result

$$\alpha(t_{n+1}, y_{n+1}) = \xi,$$
i.e., continuity consistency. Here, using the notation of (6), \( f_n := F(t_n, y_n) \) whilst \( f^P_n := F(t_n, y^P_n) \).

Methods of this type are clearly powerful, not only because they are continuity consistent, but because they do not violate the continuity requirements of the underlying ODE solver. This can be shown, since at every stage the formulae constructed by the code refer only to arguments on one side of any given derivative discontinuity. A further attraction of multistepsize methods is that the iterations required to solve the equations of the form (10) and (11) are independent of the derivative function \( f \), and so no additional derivative function evaluations are required. The iterations are implicit merely in the approximation scheme used for \( y \) and the lag-function \( \alpha \). For a fuller discussion, the reader is once again referred to [15].

9. Practical schemes

Whereas the implementation of stepsize control in extended ODE methods for state-independent problems may be considered relatively straightforward, our observations suggest that the same need not be true for state-dependent problems. In general, modifications are needed to meet continuity requirements. Our observations are however theoretical, and thus it is not clear how significant they are in practice. That will depend upon a number of factors including the relative orders of different integration stages, as well as the exact nature of the problem under solution. Indeed, the issue becomes even less clear given the problem of locating derivative discontinuities\(^3\), or in the case where discontinuities occur only in higher derivatives. Difficulties can also arise where lag-evaluation points lie in the current integration interval. This issue is addressed in the context of explicit Runge–Kutta methods in [2,3]. All these issues deserve closer investigation.

Before concluding, it should be stressed that the issue of secondary stepsize control is nontrivial even for state-independent problems. There is, for example, no reason to suppose that an appropriate, or in scale (see Section 2), stepsize used before a derivative discontinuity will remain in scale afterwards. Thus, it could be argued that a re-start procedure similar to that which would be triggered after a failure of the type described in Section 6 may actually be appropriate since the numerical properties of the solution will in general change across derivative discontinuities. We do not, however, pursue this argument here. For Adams linear multistep methods of the type used in [5,15], multistepsize techniques appear both simple and natural. That the stepsize corrections made are typically only small is perhaps to be expected because of the high order of the difference between predictor and corrector stages. This is, in turn, a consequence of the higher order of the information passed between steps in multistep methods.

10. Conclusion

In the numerical solution of state-dependent problems, special care should be taken to ensure the correct treatment of derivative discontinuities when and where they arise. A useful conceptual tool for doing this is continuity consistency. Continuity consistency is required to ensure that numerical

\(^3\)Immediately adjacent derivative discontinuities may be difficult to resolve in practice and, in some cases, may be safely ignored. See [15] for further details.
schemes remain consistent with the past numerical solutions in terms of which they are defined. Although the realisation of continuity consistency for multistage methods appears to be nontrivial—problems arise, for example, using certain Runge–Kutta formulae—it is however not impossible. In particular, success has been achieved using a class of methods derived from ODE predictor–corrector schemes.

References