ELSEVIER

Journal of Computational and Applied Mathematics 58 (1995) 151–169

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Equilibrium states of adaptive algorithms for delay differential equations

Desmond J. Higham*, Ioannis Th. Famelis[1]

Department of Mathematics and Computer Science, University of Dundee, Dundee, DD1 4HN, United Kingdom

Received 24 February 1993; revised 17 November 1993

Abstract

This work examines the performance of explicit, adaptive, Runge–Kutta based algorithms for solving delay differential equations. The results of Hall (1985) for ordinary differential equation (ODE) solvers are extended by adding a constant-delay term to the test equation. It is shown that by regarding an algorithm as a discrete nonlinear map, fixed points or *equilibrium states* can be identified and their stability can be determined numerically. Specific results are derived for a low order Runge–Kutta pair coupled with either a linear or cubic interpolant. The qualitative performance is shown to depend upon the interpolation process, in addition to the ODE formula and the error control mechanism. Furthermore, and in contrast to the case for standard ODEs, it is found that the parameters in the test equation also influence the behaviour. This phenomenon has important implications for the design of robust algorithms. The choice of error tolerance, however, is shown not to affect the stability of the equilibrium states. Numerical tests are used to illustrate the analysis. Finally, a general result is given which guarantees the existence of equilibrium states for a large class of algorithms.

Keywords: Runge–Kutta method; Error control; Fixed point; Delay

1. Introduction

Using a standard ordinary differential equation (ODE) solver as the basis of a delay differential equation (DDE) algorithm is, conceptually, a straightforward matter. It is known, however, that great care must be taken in order to preserve desirable convergence and stability properties (see, for example, [8, 11]). In this work we examine explict Runge–Kutta (RK) formulae when adapted to solve a DDE test equation. A key feature of our analysis is that it takes account of the whole

* Corresponding author. E-mail: na.dhigham@na-net.ornl.gov.

[1] Present address: Department of Mathematics, National Technical University of Athens, Zografou Campus, Athens, Greece.

SSDI 0377-0427(93)E0268-Q

algorithm — the ODE formula, the interpolation process and the error control strategy. Our results enable us to make detailed predictions about the performance of the algorithm. The work is based on the equilibrium theory that Hall [5] developed to analyse ODE solvers. We outline below our basic notation and definitions.

Given an initial value ODE

$$y'(t) = f(t, y(t)), \quad y(0) = y_0, \tag{1.1}$$

an s-stage explicit RK formula advances the approximation $y_n \approx y(t_n)$ to $y_{n+1} \approx y(t_{n+1})$ according to

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_i &= f\left(t_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad 2 \leqslant i \leqslant s, \\ y_{n+1} &= y_n + h_n \sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{1.2}$$

Here, h_n is the current stepsize and the coefficients $\{a_{ij}, b_i, c_i\}$ define a particular formula. The stepsize h_n is usually varied from step to step, in order to control some estimate of the error. Typically, the error estimate has the form

$$\text{est}_{n+1} = \| \text{err}_{n+1} \|, \quad \text{where } \text{err}_{n+1} = h_n \sum_{i=1}^{s} e_i k_i. \tag{1.3}$$

The quantity err_{n+1} may be an estimate of the local error in y_{n+1}, or, in the case of local extrapolation, of the local error in some other approximation. The form (1.3) also covers defect control [1]. The result (1.2) is accepted if $\text{est}_{n+1} \leqslant \text{TOL}$, where TOL is a user-supplied tolerance. If $\text{est}_{n+1} > \text{TOL}$, then the procedure is repeated with a smaller stepsize. An asymptotically-based formula for choosing the next stepsize is

$$h_{\text{new}} = \left(\frac{\theta \text{TOL}}{\text{est}_{n+1}}\right)^{1/q} h_n. \tag{1.4}$$

Here, q is the largest integer such that $\text{est}_{n+1} = O(h_n^q)$, and $\theta^{-1/q} h_{\text{new}}$ is the optimal stepsize in the sense that, asymptotically, it is the largest stepsize with which the next attempted step will be accepted. The constant safety factor $\theta \in (0, 1)$ is included to reduce the possibility of a step rejection. Formula (1.4) can be used after acceptance or rejection. Other alternatives, such as simply halving the stepsize, are sometimes used following a rejected step. In our analysis and testing we assume that (1.4) always determines the next stepsize; however, the qualitative predictions that we make do not depend upon the precise details of the step rejection process.

The algorithm above simplifies considerably when (1.1) is taken to be the standard, scalar, linear test equation

$$y'(t) = \lambda y(t), \quad \lambda \in \mathbb{R}, \ \lambda < 0, \qquad y(0) = y_0. \tag{1.5}$$

In this case one successful step of the algorithm may be written (see, for example, [5])

$$y_{n+1} = S(h_n\lambda)\, y_n, \qquad h_{n+1} = \left(\frac{\theta \mathrm{TOL}}{|E(h_n\lambda)\, y_n|} \right)^{1/q} h_n, \tag{1.6}$$

where S is the well-known stability polynomial of the RK formula, and E is the error polynomial.

We may convert (1.1) into a DDE by allowing the right-hand side to depend upon the solution at some time $t - \tau$, where τ is a fixed positive constant; that is,

$$y'(t) = F(t, y(t), y(t-\tau)), \qquad y(t) = \Phi(t) \quad \text{for } t \in [-\tau, 0]. \tag{1.7}$$

A standard approach for solving (1.7) is to apply an ODE solver to a problem of the form

$$y'(t) = F(t, y(t), q(t-\tau)), \qquad y(0) = \Phi(0).$$

For $t \in [-\tau, 0]$, we can take $q(t) = \Phi(t)$, and for $t > 0$, $q(t)$ is found by interpolating previously computed data. In this work we concentrate on two widely-used interpolants. Suppose that the point t lies in the interval $[t_{n-m}, t_{n-m+1})$ with $t = t_{n-m} + \sigma h_{n-m}$, so that $0 \leqslant \sigma < 1$, and suppose approximations $y_{n-m} \approx y(t_{n-m})$, $y_{n-m+1} \approx y(t_{n-m+1})$, $f_{n-m} \approx y'(t_{n-m})$ and $f_{n-m+1} \approx y'(t_{n-m+1})$ are available. Then the *linear Lagrange* interpolant is defined by interpolating $\{y_{n-m}, y_{n-m+1}\}$:

$$q(t) = (1-\sigma)\, y_{n-m} + \sigma y_{n-m+1}. \tag{1.8}$$

The *cubic Hermite interpolant* is defined by interpolating $\{y_{n-m}, f_{n-m}, y_{n-m+1}, f_{n-m+1}\}$:

$$q(t) = d_1(\sigma) y_{n-m} + d_2(\sigma) y_{n-m+1} + h_{n-m} e_1(\sigma) f_{n-m} + h_{n-m} e_2(\sigma) f_{n-m+1}, \tag{1.9}$$

where

$$d_1(\sigma) := 2\sigma^3 - 3\sigma^2 + 1, \qquad d_2(\sigma) := -2\sigma^3 + 3\sigma^2,$$

$$e_1(\sigma) := \sigma^3 - 2\sigma^2 + \sigma, \qquad e_2(\sigma) := \sigma^3 - \sigma^2.$$

We mention that under certain circumstances, suitable approximations $\{y_{n-m+1}, f_{n-m+1}\}$ may not be readily available — this issue is addressed later.

A DDE analogue of (1.5) is the test equation

$$y'(t) = \lambda y(t) + \mu y(t-\tau), \qquad y(t) = \Phi(t) \quad \text{for } t \in [-\tau, 0], \tag{1.10}$$

where $\lambda, \mu \in \mathbb{R}$, $\tau > 0$ and $\Phi(t)$ is presumed to be continuous. A great deal of research has been done on the long term behaviour of constant stepsize methods applied to (1.10); see, for example, [12] and the references therein. Typically, assumptions are made about the parameters $\{\lambda, \mu, \tau\}$ to ensure that $y(t) \to 0$ as $t \to \infty$, and the question addressed is: what restriction (if any) must be placed on the stepsize to ensure that $y_n \to 0$ as $n \to \infty$? In particular, *delay-independent* stability is often considered. It is known (see, for example, [16]) that $\lambda \leqslant -|\mu|$ and $\lambda < -|\mu|$ are necessary and sufficient, respectively, to guarantee that $y(t) \to 0$ as $t \to \infty$ for all choices of τ. It is, therefore, natural to ask whether $y_n \to 0$ as $n \to \infty$ for all choices of τ, and important results in this area have

been derived in [16, 12]. We also point out that these references allow λ and μ in (1.10) to be complex. Our work has a different emphasis. Here, we take a particular problem of the form (1.10) and seek to determine how a modern, adaptive algorithm is likely to behave. We will asume that $\lambda < -|\mu|$, so that $y(t) \to 0$ is guaranteed. We mention that Baker and Paul [14] recently considered a related issue concerning the behaviour of fixed stepsize algorithms on (1.10).

It is worth noting that DDEs generally have solutions with low order derivative discontinuities. These are propagated forward in time, eventually becoming of sufficiently high order that they can be ignored. Hence, efficient handling of discontinuities is an important issue at the start of an integration. We do not discuss this aspect further, however, since we are concerned with the long term behaviour of adaptive algorithms.

In the next section, we review the equilibrium theory of Hall, as applied to (1.5). The following three sections extend this approach to the DDE (1.10), each section dealing with a different algorithm. We find conditions under which equilibrium states exist and then investigate numerically how the stability of an equilibrium state depends on the test equation parameters. In each case we are able to prove that the stability is independent of the error tolerance, TOL. Numerical results are presented to illustrate the applicability of the theory. The final section summarises the key differences that arise on moving from the ODE (1.5) to the DDE (1.10). We also give a general result that guarantees the existence of an equilibrium state for a large class of algorithms.

2. Equilibrium theory for ODEs

Ignoring step rejections, the adaptive RK recurrence (1.6) may be regarded as a discrete nonlinear iteration of the form

$$\begin{bmatrix} y_{n+1} \\ h_{n+1} \end{bmatrix} = G\left(\begin{bmatrix} y_n \\ h_n \end{bmatrix}\right). \tag{2.1}$$

Hall [5] identified a fixed point, or *equilibrium state*, of this iteration. To define this state, first we let h_L be the stepsize that corresponds to the absolute stability boundary; that is, h_L is the smallest (positive) stepsize such that $|S(h_L\lambda)| = 1$. Now let y_L satisfy $|E(h_L\lambda)y_L| = \theta\mathrm{TOL}$. Then it follows from (1.6) that with $y_n = y_L$ and $h_n = h_L$,

$$|y_{n+1}| = |S(h_L\lambda)y_L| = |y_L| = |y_n| \quad \text{and} \quad h_{n+1} = \left(\frac{\theta\mathrm{TOL}}{|E(h_n\lambda)y_L|}\right)^{1/q} h_L = h_n.$$

Hence, we see that $|y_n|$ and h_n remain constant. If $S(h_L\lambda) = 1$, then y_n is constant and we have a period one fixed point of (2.1). Otherwise, $S(h_L\lambda) = -1$ so that y_n oscillates in sign, giving a fixed point of (2.1) with period two.

Note that the fixed point identified by Hall is a reasonable solution to the ODE — it uses the largest stable stepsize and it produces a global error that is O(TOL). Hall argued that the fixed point may arise in practice if it is stable with respect to small perturbations. This stability is governed by the Jacobian, G'. For $S(h_L\lambda) = 1$, first order stability of the period one fixed point is

equivalent to

$$\rho\left(G'\left(\begin{bmatrix} y_L \\ h_L \end{bmatrix}\right)\right) < 1, \tag{2.2}$$

where $\rho(\cdot)$ denotes the spectral radius, and for $S(h_L\lambda) = -1$, first order stability of the period two fixed point is equivalent to

$$\rho\left(G'\left(\begin{bmatrix} y_L \\ h_L \end{bmatrix}\right) G'\left(\begin{bmatrix} -y_L \\ h_L \end{bmatrix}\right)\right) < 1. \tag{2.3}$$

Hall showed that conditions (2.2) and (2.3) can both be reduced to

$$\rho\left(\begin{bmatrix} 1 - \dfrac{h_L\lambda}{q}\dfrac{E'(h_L\lambda)}{E(h_L\lambda)} & -\dfrac{1}{q} \\ \dfrac{h_L\lambda S'(h_L\lambda)}{S(h_L\lambda)} & 1 \end{bmatrix}\right) < 1. \tag{2.4}$$

The key point about (2.4) is that the condition is independent of λ; it depends only on the RK algorithm. (Specifying a particular RK formula automatically determines the point $h_L\lambda$ on the absolute stability boundary.) Hence, a single algebraic condition involving the RK and error control coefficients governs the stability of the fixed point on all problems of the form (1.5).

Results in [5] showed that some algorithms satisfy (2.4) while others do not. When (2.4) holds, the stepsize approaches the value h_L and remains virtually constant at that level with no step rejections occurring. The numerical solution y_n also settles into a corresponding period one or two state. On the other hand, if (2.4) does not hold then the stepsize is seen to oscillate above and below h_L. Steps are frequently rejected when stepsizes above the h_L level are chosen. The numerical solution follows a similar nonsmooth pattern. Such behaviour is undesirable for two reasons. First, the step rejections (typically one in every three steps) represent wasted computation. Second, the global error in the numerical solution, while remaining about the size of TOL, varies erratically and can differ by factors of more than $\frac{3}{2}$ from step to step.

The analysis above has been extended to the problem $y'(t) = Ay(t)$ where A is a constant matrix. The scalar λ must now be interpreted as a dominant eigenvalue of A; that is, an eigenvalue for which the condition $|S(h\lambda)| < 1$ is most restrictive on h. When λ is complex, the analysis only covers the case where A is normal and the Euclidean norm is used in (1.3) [6, 7]. We also mention that Higham and Trefethen [10] argue that when A is highly nonnormal, predictions based on eigenvalues are likely to be invalid.

The main purpose of this work is to extend Hall's analysis to the test equation (1.10). In particular, we wish to demonstrate that a DDE algorithm does not automatically inherit the characteristics of the underlying ODE solver. The choice of interpolant and the values of the parameters in the test equation also play important roles.

3. Improved Euler method with linear Lagrange interpolation

The ODE formulae that we analyse here are Euler's method and the Improved Euler method. When applied to (1.1) these formulae can be regarded as a two-stage embedded RK pair of orders

1 and 2:

$$k_1 = f(t_n, y_n), \tag{3.1}$$

$$k_2 = f(t_n + h_n, y_n + h_n k_1), \tag{3.2}$$

$$y_{n+1}^{\mathrm{E}} = y_n + h_n k_1, \tag{3.3}$$

$$y_{n+1}^{\mathrm{IE}} = y_n + 0.5 h_n (k_1 + k_2). \tag{3.4}$$

In this section, we consider the case where y_{n+1}^{IE} is used for the numerical approximation y_{n+1}, with $\mathrm{est}_{n+1} = \| y_{n+1}^{\mathrm{E}} - y_{n+1}^{\mathrm{IE}} \|$, so that $q = 2$ in (1.4). We suppose that the linear Lagrange interpolant (1.8) is used for $q(t)$.

The ODE stability polynomial for the Improved Euler formula is $S(z) = 1 + z + z^2/2$, and hence the largest stable stepsize for (1.5) is given by $h_{\mathrm{L}} = -2/\lambda$. Since $S(-2) = +1$, the corresponding equilibrium state is a period one fixed point. Hence, it is reasonable to look for an analogous period one fixed point for the DDE algorithm on (1.10). Our approach is, therefore, to seek a constant solution of the recurrence with, say, $h_n \equiv h_{\mathrm{D}}$ and $y_n \equiv y_{\mathrm{D}}$. (Equivalently, we are asking for the characteristic polynomial to have a root equal to $+1$.) To proceed with the analysis, we must be precise about the ratio of the delay to the stepsize, since this determines the step number of the recurrence. Let the integer m and the real number $\sigma \in [0, 1)$ be defined by

$$(m-1)h_{\mathrm{D}} < \tau \leqslant m h_{\mathrm{D}}, \qquad \tau + \sigma h_{\mathrm{D}} = m h_{\mathrm{D}}. \tag{3.5}$$

Hence, when a constant stepsize of h_{D} is used, $t_n - \tau$ lies in the interval $[t_{n-m}, t_{n-m+1})$ and the numerical method applied to (1.10) produces an $(m+1)$-step recurrence. We consider first the case $m = 1$ (that is, $0 < \tau \leqslant h_{\mathrm{D}}$). This is actually a special case. The value $q(t_n + h_{\mathrm{D}} - \tau)$ required for k_2 in (3.2) is not available at the start of the stage since the interpolant needs the data y_{n+1}. To keep the method explicit, we will therefore assume that $q(t)$ in (1.8) uses the Euler approximation $y_n + h_n k_1$, rather than y_{n+1}, in this case.

Under these assumptions, the interpolated value $q(t_n - \tau)$ becomes $(1-\sigma)y_{\mathrm{D}} + \sigma y_{\mathrm{D}} = y_{\mathrm{D}}$ and hence, in (3.1),

$$k_1 = (\lambda + \mu) y_{\mathrm{D}}. \tag{3.6}$$

Using $q(t_{n+1} - \tau) = (1-\sigma)y_{\mathrm{D}} + \sigma[y_{\mathrm{D}} + h_{\mathrm{D}} k_1]$, we find that k_2 in (3.2) reduces to

$$k_2 = (\lambda + \mu)(1 + h_{\mathrm{D}}(\lambda + \sigma\mu)) y_{\mathrm{D}}. \tag{3.7}$$

Now, in order for y_n to be constant from step to step, we must have $k_1 + k_2 = 0$ in (3.4). If $y_{\mathrm{D}} \neq 0$, then using $\sigma = 1 - \tau/h_{\mathrm{D}}$ it follows from (3.6) and (3.7) that

$$h_{\mathrm{D}} = \frac{-2 + \mu\tau}{\lambda + \mu}. \tag{3.8}$$

For h_n to remain constant we need $0.5 h_{\mathrm{D}} |k_2 - k_1| = \theta \mathrm{TOL}$, which gives

$$|y_{\mathrm{D}}| = \frac{2\theta \mathrm{TOL}}{h_{\mathrm{D}}^2 (\lambda + \mu)(\lambda + \mu\sigma)}. \tag{3.9}$$

The values (3.8) and (3.9) define an equilibrium state, provided that h_D satisfies $h_D \geqslant \tau > 0$. This condition reduces to $\tau \leqslant -2/\lambda$, and hence is satisfied for sufficiently small τ.

To analyse the stability of this equilibrium state, we regard the algorithm as a nonlinear map $v_{n+1} = G(v_n)$, where $v_n = [y_n, h_n, y_{n-1}, h_{n-1}]^T$. The Jacobian has the form

$$G'(v) = \begin{bmatrix} \frac{\partial G_1}{\partial y_n} & \frac{\partial G_1}{\partial h_n} & \frac{\partial G_1}{\partial y_{n-1}} & \frac{\partial G_1}{\partial h_{n-1}} \\ \frac{\partial G_2}{\partial y_n} & \frac{\partial G_2}{\partial h_n} & \frac{\partial G_2}{\partial y_{n-1}} & \frac{\partial G_2}{\partial h_{n-1}} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

where the partial derivatives in the first two rows are generally quite complicated functions. The symbolic algebra package Maple was used to evaluate $\rho(G'(v))$ at the fixed point, and it was found that the value varied with the parameters $\{\lambda, \mu, \tau\}$ in the test equation. This contrasts with the ODE case; the spectral radius in (2.4) remains constant over all test problems of the form (1.5). Hence, the stability of the DDE equilibrium state is not simply a characteristic of the algorithm itself, but also depends on the particular test equation. It is possible, however, to show that the stability is independent of the error tolerance, TOL. A proof is given at the end of this section.

We illustrate this analysis with some numerical examples. In these tests, and all others presented here, we used an adaptive DDE solver written in Matlab. The program was adapted from Matlab's built-in ode23.m ODE solver. The equation (1.10) was solved over $[0, 300]$ with $y(t) \equiv 1$ for $t \in [-\tau, 0]$. We used a safety factor of $\theta = 0.81$ and an error tolerance of $\mathrm{TOL} = 10^{-3}$.

Example 3.1. In this example, we take $\lambda = -2$, $\mu = 0.5$ and $\tau = 0.9$. The relevant values from (3.8) and (3.9) are $h_D = 1.03$ and $|y_D| = 5.2 \cdot 10^{-4}$, giving a spectral radius of 0.95. Fig. 1 plots the solution and stepsize values chosen by the code, with an asterisk (*) denoting a stepsize that led to a rejected step. We see the typical smooth behaviour associated with a stable equilibrium.

Example 3.2. We now choose $\lambda = -1$, $\mu = -0.5$ and $\tau = 0.5$. This gives $h_D = 1.5$ and $|y_D| = 3.6 \cdot 10^{-4}$ with a spectral radius of 1.03. The solution details are given in Fig. 2. In this example, the spectral radius is slightly bigger than 1 and we see that small amplifications about the equilibrium are slowly magnified until an eventual step rejection occurs.

We move on now to the general case, where $m > 1$ in (3.5). (This analysis also applies in the $m = 1$ case when the algorithm is regarded as implicit.) Imposing $h_n \equiv h_D$ and $y_n \equiv y_D$, we have $q(t_n - \tau) = q(t_{n+1} - \tau) = y_D$ and

$$k_1 = (\lambda + \mu) y_D, \qquad k_2 = (\lambda + \mu)(1 + h_D \lambda) y_D,$$

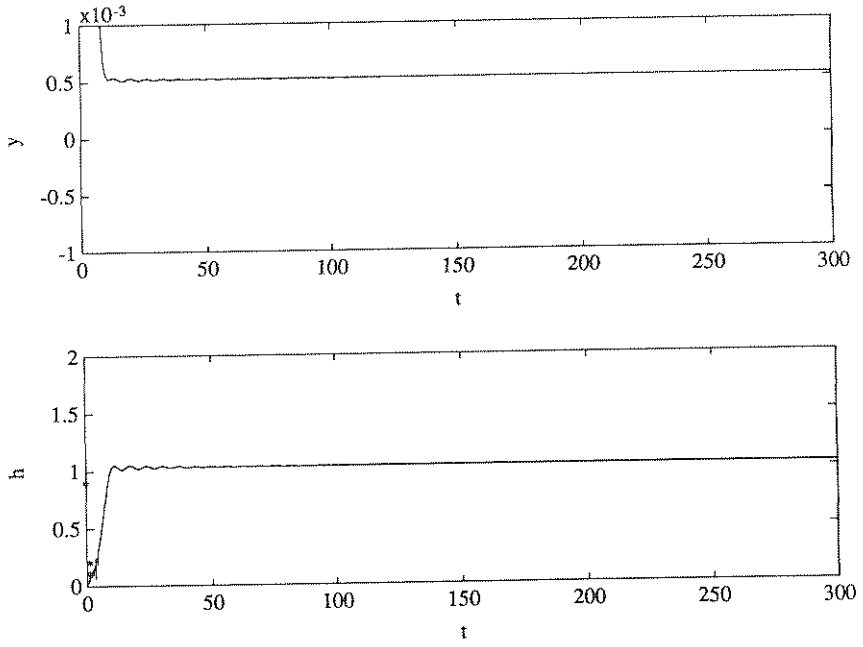

Fig. 1. Numerical solution and stepsizes for Example 3.1.

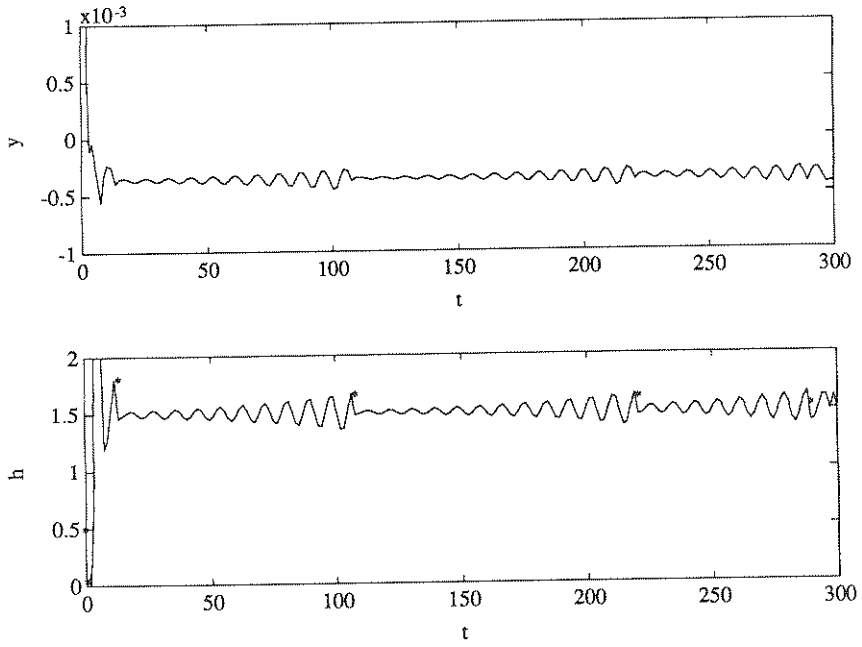

Fig. 2. Numerical solution and stepsizes for Example 3.2.

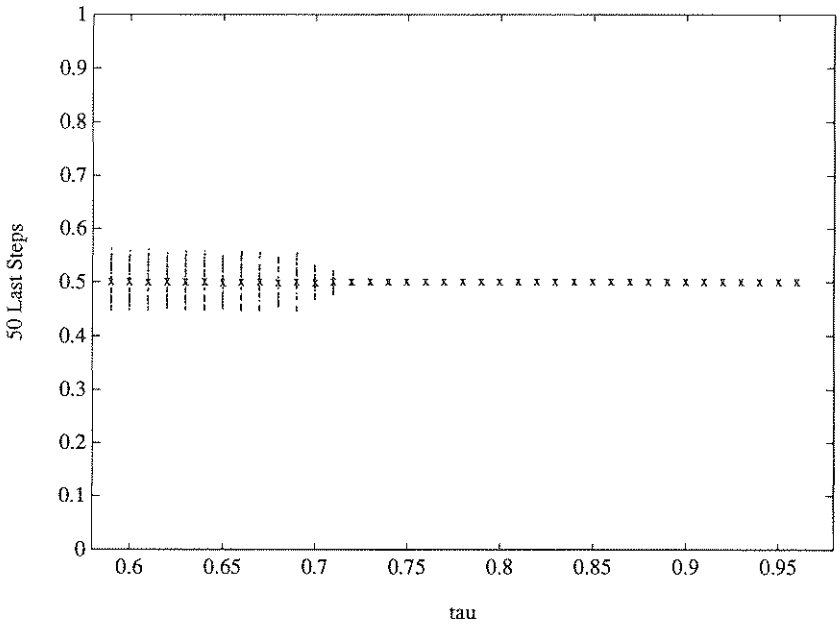

Fig. 3. Stepsizes for Example 3.3.

and the conditions for a period one solution reduce to

$$h_D = \frac{-2}{\lambda}, \qquad |y_D| = \frac{2\theta \mathrm{TOL}}{h_D^2(\lambda+\mu)\lambda}. \tag{3.10}$$

Note that $-2/\lambda$ is guaranteed to be positive (since we are assuming $\lambda < -|\mu|$) and hence (3.10) defines a valid equilibrium state provided that $m > 1$, which reduces to $\tau > -2/\lambda$. It follows that this $m > 1$ state exists whenever the $m = 1$ case derived earlier does not. We also point out that the value $-2/\lambda$ in (3.10) is precisely the stepsize limit that arises when Improved Euler is used on the ODE test equation (1.5). We will discuss this further in Section 6.

Example 3.3. This example illustrates the case $m = 2$. We set $\lambda = -4$, $\mu = -1.5$ and let τ vary over [0.59, 0.96] in steps of 0.01. In this case (3.10) gives $h_D = 0.5$. For each set of parameters we solve the corresponding test equation. Fig. 3 plots a sequence of dots representing the last 50 stepsizes used by the program. The symbol '$\times$' marks the average of the fifty values. Fig. 4 shows the value of the spectral radius of the Jacobian at the fixed point as τ varies. We see that in the range of τ where the spectral radius is bigger than 1, the stepsize sequence is not constant, but oscillates about the 0.5 level. When the spectral radius is below 1, the last 50 steps are visually indistinguishable from the value 0.5.

Example 3.4. Figs. 5 and 6 illustrate the case where $\lambda = -4$, $\tau = 0.95$ and μ varies between -3.6 and -1.45. Again, the spectral radius of the Jacobian at the fixed point determines the behaviour.

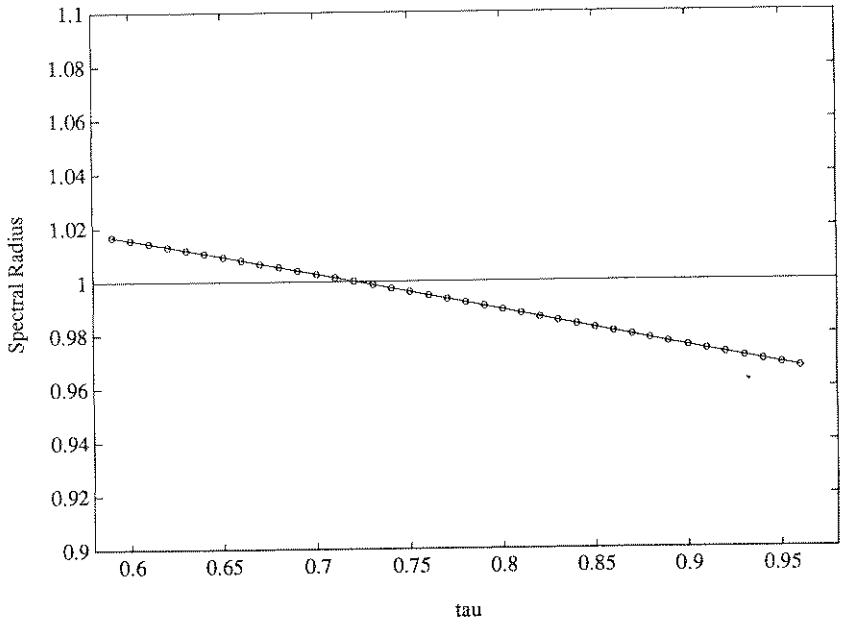

Fig. 4. Spectral radii for Example 3.3.

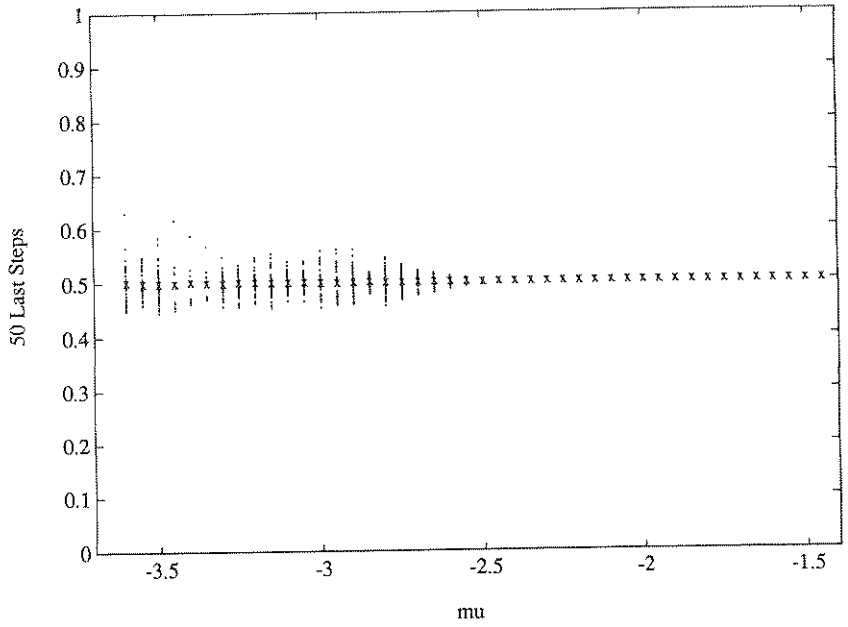

Fig. 5. Stepsizes for Example 3.4.

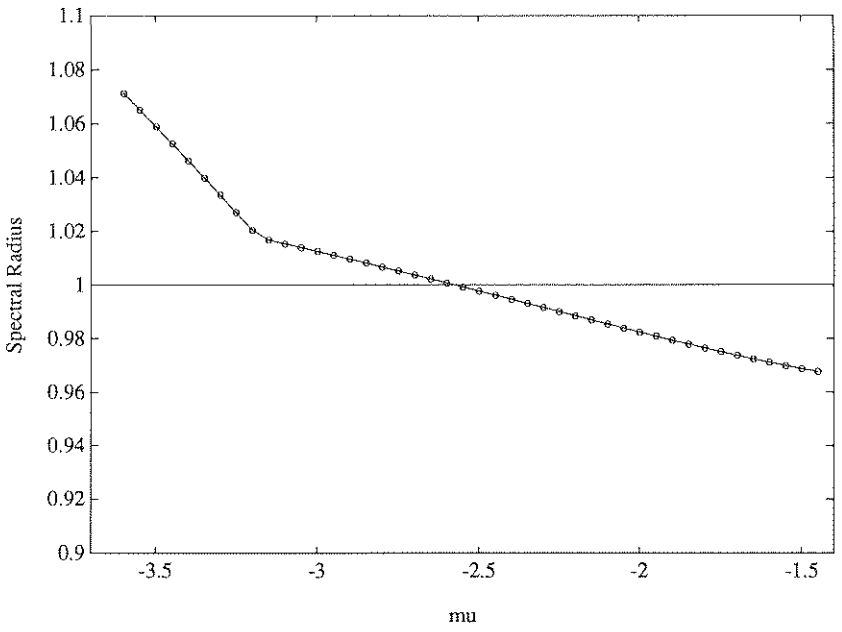

Fig. 6. Spectral radii for Example 3.4.

Examples 3.3 and 3.4 emphasise that the stability of the equilibrium state depends upon the parameters in the test equation. We conclude this section by showing that the stability does not depend upon the error tolerance. Hence, altering the value of TOL would not affect the long term behaviour, qualitatively.

Theorem 3.5. *The linearised stability of the equilibrium states defined in* (3.8), (3.9) *and in* (3.10) *is independent of the error tolerance*, TOL.

Proof. We give a proof for the general case $m \geqslant 3$. The cases $m = 1, 2$ can be handled in a similar way. We remark that the same style of proof was used in a slightly different context in [9, Theorem 3.2].

We first write the general recurrence and then examine the Jacobian at the fixed point. When the stepsize is constant we have $t_n - \tau \in [t_{n-m}, t_{n-m+1}]$ and $t_{n+1} - \tau \in [t_{n-m+1}, t_{n-m+2}]$, and the first two stages take the form

$$k_1 = \lambda y_n + \mu((1 - \sigma_{n-m}) y_{n-m} + \sigma_{n-m} y_{n-m+1}), \tag{3.11}$$

$$k_2 = \lambda(y_n + h_n k_1) + \mu((1 - \sigma_{n-m+1}) y_{n-m+1} + \sigma_{n-m+1} y_{n-m+2}), \tag{3.12}$$

where

$$\sigma_{n-m} = 1 + \frac{h_{n-m+1} + h_{n-m+2} + \cdots + h_{n-1} - \tau}{h_{n-m}},$$

$$\sigma_{n-m+1} = 1 + \frac{h_{n-m+2} + h_{n-m+3} + \cdots + h_n - \tau}{h_{n-m+1}}.$$

The iteration may then be written

$$\begin{bmatrix} y_{n+1} \\ h_{n+1} \\ y_n \\ h_n \\ \vdots \\ \vdots \\ y_{n-m+1} \\ h_{n-m+1} \end{bmatrix} = \begin{bmatrix} G_1([y_n, h_n, y_{n-1}, h_{n-1}, \ldots, y_{n-m}, h_{n-m}]^{\mathrm{T}}) \\ G_2([y_n, h_n, y_{n-1}, h_{n-1}, \ldots, y_{n-m}, h_{n-m}]^{\mathrm{T}}) \\ y_n \\ h_n \\ \vdots \\ \vdots \\ y_{n-m+1} \\ h_{n-m+1} \end{bmatrix}, \tag{3.13}$$

where

$$G_1([y_n, h_n, y_{n-1}, h_{n-1}, \ldots, y_{n-m}, h_{n-m}]^{\mathrm{T}}) = y_n + 0.5h_n(k_1 + k_2), \tag{3.14}$$

$$G_2([y_n, h_n, y_{n-1}, h_{n-1}, \ldots, y_{n-m}, h_{n-m}]^{\mathrm{T}}) = \left(\frac{\theta \mathrm{TOL}}{0.5h_n(k_2 - k_1)}\right)^{1/2} h_n. \tag{3.15}$$

(We assume, for definiteness, that $k_2 > k_1$ at the fixed point. If $k_2 < k_1$ then $k_2 - k_1$ should be replaced by $k_1 - k_2$ in (3.15), and the result below remains valid.)

Now, at the fixed point defined in (3.10), we see that h_{D} is independent of TOL and y_{D} depends linearly upon TOL. Taking the appropriate partial derivatives in (3.14) and (3.15), using (3.11) and (3.12), the dependence upon TOL of the first two rows of the Jacobian matrix at the fixed point may be expressed as

$$\begin{bmatrix} \text{ind.} & \propto \text{TOL} & \text{ind.} & \propto \text{TOL} & \cdots & \cdots & \text{ind.} & \propto \text{TOL} \\ \propto \text{TOL}^{-1} & \text{ind.} & \propto \text{TOL}^{-1} & \text{ind.} & \cdots & \cdots & \propto \text{TOL}^{-1} & \text{ind.} \end{bmatrix}. \tag{3.16}$$

Here, ind. denotes that the element is independent of TOL, with $\propto$ TOL and $\propto \text{TOL}^{-1}$ denoting linear and inverse linear dependence, respectively. The remaining rows of the Jacobian consist of the rows of the $2m \times 2m$ identity matrix, padded on the right by two zero elements. Letting $D = \mathrm{diag}(1, \text{TOL}, 1, \text{TOL}, \ldots, 1, \text{TOL})$, it follows that the similarity transformation $G' \to DG'D^{-1}$, which does not alter the eigenvalues, produces a matrix that is independent of TOL. Hence the result is proved. □

4. Improved Euler method with cubic Hermite interpolation

We now look at the effect of altering the interpolation formula. Suppose that the ODE solver described in the previous section uses the cubic Hermite interpolant (1.9). This interpolant requires first derivative approximations $\{f_n\}$, and we suppose that these are computed by evaluating the differential equation; so, from (1.7), $f_n := F(t_n, y_n, q(t_n - \tau))$.

Our approach is to look for an equilibrium state on the test equation (1.10) with $h_n \equiv h_D$, $y_n \equiv y_D$ and $f_n \equiv f_D$. We let m and σ be defined as in (3.5). Once again, the algorithm is not explicit when $m = 1$, since y_{n+1} and f_{n+1} are not available when $q(t)$ needs them. There are several ways of redefining the $m = 1$ algorithm to make it explicit, but since we are mainly concerned with the qualitative effect of changing the interpolant, we will assume that the implicit equations are solved when $m = 1$.

Under the assumption that h_n, y_n and f_n are constant, the cubic Hermite interpolant in (1.9) gives

$$q(t_n - \tau) = q(t_{n+1} - \tau) = [d_1(\sigma) + d_2(\sigma)]y_D + h_D[e_1(\sigma) + e_2(\sigma)]f_D = y_D + h_D p(\sigma) f_D, \quad (4.1)$$

where $p(\sigma) = \sigma(\sigma - 1)(2\sigma - 1)$. Since $f_n = \lambda y_n + \mu q(t_n - \tau)$, we have the relation $f_D = \lambda y_D + \mu[y_D + h_D p(\sigma) f_D]$, which gives

$$f_D = \frac{(\lambda + \mu) y_D}{1 - h_D \mu p(\sigma)}. \quad (4.2)$$

The two stages in (3.1) and (3.2) then simplify to

$$k_1 = f_D, \quad (4.3)$$

$$k_2 = (1 + h_D \lambda) f_D. \quad (4.4)$$

In order for the Improved Euler equation in (3.4) to reproduce a constant solution, we need $k_1 + k_2 = 0$. From (4.3) and (4.4) this reduces to

$$(2 + h_D \lambda) f_D = 0. \quad (4.5)$$

For a constant stepsize, we require $0.5 h_n |k_2 - k_1| = \theta \mathrm{TOL}$, which becomes

$$|y_D| = \frac{2\theta \mathrm{TOL} |1 - h_D \mu p(\sigma)|}{h_D^2 (\lambda + \mu) \lambda}. \quad (4.6)$$

It is clear that $h_D = -2/\lambda$ solves (4.5), and hence substituting this value into (4.6) and (4.2) we obtain an equilibrium state $\{h_D, y_D, f_D\}$. Once more, we observe that $-2/\lambda$ is also the stepsize limit for stability on the ODE (1.5). The stability of the equilibrium state $\{h_D, y_D, f_D\}$ can be determined by writing the iteration in the form $v_{n+1} = G(v_n)$, where $v_n = [y_n, h_n, y_{n-1}, h_{n-1}, f_{n-1}, \ldots, y_{n-m}, h_{n-m}, f_{n-m}]^T$, and examining the spectral radius of the Jacobian at the fixed point.

As in Section 3, it is possible to prove that the equilibrium stability is independent of TOL.

Theorem 4.1. *The linearised stability of the equilibrium state defined above is independent of the error tolerance*, TOL.

Proof. The result can be proved in a similar manner to Theorem 3.5. Here we briefly outline a proof for the case $m \geq 3$.

The Jacobian of the iteration has the following block structure

$$\begin{bmatrix} A_{2\times 2} & B_{2\times(3m-6)} & C_{2\times 3} & D_{2\times 3} \\ I_{2\times 2} & 0_{2\times(3m-6)} & 0_{2\times 3} & 0_{2\times 3} \\ E_{1\times 2} & F_{1\times(3m-6)} & G_{1\times 3} & H_{1\times 3} \\ 0_{(3m-6)\times 2} & I_{(3m-6)\times(3m-6)} & 0_{(3m-6)\times 3} & 0_{(3m-6)\times 3} \\ 0_{3\times 2} & 0_{3\times(3m-6)} & I_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \tag{4.7}$$

where the subscripts denote the dimensions of the blocks.

Noting that h_D is independent of TOL, while y_D and f_D depend linearly upon TOL, it can be shown that when the Jacobian is evaluated at the fixed point, the nontrivial blocks above have the following patterns of dependency:

$$A_{2\times 2} = \begin{bmatrix} \text{ind.} & \propto \text{TOL} \\ \propto \text{TOL}^{-1} & \text{ind.} \end{bmatrix},$$

$$B_{2\times(3m-6)} = \begin{bmatrix} \text{ind.} & \propto \text{TOL} & \text{ind.} & \cdots & \text{ind.} & \propto \text{TOL} & \text{ind.} \\ \propto \text{TOL}^{-1} & \text{ind.} & \propto \text{TOL}^{-1} & \cdots & \propto \text{TOL}^{-1} & \text{ind.} & \propto \text{TOL}^{-1} \end{bmatrix},$$

$$C_{2\times 3} = \begin{bmatrix} \text{ind.} & \propto \text{TOL} & \text{ind.} \\ \propto \text{TOL}^{-1} & \text{ind.} & \propto \text{TOL}^{-1} \end{bmatrix},$$

$$D_{2\times 3} = \begin{bmatrix} \text{ind.} & \propto \text{TOL} & \text{ind.} \\ \propto \text{TOL}^{-1} & \text{ind.} & \propto \text{TOL}^{-1} \end{bmatrix},$$

$$E_{1\times 2} = [\text{ind.} \ \propto \text{TOL}],$$

$$F_{1\times(3m-6)} = [\text{ind.} \ \propto \text{TOL} \ \text{ind.} \ \cdots \ \text{ind.} \ \propto \text{TOL} \ \text{ind.}],$$

$$G_{1\times 3} = [\text{ind.} \ \propto \text{TOL} \ \text{ind.}],$$

$$H_{1\times 3} = [\text{ind.} \ \propto \text{TOL} \ \text{ind.}].$$

It follows that with $D = \text{diag}(1, \text{TOL}, \underbrace{1, \text{TOL}, 1}, \underbrace{1, \text{TOL}, 1}, \ldots, \underbrace{1, \text{TOL}, 1})$, the similarity transformation $G' \to DG'D^{-1}$ produces a matrix that is independent of TOL. □

5. Euler method with linear Lagrange interpolation

Here, we take the algorithm described in Section 3 and alter the ODE formula. We use Euler's method (3.3) to advance the solution, keeping the same error estimate, $\text{est}_{n+1} = \| y^{\text{E}}_{n+1} - y^{\text{IE}}_{n+1} \|$, and the same interpolant (1.8).

Euler's method has stability polynomial $S(z) = 1 + z$. Hence, on the test ODE (1.5) the stability limit is given by $h_L\lambda = -2$, with $S(-2) = -1$. It follows that the corresponding equilibrium

state in Section 2 has period two. Hence, we look for an analogous period two solution to the recurrence on the DDE (1.10). Specifically, we set $h_n \equiv h_D$ and $y_{n+k} = (-1)^k y_D$, and we let m and σ be defined by (3.5). It then follows that the linear Lagrange interpolant gives $q(t_n - \tau) = -q(t_{n+1} - \tau) = (-1)^m y_D(1 - 2\sigma)$. Hence we have

$$k_1 = \lambda y_D + \mu(-1)^m y_D(1 - 2\sigma). \tag{5.1}$$

Now, for our period two solution, we require $y_n + h_n k_1 = -y_n$, which leads to

$$h_D = \frac{-2}{\lambda + (-1)^m \mu(1 - 2\sigma)}. \tag{5.2}$$

The second stage (which is needed only for the error estimate) becomes

$$k_2 = y_D(\lambda(1 + h_D\lambda) + (-1)^m \mu(1 - 2\sigma)(h_D\lambda - 1)), \tag{5.3}$$

and the condition for a constant stepsize, $0.5h_n|k_2 - k_1| = \theta\mathrm{TOL}$, forces

$$|y_D| = \frac{2\theta\mathrm{TOL}}{|(h_D\lambda)^2 + (-1)^m h_D\mu(1 - 2\sigma)(h_D\lambda - 2)|}. \tag{5.4}$$

Using the relation $mh_D = \tau + \sigma h_D$, we may eliminate σ from (5.2) to give

$$h_D = \frac{-2(1 + (-1)^m \mu\tau)}{\lambda + (-1)^m \mu(1 - 2m)}. \tag{5.5}$$

Eqs. (5.4) and (5.5) define a period two solution, provided that the condition $(m-1)h_D < \tau \leqslant mh_D$ holds.

We remark that the stepsize h_D and solution y_D derived in this section are completely different, in general, to those that arose in the previous two sections. In particular, the values here depend upon the parity of m. Furthermore, the stepsize h_D is generally different from the stepsize $h_L = -2/\lambda$ that arises when Euler's method is applied to the ODE (1.5). Note also that h_D is not necessarily smaller than h_L.

As described in Section 2 for the ODE case, the stability of the period two solution can be determined by writing the iteration as a map $v_{n+1} = G(v_n)$, where $v_n = [y_n, h_n, y_{n-1}, h_{n-1}, \ldots, y_{n-m}, h_{n-m}]^T$ and examining the spectral radius of the product of the Jacobian evaluated at the two points.

Example 5.1. Here we take $\lambda = -2$, $\mu = 1$ and $\tau = 0.5$. This gives $h_D = 1$ with $m = 1$ in (5.5), so that $|y_D| = 4.05 \cdot 10^{-4}$ in (5.4). The relevant spectral radius is 0.55, so the fixed point is highly stable. Fig. 7 plots the stepsizes and solution values produced by the code, and we see that the period two solution is quickly located.

Example 5.2. In this example, we have $\lambda = -3$, $\mu = 0.8$ and $\tau = 1$. Here, $h_D = 2/3$ with $m = 2$ in (5.5), and $|y_D| = 4.05 \cdot 10^{-4}$ in (5.4). In this case, the fixed point is very unstable as the relevant spectral radius is 1.63. We see from Fig. 8 that the solution and stepsize oscillate about the equilibrium values and many steps are rejected.

As in the previous two sections, the equilibrium states can be shown to be tolerance-independent.

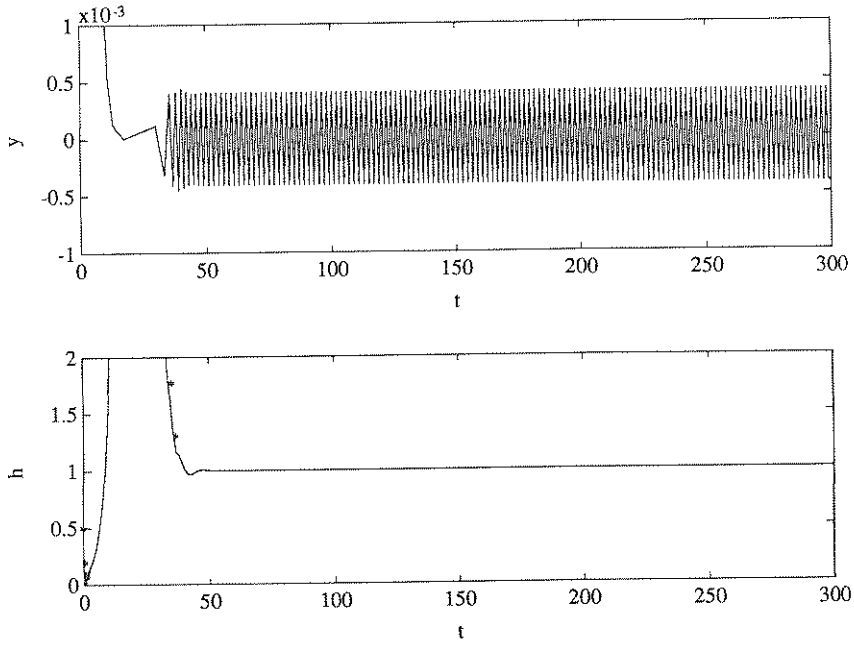

Fig. 7. Solution and stepsizes for Example 5.1.

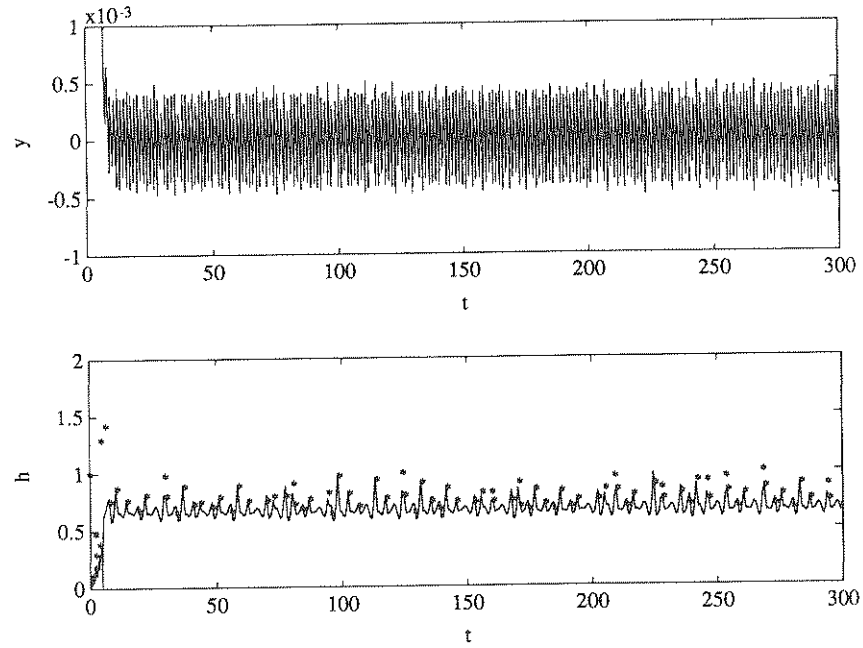

Fig. 8. Solution and stepsizes for Example 5.2.

Theorem 5.3. *The linearised stability of the equilibrium states defined above is independent of the error tolerance*, TOL.

Proof. Our outline proof follows those in the previous sections. The relevant Jacobian matrix has the block form

$$\begin{bmatrix} A_{4\times 2} & B_{4\times(2m-4)} & C_{4\times 4} \\ I_{2\times 2} & 0_{2\times(2m-4)} & 0_{2\times 4} \\ 0_{(2m-4)\times 2} & I_{(2m-4)\times(2m-4)} & 0_{(2m-4)\times 4} \end{bmatrix}. \tag{5.6}$$

By examining the dependency pattern of the first four rows, it is possible to find a diagonal similarity transformation $G' \to DG'D^{-1}$ that removes the dependence upon TOL. □

6. Discussion and extensions

Our numerical tests suggest that when a stable equilibrium state exists, it invariably attracts the numerical solution. Loosely, the error control ensures that the numerical solution approaches the true fixed point $y \equiv 0$, after which the linear attractivity becomes relevant. In the case where the equilibrium state is unstable, the numerical solution and stepsize oscillate about their equilibrium values. In this case, we have observed that it is possible for nonuniform fixed points with high period involving one or more rejected steps per period to arise.

Overall, we conclude that the equilibrium theory of Hall [5] for ODEs is also applicable to DDE algorithms. However, some key differences arise on moving from the ODE (1.5) to the DDE (1.10). In particular, for the algorithms studied here, the stability of the equilibrium state depends on the parameters in the test equation and also on the interpolation process. Hence, equilibrium state stability is not simply a characteristic that is inherited from the underlying ODE solver. This suggests that in order to guarantee smooth behaviour, an alternative mechanism for error control and stepsize selection must be used. Ideas from the area of control theory have recently been applied to ODE solvers [2, 3]; clearly an extension of this approach to the DDE case would be worthwhile.

The analysis here applies to the linear test equation (1.10). It can be argued, however, that, as for the ODE case, the results should be applicable to more general nonlinear equations, provided that linearisation about a steady state is valid. In particular, we performed a series of tests on the nonlinear equation

$$y'(t) = \lambda y(t) + \mu \frac{y(t-\tau)}{1 + (y(t-\tau))^n}, \tag{6.1}$$

which has been proposed as a model for blood-related diseases [4]. Here $\lambda < 0$, $\mu, \tau > 0$ and n is an even integer. Linearising about the steady state $y(t) \equiv 0$ produces the linear model (1.10). Experiments on (6.1) with $n = 10$ and with λ and μ chosen so that $y(t) \to 0$ as $n \to \infty$ gave results that were virtually identical to those on the corresponding linear problem.

It is clearly possible to extend the analysis presented here to more general Runge–Kutta based DDE algorithms. When high order RK formulae are used, interpolants with an appropriate order of accuracy must be chosen. The derivation of such interpolants has recently been an active area of research, and several choices are available; see, for example, [13, 15]. Our approach of looking for period one or two fixed points, where the stepsize is constant could be applied to such algorithms. However, it is not clear whether the existence of an equilibrium state can always be guaranteed. Further, high order interpolants must be based on several pieces of data, many located at off-step points, and it is not clear what conditions should be imposed in order to define an equilibrium state. There is much scope here for further work.

It is possible, however, to establish the existence of an equilibrium state for a general class of DDE algorithms, and we finish with this result. The theorem below shows that when Lagrange interpolation is used, an RK formula for which $S(h_L\lambda) = +1$ on the ODE stability boundary has a period one equilibrium state on (1.10) with $h_D = h_L$. This generalises the findings for the Improved Euler method in Section 3. Note that the result applies to RK formulae and interpolation schemes of any order.

Theorem 6.1. *If an explicit RK formula satisfies $S(h_L\lambda) = +1$ on the ODE stability boundary, and if the interpolant $q(t)$ is chosen to be a Lagrange polynomial that interpolates $\{y_n\}$ values, then the corresponding DDE algorithm applied to* (1.10) *has a period one fixed point with $h_D = h_L$ and y_n constant.*

Proof. If such a fixed point exists with, say, $y_n \equiv y_D$, then the corresponding Lagrange interpolant reduces to a constant function; that is, $q(t) \equiv y_D$. The RK formula will then be integrating a problem of the form

$$y'(t) = \lambda y(t) + \mu y_D. \tag{6.2}$$

It is easily verified that the RK formula applied to (6.2) produces

$$y_{n+1} = S(h_n\lambda)\, y_n + (S(h_n\lambda) - 1)\frac{\mu y_D}{\lambda}.$$

Hence with stepsize $h_n = h_L$, we get $y_{n+1} = y_n$. Now the error estimate, est_{n+1}, is a linear function of y_D. Hence we can always choose y_D to make $\mathrm{est}_{n+1} = \theta\mathrm{TOL}$. This ensures that the stepsize and solution remain constant from step to step. □

References

[1] W.H. Enright, Analysis of error control strategies for continuous Runge–Kutta methods, *SIAM J. Numer. Anal.* **26** (1989) 588–599.

[2] K. Gustafsson, Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods, *ACM Trans. Math. Software* **17** (1991) 533–554.

[3] K. Gustafsson, M. Lundh and G. Söderlind, A PI stepsize control for the numerical solution of ordinary differential equations, *BIT* **28** (1988) 270–287.

[4] J.K. Hale, Homoclinic orbits and chaos in delay equations, in: B.D. Sleeman and R.J. Jarvis, Eds., *Proc. Ninth Dundee Conf. on Ordinary and Partial Differential Equations* (Wiley, New York, 1986).
[5] G. Hall, Equilibrium states of Runge–Kutta schemes, *ACM Trans. Math. Software* **11** (1985) 289–301.
[6] G. Hall, Equilibrium states of Runge–Kutta schemes, Part II, *ACM Trans. Math. Software* **12** (1986) 183–192.
[7] G. Hall and D.J. Higham, Analysis of stepsize selection schemes for Runge–Kutta codes, *IMA J. Numer. Anal.* **8** (1988) 305–310.
[8] D.J. Higham, The tolerance proportionality of adaptive ODE solvers, *J. Comput. Appl. Math.* **45** (1993) 227–236.
[9] D.J. Higham, Runge–Kutta stability on a Floquet problem, *BIT* **34** (1994) 88–98.
[10] D.J. Higham and L.N. Trefethen, Stiffness of ODEs, *BIT* **33** (1993) 285–303.
[11] K.J. In't Hout, A new interpolation procedure for adapting Runge–Kutta methods to delay differential equations, *BIT* **32** (1992) 634–649.
[12] M.Z. Liu and M.N. Spijker, The stability of the θ-methods in the numerical solution of delay differential equations, *IMA J. Numer. Anal.* **10** (1990) 31–48.
[13] B. Owren and M. Zennaro, Derivation of efficient, continuous explicit Runge–Kutta methods, *SIAM J. Sci. Statist. Comput.* **13** (1992) 1488–1501.
[14] C.T.H. Baker and C.A.H. Paul, Computing stability regions—Runge–Kutta methods for delay differential equations, *IMA J. Numer. Anal.* **14** (1994) 347–362.
[15] J.H. Verner, Differentiable interpolants for high-order Runge–Kutta methods, *SIAM J. Numer. Anal.* **30** (1993) 1446–1466.
[16] M. Zennaro, P-stability properties of Runge–Kutta methods for delay differential equations, *Numer. Math.* **49** (1986) 305–318.