A FREE-STEPSIZE IMPLEMENTATION OF SECOND ORDER STABLE METHODS FOR NEUTRAL DELAY DIFFERENTIAL EQUATIONS

Alfredo Bellen [*,1] **Marino Zennaro** [*,1]

[*] *Dipartimento di Scienze Matematiche, Universita' di Trieste, Italy*

Keywords: Delay Differential Equations, Neutral Equation, Contractivity, Asymptotic Stability, Numerical Analysis.

It is known that the solution of the general Delay Differential Equation (DDE)

$$\begin{cases} y'(t) = f\big(t, y(t), y(t-\tau(t))\big), & t_0 \le t \le t_f, \\ y(t) = \phi(t), & t \le t_0, \end{cases}$$

where $\tau(t) \ge 0$, undergoes the *smoothing* effect, that is the solution, which is not differentiable at the point t_0 even if the functions f, τ and ϕ are of class C^∞, becomes smoother and smoother at subsequent points $\xi_0 = t_0 < \xi_1 < \dots$ (called *breaking points*) recursively generated as follows. If ξ_k is a breaking point where the function $y(t)$ is of class C^p, any ξ_j such that $\xi_k = \xi_j - \tau(\xi_j)$ is a subsequent breaking point where $y(t)$ is of class at least C^{p+1}. On the contrary, the smoothing effect does not apply to the DDE of *neutral* type (NDDE)

$$\begin{cases} y'(t) = f\big(t, y(t), y(t-\tau(t)), y'(t-\tau(t))\big), \\ \qquad\qquad\qquad\qquad t_0 \le t \le t_f, \\ y(t) = \phi(t), \quad t \le t_0, \end{cases}$$

where $y'(t)$ has a jump discontinuiuty at any breaking point. This prevents the convergence of any numerical step-by-step method to exceed the order one, unless the breaking points are all included as mesh points. This is a severe restriction, in particular for equations with many non monotonic delay functions $t - \tau(t)$ where the breaking points may proliferate in a chaotic manner as t increases.

Nevertheless, when the neutral equation may be transformed into the form (see Hale [4])

$$\begin{cases} \dfrac{d}{dt}\Big(y(t) - G\big(t, y(t-\tau(t))\big)\Big) = \\ H\Big(t, y(t), y(t-\tau(t))\big)\Big), \quad t_0 \le t \le t_f, \\ y(t) = \phi(t), \quad t \le t_0, \end{cases} \tag{1}$$

it is natural to solve it for the function

$$\Phi(t) = y(t) - G\big(t, y(t-\tau(t))\big)$$

that, unlike $y(t)$, is of class $C^1[t_0, t_f]$ provided that the function $H(t, u, v)$ is continuous, as usually occurs. This is the case, for instance, of the linear equation

$$y'(t) = Ly(t) + My(t-\tau) + Ny'(t-\tau), \tag{2}$$

where the delay τ is constant, which may be written as

$$\frac{d}{dt}\Big(y(t) - Ny(t-\tau)\Big) = L\Big(y(t) - Ny(t-\tau)\Big) + (M + LN)y(t-\tau).$$

In these cases the equation (1) takes the form of a Delay Differential Algebraic Equation (DDAE)

$$\begin{cases} \Phi'(t) = K\Big(t, \Phi(t), y\big(t-\tau(t)\big)\Big), \\ y(t) = \Phi(t) + G\Big(t, y\big(t-\tau(t)\big)\Big), \\ \qquad\qquad\qquad t_0 \le t \le t_f, \\ \Phi(t_0) = \phi(t_0) - G\Big(t_0, \phi\big(t_0 - \tau(t_0)\big)\Big), \\ y(t) = \phi(t), \quad t \le t_0, \end{cases} \tag{3}$$

[1] Partially supported by MURST and INdAM-GNCS

to be solved for $\Phi(t)$ and $y(t)$, where

$$K(t,u,v) = H\Big(t, u + G(t,v), v\Big).$$

For non vanishing delays, equation (3) is basically an ODE and may be solved step-by-step by a continuous method. In fact, for sufficiently small stepsizes, the algebraic component y may be separately computed, by recursion, after the differential component Φ .

Once we have integrated equation (3) till the point t_n by a *continuous* numerical method, the subsequent step for approximating Φ and y consists in solving, for $\Lambda(t)$ and $w(t)$, the local problem

$$\begin{cases} \Lambda'(t) = K\Big(t, \Lambda(t), \eta\big(t-\tau(t)\big)\Big), \\ w(t) = \Lambda(t) + G\Big(t, \eta\big(t-\tau(t)\big)\Big), \\ \qquad\qquad\qquad t_n \le t \le t_{n+1}, \\ \Lambda(t_n) = \eta(t_n) - G\Big(t_n, \eta\big(t_n - \tau(t_n)\big)\Big). \end{cases} \tag{4}$$

where $\eta(t)$ is the numerical continuous approximation of $y(t)$ previously computed in the past intervals. The system (4) is first solved for $\Lambda(t)$ with some continuous approximation $\lambda(t)$ and then for $w(t)$ with continuous approximation $\eta(t)$ given by

$$\eta(t) = \lambda(t) + G\Big(t, \eta\big(t-\tau(t)\big)\Big)$$

or, in order to avoid tracing back the solution until $\phi(t)$, by

$$\eta(t) = \mathcal{P}\Big(\lambda(\cdot) + G\Big(\cdot, \eta\big(\,\cdot - \tau(\cdot)\big)\Big)\Big)(t), \tag{5}$$

where $\mathcal{P}\big(u(\cdot)\big)$ is the interpolation of u in a suitable polynomial space and nodes in $[t_n, t_{n+1}]$.

Since the right-hand side function of the differential equation that defines the auxiliary function $\Phi(t)$ is Lipschitz continuous, any method of (global) uniform order $p \ge 2$ has a local error which is an $O(h^2)$ on any step interval including a discontinuity point, and an $O(h^{p+1})$ elsewhere (see [1], [3]). Assuming that the number of discontinuties is finite in $[t_0, t_f]$, we conclude that the method performs to the order 2 with free stepsizes, not necessarily including the discontinuity points into the mesh Δ.

Nevertheless, there still are some troubles with discontinuities for the continuous approximation $\eta(t)$ of the solution $y(t)$. In fact, if the interpolation (5) is used for the computation of $G\big(t, \eta(t - \tau(t))\big)$ in (4), the lack of smoothness of y at the discontinuity points causes a local loss of accuracy to the interpolation process down to an $O(h)$ whenever the delayed argument $t - \tau(t)$ crosses one of such points. However, this unpleasant phenomenon takes place only locally at those mesh points close to the discontinuity points, unless they are all included in the mesh Δ, in which case no loss of accuracy occurs. The global order of accuracy of $\eta(t)$ at all the other mesh points remains an $O(h^2)$.

We conclude this part by stressing the remarkable and, at the first sight, surprising property that, by handling the NDDE in the form (1), the loss of accuracy at some nodal values y_{n_i} does not affect the accuracy at subsequent points of the approximate trajectory where the expected accuracy is recovered. On the other hand, if interpolation is not used and $\eta(t)$ is computed by tracing it back until the initial function $\phi(t)$, not even these local losses of accuracy occur.

The occurencies mentioned above are illustrated by the following example.

Example Let us consider the equation

$$\begin{cases} y'(t) = y(t) + y'(t-1), \quad 0 \le t \le 3, \\ y(t) = 1, \quad t \le 0, \end{cases} \tag{6}$$

whose solution is the continuous function

$$y(t) = \begin{cases} e^t & \text{in } [0,1] \\ e^t + (t-1)e^{t-1} & \text{in } [1,2] \\ e^t + e^{t-1} + \dfrac{1}{2}(t-2)(t+2e)e^{t-2} & \text{in } [2,3] \end{cases}$$

We first apply the Heun method of order 2 to the equation in the form (6) with linear interpolation and constant stepsize $h = 3/N$, where N is the number of steps used to cover $[0,3]$. In Table 1 we display the results obtained by including the discontinuity points $\xi_1 = 1$ and $\xi_2 = 2$ in the mesh, whereas in Table 2 we display the results obtained by excluding them from the mesh. The first point $t_0 = 0$ and the final point $t_f = 3$ (which is a discontinuity point) are always included in the mesh.

We denote the continuous numerical solution by $\eta(t)$ and the nodal values by $y_n = \eta(t_n)$.

As expected, in Table 1 the maximum global order is an $O(h^2)$, whereas in Table 2 it is an $O(h)$.

N	$\max_{1\le n\le N} \lvert y_n - y(t_n)\rvert$
30	$2.48 \cdot 10^{-1}$
300	$2.69 \cdot 10^{-3}$
3000	$2.71 \cdot 10^{-5}$

Table 1. Numerical results for equation (6) with meshes including $\xi_1 = 1$ and $\xi_2 = 2$.

N	$\max_{1\leq n\leq N} \lvert y_n - y(t_n)\rvert$
31	$4.01 \cdot 10^{-1}$
301	$2.35 \cdot 10^{-2}$
3001	$2.16 \cdot 10^{-3}$

Table 2. Numerical results for equation (6) without including $\xi_1 = 1$ and $\xi_2 = 2$ in the mesh.

Now let us rewrite equation (6) in the form (3), that is

$$\begin{cases} \Phi'(t) = \Phi(t) + y(t-1), \quad 0 \leq t \leq 3, \\ y(t) = \Phi(t) + y(t-1), \\ \Phi(0) = 0, \\ y(t) = 1, \quad t \leq 0. \end{cases} \tag{7}$$

The auxiliary function turns out to be

$$\Phi(t) = \begin{cases} e^t - 1 & \text{in } [0,1] \\ e^t + (t-2)e^{t-1} & \text{in } [1,2] \\ e^t + \dfrac{1}{2}(t-2)(t+2e-2)e^{t-2} & \text{in } [2,3] \end{cases}$$

We apply again the Heun method with linear interpolation and constant stepsize $h = 3/N$ to approximate the auxiliary function $\Phi(t)$, and we use linear interpolation of nodal values to compute the numerical continuous solution $\eta(t)$ from the recursive formula which defines $y(t)$. We use the same meshes as before and report the relevant results in Table 3 and Table 4 where the nodal approximations of the auxiliary function Φ are denoted by Φ_n . We call n_1 and n_2 the indices of the two mesh points at which the loss of accuracy occurs, due to the interpolation process. The mesh point t_{n_1} is close to $\xi_2 = 2$, whereas $t_{n_2} = t_N = 3$. This is illustrated in Figure **??**.

Moreover, in Table 5 we display the results obtained avoiding interpolation for the recursive formula and not including the discontinuities $\xi_1 = 1$ and $\xi_2 = 2$ in the mesh.

N	$\max\limits_{1\leq n\leq N} \lvert y_n - y(t_n)\rvert$	$\max\limits_{1\leq n\leq N} \lvert \Phi_n - \Phi(t_n)\rvert$
30	$1.90 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$
300	$2.05 \cdot 10^{-3}$	$1.70 \cdot 10^{-3}$
3000	$2.07 \cdot 10^{-5}$	$1.71 \cdot 10^{-5}$

Table 3. Numerical results for equation (7) with meshes including $\xi_1 = 1$ and $\xi_2 = 2$.

N	$\max\limits_{1\leq n\leq N} \lvert y_n - y(t_n)\rvert$	$\max\limits_{n\neq n_1,n_2} \lvert y_n - y(t_n)\rvert$
31	$9.46 \cdot 10^{-2}$	$9.46 \cdot 10^{-2}$
301	$1.69 \cdot 10^{-3}$	$1.30 \cdot 10^{-3}$
3001	$2.83 \cdot 10^{-4}$	$1.34 \cdot 10^{-5}$

N	$\max_{1\leq n\leq N} \lvert \Phi_n - \Phi(t_n)\rvert$
31	$1.04 \cdot 10^{-1}$
301	$1.22 \cdot 10^{-3}$
3001	$1.25 \cdot 10^{-5}$

Table 4. Numerical results for equation (7) without including $\xi_1 = 1$ and $\xi_2 = 2$ in the mesh and with interpolation.

N	$\max\limits_{1\leq n\leq N} \lvert y_n - y(t_n)\rvert$	$\max\limits_{1\leq n\leq N} \lvert \Phi_n - \Phi(t_n)\rvert$
31	$1.18 \cdot 10^{-1}$	$1.11 \cdot 10^{-1}$
301	$1.39 \cdot 10^{-3}$	$1.29 \cdot 10^{-3}$
3001	$1.43 \cdot 10^{-5}$	$1.31 \cdot 10^{-5}$

Table 5. Numerical results for equation (7) without including $\xi_1 = 1$ and $\xi_2 = 2$ in the mesh and without interpolation.

All the results are consistent with our expectations.

Representation (3) also allows to perform a delay-independent stability analysis for the solution $y(t)$ as well as for the numerical approximation $\eta(t)$ (see Torelli and Vermiglio [3]). In particular, by assuming $\mu[L] < 0$, one can prove that, for the linear equation (2), the condition

$$\frac{\|LN + M\|}{-\mu[L]} + \|N\| \leq 1$$

implies the boundedness of the solution, while the stronger condition

$$\frac{\|LN(s) + M\|}{-\mu[L]} + \|N\| \leq \rho < 1$$

implies the asymptotic stability. The term $\mu[L]$, called *the logarithmic norm* of L, depends on the

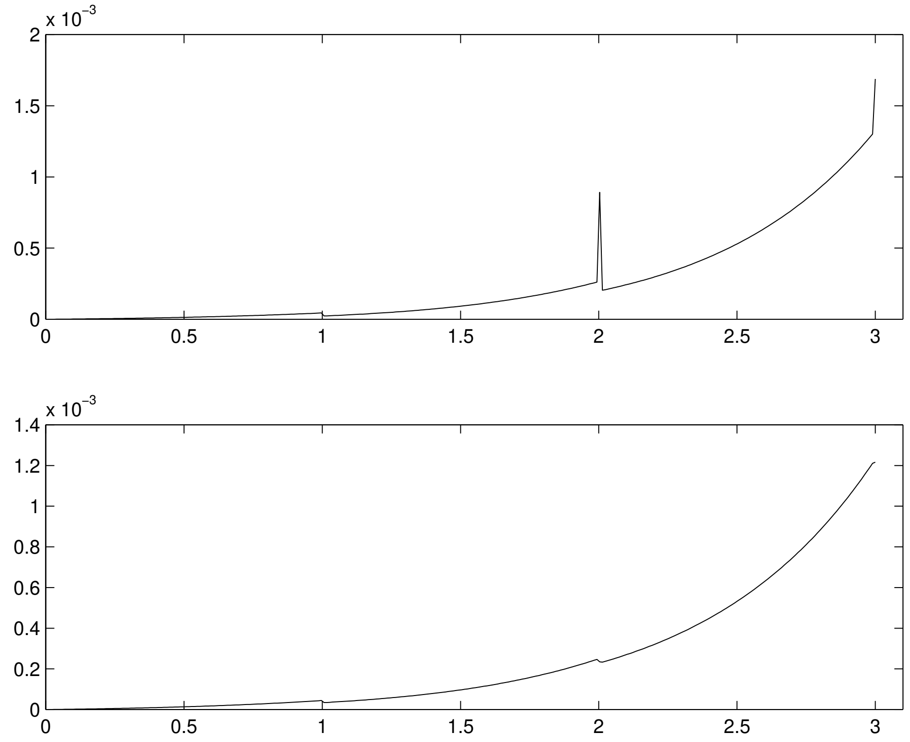

Fig. 1. Errors for the approximations of y (top) and Φ (bottom) with $N = 301$ for equation (7) by using interpolation (see Table 4).

norm involved and, for the inner product norm $\| \cdot \| = \sqrt{\langle \cdot | \cdot \rangle}$, it is

$$\mu[L] = \frac{\|L + L^T\|}{2}.$$

The approach is based on viewing the DDE as an ODE with forcing term and naturally extends to any kind of delay τ, even "state-dependent" $\tau = \tau(t, y(t))$. The same arguments used for the stability analysis of the solution $y(t)$ apply to the numerical approximation $\eta(t)$ and allow to select some second order continuous ODE-methods which preserve the asymptotic stability for any constant integration stepsize. A prominent method in this class is the two-stage Lobatto III-C method with linear interpolation (see Bellen and Zennaro [2]).

REFERENCES

[1] A. Bellen and M. Zennaro: *Numerical Methods for Delay Differential Equations*, Oxford University Press (in preparation).

[2] A. Bellen and M. Zennaro: *Strong contractivity properies of numerical methods for ordinary and delay differential equations*, Appl. Numer. Math. 9 (1992), 321-346.

[3] L. Torelli and R. Vermiglio: *Stability analysis of neutral delay differential equations in Hale's form*, Technical report RR UDMI/04/98, University of Udine, 1998

[4] J.K. Hale and S.M. Verduyn-Lunel: *Introduction to functional differential equations*, (1991), Springer Verlag, New York.