Report on Numerical Approximations of FDE's with Method of Steps

Simon Lacoste-Julien

May 30, 2001

Abstract

This is a summary of a meeting I had with Hans Vangheluwe on Thursday May 24 about numerical methods to solve *Functional Differential Equations* (FDE). I describe some observations I've made about them and I give a proof for an upper bound on the global error made when solving a DDE by the method of steps with a numerical scheme with known upper bound for an ODE.

Introduction

If a DDE has no delay between 0 and some $r > 0$, then, using the initial function, the DDE becomes an ODE on an interval of length r and we can solve it using normal numerical methods for ODE. Then, using this solution as a new initial function, we can repeat this procedure on another step of length r, and so on, like the usual method of steps. The question now is what are the conditions needed so that this method converges? Some are presented in section 2. Some observations pertinent to numerical methods are given in section 1.

1 Observations

1.1 Convergence

Euler method with linear interpolation is proved to converge for any Volterra functional DE[1] (where DDE is a special case of) in [3]. Unfortunately, it doesn't say at which rate the method converges. Very abstract topological notions (Banach spaces, etc.) and powerful tools of functional analysis are used in this article, but it is a little hard to understand it for now... Almost all articles I've

[1] A VFDE is:

$$y'(t) = F(y,t), \quad t \in [t_0, \beta]; \qquad y(t) = \phi(t), \quad t \in [\alpha, t_0]$$

where $F : \mathcal{C}([\alpha,\beta] \to E^n) \times E^n$ is a Volterra Functional, that is, $F(y,t)$ depends on t and on $y(s)$ for $s \in [\alpha, t]$, but is **independent** of $y(s)$ for $s > t$; and $\phi \in \mathcal{C}([\alpha,\beta] \to E^n)$ is a specified initial function.

read which treated of numerical methods made reference to this article, so it seems to be a strong reference in the domain.

El'gol'ts writes in [5] about approximate methods for the integration of differential equations with deviating arguments:

> The convergence of these constructed approximate methods to the exact solution for the steps approaching to zero, and error bounds may be completely obtained *as for equations without deviating arguments*; therefore, it is scarcely required to go into explicit details and is more expedient only to stress some peculiarities arising from the application of these methods to equations with a deviating argument. (p. 235)

This could explain why I haven't seen much developments about error bounds for numerical solution to DDE's in the articles I have browsed. The only explicit descriptions were developed again in the abstract framework of topological vector space (for example, see [2]). Still, this should be investigated.

1.2 Smoothness

Contrary to ODE's, the fact that F is C^∞ doesn't imply that the solution to a DDE will be smooth. For example, consider the simple C^∞ DDE with constant delay:

$$y'(t) = y(t-1), \quad t \in [0, \infty[; \qquad y(t) = 1, \quad t \in [-1, 0]$$

This has the unique solution on $[-1, 1]$:

$$y = \begin{cases} 1 & \text{for } t \leq 0 \\ 1 + t & \text{for } t \geq 0 \end{cases}$$

which doesn't even have a continuous first derivative at $t = 0$! As noted in [5, p. 9], this will happen every time the initial function doesn't satisfy the DDE at $t = t_0$. But using the methods of steps, we can observe that the solution does get smoother as the steps advance; and more precisely, in the case of one constant delay, that the n^{th} derivative of the solution is continuous at the $(\mathrm{n}+1)^{\mathrm{th}}$ step. I don't know yet the impact of those discontinuities on the convergence of numerical methods, but it seems from [6] that the numerical scheme needs to track down those discontinuities to be successful. For example, we couldn't use in general a n^{th} order numerical method which assumes that the solution to the DDE is C^n. But we could instead use lower order methods, until the n^{th} step, where the solution will become smooth enough for our numerical scheme.

1.3 Interpolation

When using numerical methods to solve a DDE with variable delays, we often need points that are in between points we have already computed. We thus normally use interpolation also. The question that could arise is: what is its effect on the order of convergence of the method? I observed that using linear

interpolation on Euler's method (which is linear) didn't alter its order of convergence. Similarly, I think we can conjecture that using an interpolation method of order at least as high as the one of our numerical method to solve the ODE won't change its order of convergence. But this also needs to be investigated.

2 My Bound

We consider a DDE with one delay, for simplicity:

$$y'(t) = f\big(t, y(t), y(t-\tau(t))\big), \quad t - \tau(t) \geq \alpha \quad \forall t \in [t_0, \beta]$$

$$y(t) = \phi(t) \text{ on } [\alpha, t_0] \tag{1}$$

We assume further that $\tau(t) \geq r > 0$ so that we can apply the method of steps, that f is continuous and globally Lipschitzian on $\mathbb{R}^2$ (for its last 2 variables) with Lipschitz constant L, and that τ is continuous, so that a unique solution which depends continuously (in the sup norm) on the initial function ϕ exists on the entire interval $[\alpha, \beta]$ (the proof of this assertion can be found in [4]).

Now, we consider what happens when applying the method of steps with a numerical scheme of $O(g(h))$, that is, the *global error* ***taking in consideration interpolation between the points we compute*** is smaller than $c \cdot g(h)$ for some constant c, where h is the biggest step we have used in the numerical method. Usually, some smoothness conditions are needed for those errors bounds. (For example, convergence of Euler method with $O(h)$ is guaranteed if the solution is at least C^2, so that we can apply Taylor's theorem of order 1) From the comment in section 1.2, we see that we'll then have to impose some stronger conditions on the initial function to be able to use those error bounds. Hence, we'll also need the following assumptions:

1. f, τ and ϕ are smooth enough so that we can use those error bounds (being C^∞ could be nice...)

2. The initial function ϕ needs to satisfy as many times the DDE as needed at t_0 so that the solution becomes smooth enough for our purpose (see section 1.2). For example, to use Euler's method bound, we need that ϕ is C^2 and that:

   - $\phi'(t_0) = f\big(t_0, \phi(t_0), \phi(t_0 - \tau(t_0))\big)$
   - $\phi''(t_0) = f_1 + f_2 \cdot \phi'(t_0) + f_3 \cdot \phi'(t_0 - \tau(t_0)) \cdot (1 - \tau'(t_0))$
     Where f_1 denotes the partial derivative of f with respect to its first variable. (We got this condition by implicitly differentiating the DDE with respect to t.)

   This will ensure that the solution is C^2 everywhere.

3. The interpolation method used in our numerical method will make the solution smooth enough to use our error bounds. (for example, using cubic spline interpolation would be enough to use Euler's method since it yields a function which is C^2)

I'm not completely sure for now if all these assumptions are enough for the proof that will follow. The crucial point is: will I be justified to use the global error bound given for the ODE? Since I'm giving a summary of what I had presented to Hans Vangheluwe, I will leave as it is, for now. But it would need more justifications...

So here we go:

step 1

Given the initial function ϕ, since $\tau(t) \geq r$, to obtain the solution of equation (1), we can simply solve the ODE:

$$y'(t) = f\big(t, y(t), \phi(t - \tau(t))\big) \quad \text{for } t \in [t_0, t_0 + r] \tag{2}$$

with our numerical scheme of order $g(h)$. We call w the obtained solution on $[t_0, t_0 + r]$ and y the true solution.

step 2

Now, we let:

- w be the numerical solution (in continuation of w from step 1) of:

$$w'(t) = f\big(t, w(t), w(t - \tau(t))\big) \quad \text{for } t \in [t_0 + r, t_0 + 2r]; \tag{3}$$

- x be the true solution of $w'(t) = f\big(t, w(t), w(t-\tau(t))\big)$ for $t \in [t_0+r, t_0+2r]$;
- y be the true solution of $y'(t) = f\big(t, y(t), y(t-\tau(t))\big)$ for $t \in [t_0+r, t_0+2r]$.

Hence, we see that y is the true solution, while x is a theoretical solution obtained to an approximated problem and w is a doubly approximated solution (one approximation from the numerical scheme, another approximation from the ODE using the numerical solution on $[t_0, t_0 + r]$).

From our error bound on our numerical method (and assuming our assumptions of smoothness are strong enough to use it), we know that

$$|w(t) - x(t)| \leq c \cdot g(h) \quad \text{for } t \in [t_0 + r, t_0 + 2r] \tag{4}$$

And since everything is continuous (the weakest condition of smoothness), we can integrate (2) and (3) to get x and y respectively:

$$x(t) = w(t_0 + r) + \int_{t_0+r}^{t} f\big(s, x(s), w(s - \tau(s))\big) ds$$

$$y(t) = y(t_0 + r) + \int_{t_0+r}^{t} f\big(s, y(s), y(s-\tau(s))\big)ds$$

for $t \in [t_0+r, t_0+2r]$. Subtracting, taking the absolute value, using the triangle inequality and the Lipschitz condition on f (this explains the Lipschitz constant L), we get:

$$|x(t)-y(t)| \leq |w(t_0+r)-y(t_0+r)| + \int_{t_0+r}^{t} L\cdot\max\big\{|x(s)-y(s)|, |w(s-\tau(s))-y(s-\tau(s))|\big\}ds$$

Now, we need to play finely with the inequalities... Since for $t \leq t_0 + r$, we have that w is the numerical solution of (2) and y the true solution, we have that

$$|w(s-\tau(s)) - y(s-\tau(s))| \leq c\cdot g(h)) \quad \text{for } s \in [t_0+r, t_0+2r]$$

since $\tau(s) \geq r$. Hence, we get:

$$|x(t) - y(t)| \leq c\cdot g(h) + \int_{t_0+r}^{t} L \cdot \max\big\{|x(s) - y(s)|, c\cdot g(h)\big\}ds$$

Now the delicate reasoning: everything on the right hand side is positive, thus we can get the maximum value for $|x - y|$ in the integral inequality by putting the left hand side *equal* to the right hand side (try to convince yourself that this is true!). Identifying $|x(t) - y(t)|$ by $z(t)$, we get:

$$z(t) = c\cdot g(h) + \int_{t_0+r}^{t} L \cdot \max\big\{z(s), c\cdot g(h)\big\}ds$$

From this, we can see that $\max\big\{z(s), c\cdot g(h)\big\} = z(s)$, hence, we obtain a simple integral equation which can be easily transformed into an equivalent differential equation (since everything is continuous) which can be solved with an integrating factor. Sparing you the details, we obtain:

$$z(t) = c\cdot g(h)e^{L(t-t_0-r)} \tag{5}$$

This is in fact an indirect application of Gronwall-Reid lemma[2]. And thus, a genuine (maybe not very tight) upper bound for the global error to (1) on $[t_0 + r, t_0 + 2r]$ can be obtained using (4) and (5) and the triangle inequality:

$$|w - y| \leq |w - x| + |x - y| \leq c\cdot g(h) + c\cdot g(h)e^{Lr}$$

so

$$|w(t) - y(t)| \leq c\cdot g(h)(1 + e^{Lr}) \quad \text{for } t \leq t_0 + 2r \tag{6}$$

[2] Gronwall-Reid is used extensively in the theory of differential equations. It is proven in [4, p. 72].

Reid's Lemma. *Let C be a given constant and k a given positive continuous function on an interval J. Let $t_0 \in J$. Then if $v\colon J \to [0, \infty[$ is continuous and*

$$v(t) \leq C + \left|\int_{t_0}^{t} k(s)v(s)ds\right| \quad \forall t \text{ in } J,$$

it follows that

$$v(t) \leq Ce^{\left|\int_{t_0}^{t} k(s)ds\right|} \quad \forall t \text{ in } J.$$

step 3

Using the same notation, we find similarly for $t \in [t_0 + 2r, t_0 + 3r]$:

$$|x(t)-y(t)| \leq \underbrace{|w(t_0+2r) - y(t_0+2r)|}_{\leq\, cg(h)(1+e^{Lr})} + \int_{t_0+2r}^{t} L{\cdot}\max\left\{|x(s)-y(s)|, \underbrace{|w(s-\tau(s)) - y(s-\tau(s))|}_{\leq\, cg(h)(1+e^{Lr})}\right\} ds$$

where the bounds under the braces were found using (6). Using a very similar reasoning than in step 2, we get:

$$\begin{aligned} |w - y| \leq |w - x| + |x - y| &\leq c{\cdot}g(h) + c{\cdot}g(h)(1 + e^{Lr})e^{Lr} \\ &= c{\cdot}g(h)(1 + e^{Lr} + e^{2Lr}) \end{aligned}$$

With an easy induction, we can thus find that at the nth step,

$$|w(t) - y(t)| \leq c{\cdot}g(h)(1 + e^{Lr} + e^{2Lr} + \cdots + e^{(n-1)Lr}) \quad \text{for } t \leq t_0 + nr$$

So that an upper bound for the global error to (1) on an interval $[t_0, \beta]$ of length $\leq Mr$ is

$$error \leq c{\cdot}g(h)\frac{e^{MLr} - 1}{e^{Lr} - 1} \leq D{\cdot}g(h)$$

where D is a constant *independent* of the numerical method and of h. So **we see that the *order* of the numerical method doesn't change from ODE to DDE!** We thus obtain, as a corollary, the convergence of this method on a finite interval since, normally, $g(h) \to 0$ as $h \to 0$. This is quite surprising, so that the smoothness condition should be carefully verified again...

3 What's Next

The consequences of the smoothness condition should be investigated. For example, I should find an easy example where the fact that the solution is not smooth at $t{=}t_0$ makes the Euler method not to converge...

From my discussions with Hans Vangheluwe, two paths are now opened to me. First of all, in a more pragmatic way, I could investigate how we could use local truncation error to find error bounds for DDE and thus implement adaptive step size algorithms (with variable r, for example). On the other hand, I could dive into the more theoretical framework of functional analysis to look at all the abstract tools that have been developed to solve numerically those functional differential equations. In particular, I'm thinking of [1].

References

[1] Chartres, Bruce A. and Robert S. Stepleman, *A General Theory of Convergence for Numerical Methods*, SIAM J. Numer. Anal. 9 (1972), pp. 476–492.

[2] ———, *Order of Convergence of Linear Multistep Methods for Functional Differential Equations*, SIAM J. Numer. Anal. 12 (1975), pp. 875–886.

[3] Cryer, Colin W. and Lucio Tavernini, *The Numerical Solution of Volterra Functional Differential Equations by Euler's Method*, SIAM J. Numer. Anal., 9 (1972), pp. 105–129.

[4] Driver, R. D., *Ordinary and Delay Differential Equations*, Springer-Verlag, New York, 1977, QA1.A647.

[5] El'sgol'ts, L.E. and S. B. Norkin, *Introduction to the Theory of Differential Equations with Deviating Arguments*, New York, Academics Press, 1973, QA371.E3813.1971.

[6] de Gee, Maarteen, *Linear Multistep Methods for Functional Differential Equations*, Mathematics of Computation, 12 (1975), pp. 876–886.