On the automatic computation of high order variationals for the numerical integration of ODE by means of the Taylor method

Àngel Jorba
Departament de Matemàtica Aplicada i Anàlisi
Universitat de Barcelona,
Gran Via 585, 08007 Barcelona, Spain

In this presentation we will revisit one of the oldest numerical procedures for the numerical integration of ODEs, the Taylor method. Consider the initial value problem

$$\left\{ \begin{array}{rcl} x'(t) & = & f(t, x(t)), \\ x(a) & = & x_0, \end{array} \right. \tag{1}$$

where, for simplicity, we will assume that f is analytic on its domain of definition, and that $x(t)$ is defined for $t \in [a, b]$. We are interested in approximating the function $x(t)$ on $[a, b]$. The idea of the Taylor method is very simple: given the initial condition $x(t_0) = x_0$ ($t_0 = a$), the value $x(t_0 + h)$ is approximated from the Taylor series of the solution $x(t)$ at $t = t_0$,

$$\begin{array}{rcl} x_0 & = & x(t = 0), \\ x_{m+1} & = & x_m + x'(t_m)h + \dfrac{x''(t_m)}{2!}h^2 + \cdots + \dfrac{x^{(p)}(t_m)}{p!}h^p, \quad m = 0, \ldots, M-1, \end{array} \tag{2}$$

where $t_m = a + mh$ and $h = (b - a)/M$. For a practical implementation one needs an effective method to compute the values of the derivatives $x^{(j)}(t_m)$. A first procedure to obtain them is to differentiate the first equation in (1) w.r.t. t, at the point $t = t_m$. Hence,

$$x'(t_m) = f(t_m, x(t_m)), \quad x''(t_m) = f_t(t_m, x(t_m)) + f_x(t_m, x(t_m))x'(t_m),$$

and so on. Therefore, the first step to apply this method is, for a given f, to compute these derivatives up to a suitable order. Then, for each step of the integration (see (2)), we have to evaluate these expressions to obtain the coefficients of the power series of $x(t)$ at $t = t_m$. Usually, these expressions will be very cumbersome, so it will take a significant amount of time to evaluate them numerically. This, jointly with the initial effort to compute the derivatives of f, is the main drawback of this approach for the Taylor method.

This difficulty can be overcome by means of the so-called *automatic differentiation*. This is a procedure that allows for a fast evaluation of the derivatives of a given function, up to arbitrarily high orders. As far as we know, these ideas were first used in Celestial Mechanics problems ([Ste56], [Ste57]). An inconvenience of this method is that f has to belong to a special class. Fortunately, this class is large enough to contain the functions that appear in many applications. We also note that the algorithm that computes these derivatives by automatic differentiation has to be coded separately for different systems. This coding can be either done by a human (see, for instance, [Bro71] for an example with the N-body problem) or by another program.

In a previous work ([JZ05]) we presented a software that, given a function f (belonging to a suitable class), generates code to compute the jet of derivatives of the solution of $x' = f(t,x)$ at a given point (t_m, x_m). The order of the jet is given at run time. The software also generates code to compute, adaptively, an order and step size and to add the resulting Taylor series to predict a new point on the solution. Therefore, the output is a complete time-stepper with automatic order and step size control. The generated code is ANSI C, but we also provide a Fortran 77 wrapper for the main call to the time-stepper. Besides, it is possible to use different floating point arithmetic, like GMP or MPFR. This software is freely downloadable from `http:/www.maia.ub.es/~angel/taylor/` or from `ftp://ftp.ma.utexas.edu/pub/mzou`.

One of the possible improvements for the package is the automatic computation of variational equations of arbitrary order. As before, a first option is to use an algorithm for the symbolic differentiation of the initial ODEs. A second option is to use the ideas of automatic differentiation to compute the derivatives (up to a prescribed order) of the final point of the trajectory w.r.t. the initial condition. For a concrete application of these ideas, see [MB05]. In particular, to apply these techniques one requires a fast algebraic manipulator for polynomials.

In the presentation I will discuss the implementation (in C/C++) of this algebraic manipulator, and we will show how it is used to compute high order variationals. The examples will come from Celestial Mechanics.

References

[Bro71] R. Broucke. Solution of the N-Body Problem with recurrent power series. *Celestial Mech.*, 4(1):110–115, 1971.

[JZ05] À. Jorba and M. Zou. A software package for the numerical integration of ODEs by means of high-order Taylor methods. *Exp. Math.*, 14(1):99–117, 2005.

[MB05] K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based verified integrators: long-term stabilization by preconditioning. *Int. J. Differ. Equ. Appl.*, 10(4):353–384 (2006), 2005.

[Ste56] J.F. Steffensen. On the restricted problem of three bodies. *Danske Vid. Selsk. Mat.-Fys. Medd.*, 30(18):17, 1956.

[Ste57] J.F. Steffensen. On the problem of three bodies in the plane. *Mat.-Fys. Medd. Danske Vid. Selsk.*, 31(3):18, 1957.