Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 163 (2005) 525–545
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Performance of the Taylor series method for ODEs/DAEs

Roberto Barrio

GME, Departamento de Matemática Aplicada, Edificio de Matemáticas, Universidad de Zaragoza, E-50009 Zaragoza, Spain

Abstract

This paper revisits the use of the Taylor series method for the numerical integration of ODEs and DAEs. The numerical method is implemented using an efficient variable-step variable-order scheme. Several numerical tests comparing with well-established numerical codes are presented.

Keywords: Taylor methods; Automatic differentiation; ODE; DAE

1. Introduction

The Taylor method has a long history (in the works of Newton we already see the recursive computation of the Taylor coefficients of the solutions of differential equations and in his proof of the existence of solutions of ODEs Cauchy studied the convergence of the Taylor series of the solution) and it has been rediscovered several times (e.g. in Celestial Mechanics the Taylor series method is called the recurrent power series method [1–5]). In the field of numerical analysis Corliss and coworkers have studied with great detail the case of ODEs [6–9]. Besides, the use of Taylor methods for differential-algebraic equations (DAEs) has been considered the last few years [10,11], although the first attempts appeared in [6].

E-mail addresses: rbarrio@unizar.es, rbarrio@posta.unizar.es (R. Barrio).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2004.02.015

Recently, in several dynamical systems studies, like the determination of periodic orbits [12–14], the numerical integration of the differential systems is done by using the Taylor method. In this kind of problems the Taylor series method has proven its applicability.

Another important application of the Taylor method is that it can be executed using interval arithmetic and thus allows us to obtain validated numerical methods for differential equations. This subject is currently an area of great interest (see [15] and references herein for ODEs and [16] for DAEs). For instance, one important application is in computer assisted proofs in dynamical systems, like the computer assisted proof of the existence of the Lorenz attractor [17], the computer assisted proof of existence of periodic orbits, and so on.

The present paper is organized as follows: Section 2 reviews the Taylor method for ODEs and introduces several variable-stepsize (VS) and variable-order (VO) schemes. Section 3 presents several numerical test comparing with a well-established Runge–Kutta code. Section 4 reviews the Taylor method for DAEs. Section 5 gives numerical comparisons for several DAE systems.

2. Taylor methods for ODEs

Let us consider the initial value problem:

$$\frac{\mathrm{d}\boldsymbol{y}(t)}{\mathrm{d}t} = \boldsymbol{f}(t,\boldsymbol{y}(t)), \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \qquad \boldsymbol{y} \in \mathbb{R}^s, \;\; t \in \mathbb{R}. \tag{1}$$

Now, the value of the solution at $t_{i+1} = t_i + h_{i+1}$ (that is, $\boldsymbol{y}(t_{i+1})$) is approximated from the nth degree Taylor series of $\boldsymbol{y}(t)$ developed at t_i and evaluated at $t = t_{i+1}$ (the function $\boldsymbol{f}$ has to be a smooth function, in this paper we consider that $\boldsymbol{f}$ is analytic).

$$\begin{aligned}
&\boldsymbol{y}(t_0) \stackrel{\text{def}}{=} \boldsymbol{y}_0,\\
&\boldsymbol{y}(t_{i+1}) \simeq \boldsymbol{y}(t_i) + \frac{\mathrm{d}\boldsymbol{y}(t_i)}{\mathrm{d}t} h_{i+1} + \frac{1}{2!}\frac{\mathrm{d}^2\boldsymbol{y}(t_i)}{\mathrm{d}t^2} h_{i+1}^2 + \cdots + \frac{1}{n!}\frac{\mathrm{d}^n\boldsymbol{y}(t_i)}{\mathrm{d}t^n} h_{i+1}^n,\\
&\qquad \simeq \boldsymbol{y}_i + \boldsymbol{f}(t_i,\boldsymbol{y}_i) h_{i+1} + \frac{1}{2!}\frac{\mathrm{d}\boldsymbol{f}(t_i,\boldsymbol{y}_i)}{\mathrm{d}t} h_{i+1}^2\\
&\qquad\quad + \cdots + \frac{1}{n!}\frac{\mathrm{d}^{n-1}\boldsymbol{f}(t_i,\boldsymbol{y}_i)}{\mathrm{d}t^{n-1}} h_{i+1}^n \stackrel{\text{def}}{=} \boldsymbol{y}_{i+1}.
\end{aligned} \tag{2}$$

From the formulation of the Taylor series methods (Eq. (2)), the problem is reduced to the determination of the Taylor coefficients $\{\mathrm{d}^j\boldsymbol{y}(t_i)/\mathrm{d}t^j\}$. Obviously one procedure comes from the differentiation of the second member of the differential equation $\boldsymbol{f}(t,\boldsymbol{y}(t))$. This approach is useful only in theoretical studies because when n grows the complexity of the computation of the

derivatives is too cumbersome. Another approach is the use of automatic differentiation (AD) techniques [18,19] as in [6,7,20,21]. On the literature different pre-processing codes have been designed to get the recurrence formulas of the Taylor coefficients via AD [6,7,20]. In this paper we have used a preprocessor called MARSTIN [21,22] which is a MATHEMATICA package that permits to obtain the Taylor series solution of any ODE problem. The output is the FORTRAN77 program that computes the recurrence formulas to integrate the differential system by recurrent power series, that is, the coefficients of the Taylor method. Another important point is that although the ODE system (1) is a first ODE system the Taylor series method may manage directly high order differential equations just by taking into account that the Taylor coefficients of the solution and its derivatives $\boldsymbol{y}, \boldsymbol{y}', \ldots$ are evidently related

$$\frac{\mathrm{d}^n \boldsymbol{y}^{(d)}}{\mathrm{d}t^n} = (n+1)_d \frac{\mathrm{d}^{n+d}\boldsymbol{y}}{\mathrm{d}t^{n+d}}, \tag{3}$$

being $(a)_d = a(a+1)\cdots(a+d-1)$ the Pochhammer symbol.

In the practical determination of the Taylor coefficients of a function we use the classical rules for automatic differentiation of the elementary functions ($\pm$, $\times$, /, ln, sin,...) developed by Moore [23]. Note that automatic differentiation gives a recursive procedure to obtain the numerical value of the reiterated derivatives of the elementary functions at a given point [18,19]. For completion we present a list of the recurrences for several elementary functions.

Proposition 1. *If $f, g, h : t \in \mathbb{R} \mapsto \mathbb{R}$ are functions of class $\mathscr{C}^n$ and denoting by $f^{[j]}(t)$ the jth Taylor coefficient of $f(t)$ at t, that is $f^{[j]}(t) = \frac{1}{j!} f^{(j)}(t)$, we have*

- *if $h(t) = f(t) \pm g(t)$ then $h^{[n]}(t) = f^{[n]}(t) \pm g^{[n]}(t)$;*
- *if $h(t) = f(t) \cdot g(t)$ then $h^{[n]}(t) = \sum_{i=0}^{n} f^{[n-i]}(t)\ g^{[i]}(t)$;*
- *if $h(t) = f(t)/g(t)$ then if $g(t) \neq 0$*

$$h^{[n]}(t) = \frac{1}{g^{[0]}(t)} \left\{ f^{[n]}(t) - \sum_{i=1}^{n} h^{[n-i]}(t) g^{[i]}(t) \right\};$$

- *if $h(t) = f(t)^{\alpha}$ then $h^{[0]}(t) = (f^{[0]}(t))^{\alpha}$ and for $n > 0$*

$$h^{[n]}(t) = \frac{1}{n f^{[0]}(t)} \sum_{i=0}^{n-1} (n\alpha - i(\alpha+1)) f^{[n-i]}(t)\ h^{[i]}(t);$$

- *if $h(t) = \exp(f(t))$ then $h^{[0]}(t) = \exp\left(f^{[0]}(t)\right)$ and for $n > 0$*

$$h^{[n]}(t) = \frac{1}{n} \sum_{i=0}^{n-1} (n-i) f^{[n-i]}(t)\ h^{[i]}(t);$$

- *if* $h(t) = \ln(f(t))$ *then* $h^{[0]}(t) = \ln\left(f^{[0]}(t)\right)$ *and for* $n > 0$

$$h^{[n]}(t) = \frac{1}{f^{[0]}(t)}\left\{f^{[n]}(t) - \frac{1}{n}\sum_{i=1}^{n-1}(n-i)h^{[n-i]}(t)f^{[i]}(t)\right\};$$

- *if* $g(t) = \cos(f(t))$ *and* $h(t) = \sin(f(t))$ *then*

$$g^{[0]}(t) = \cos\left(f^{[0]}(t)\right), \quad g^{[n]}(t) = -\frac{1}{n}\sum_{i=1}^{n} ih^{[n-i]}(t)f^{[i]}(t);$$

$$h^{[0]}(t) = \sin\left(f^{[0]}(t)\right), \quad h^{[n]}(t) = \frac{1}{n}\sum_{i=1}^{n} ig^{[n-i]}(t)f^{[i]}(t);$$

- *if* $h(t) = \arctan(f(t))$ *then* $h^{[0]}(t) = \arctan(f^{[0]}(t))$ *and for* $n > 0$

$$\begin{cases} f_2^{[n]}(t) = \sum_{i=0}^{n} f^{[n-i]}(t)f^{[i]}(t), \\ h^{[n]}(t) = \frac{1}{1+(f^{[0]}(t))^2}\left(f^{[n]}(t) - \frac{1}{n}\sum_{i=1}^{n-1} ih^{[i]}(t)f_2^{[n-i]}(t)\right). \end{cases}$$

The above formulas may be easily increased with the recurrences of other elementary functions like tan, tanh, cosh, coth, sinh, arccos, arcsin, and so on.

Once we have the formulae to obtain the Taylor coefficients we are interested on the computational cost of such a formulae. In the case of the Taylor coefficients the complexity is well known [20,23]:

Proposition 2. *If the evaluation of* $f(t, \boldsymbol{y}(t))$ *involves* k *elementary functions* $(\times, /, \ln, \exp, \sin, \cos, \ldots)$ *then the computational complexity of the evaluation of* $f^{[0]}, f^{[1]}, \ldots, f^{[n-1]}$ *is* $kn^2 + \mathcal{O}(n)$.

Note that if the function $f(t, \boldsymbol{y}(t))$ is a linear function (therefore it involves only additions/substractions and multiplication by scalars) then the computational complexity is evidently $\mathcal{O}(n)$.

The Taylor method presents several peculiarities. One of them is that it gives directly a dense output in the form of a power series and therefore we can evaluate the solution at any time just by using the Horner algorithm. Besides, it can be formulated using interval arithmetic [15,23] giving guaranteed integration methods, field that has a growing number of applications. Also, as Taylor methods of degree n are also of order n, the use of Taylor methods of high degree gives us numerical methods of high order. Therefore, they are very useful for high-precision solution of ODEs [12,22].

Now we are going to study briefly the behaviour of the Taylor methods for stiff differential equations. It is known that in these situations it is necessary to use A-stable methods. It is clear that Taylor methods are not A-stable (they are

only in the infinite case, taking the infinite series development). In fact, it is easy to obtain that the stability function for the Taylor method of degree n is

$$R_n(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^n}{n!} \tag{4}$$

that is, the same as the explicit Runge–Kutta methods with the same order as the number of internal stages (note that no explicit Runge–Kutta method exists of order n with n stages for $n \geqslant 5$ [24]).

As Taylor methods are not A-stable we are interested in studying the size of the stability domain. In Fig. 1 we represent the stability domain for several Taylor methods. From the figures we observe as when the order grows the stability domain tends to a semicircle centered at the origin. In order to verify numerically this observation we have computed the interval of absolute stability for Taylor methods up to order 60 and we have calculated [22] a linear approximation of it in the mean squares meaning, obtaining that the radius of the semicircle is approximately $r(n) \simeq 1.3614 + 0.3725n$. In the figure on the right we have plotted the stability domain and a semicircle of radius $r(n)$ showing the good approximation of the domain by means of the semicircle. From the analysis, we can establish that high order Taylor methods can be used with moderated stiff equations because the size of the stability domain grows linearly with n in its radius. So, for large n the stability domain will be large enough. Obviously, it cannot be used for highly stiff systems where it is necessary to use implicit A-stable methods. Note that implicit versions of Taylor methods that are A-stable have been recently developed in [8,9,25].

In the practical implementation of a numerical method for the solution of ODEs the use of variable stepsizes is a crucial point because it permits

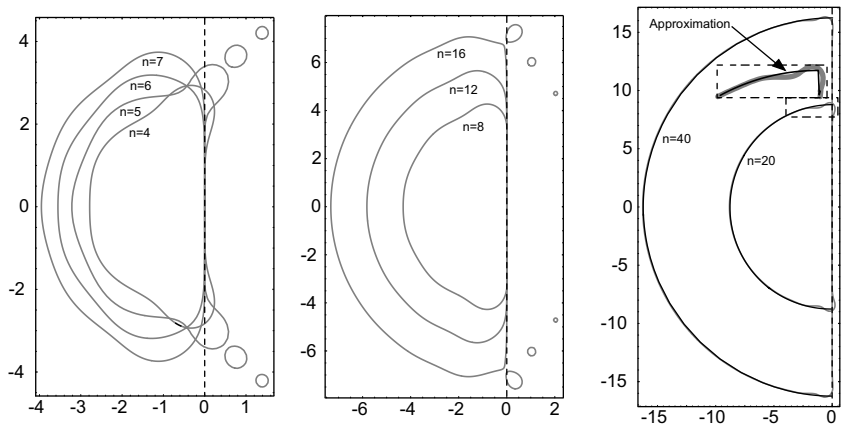

Fig. 1. Evolution of the stability domains of Taylor methods of order 4, 5, 6, 7, 8, 12, 16, 20 and 40.

to automatize the control of the error. Several formulations of VS Taylor methods can be found in [7,20–22] and references herein.

Two of the simplest criteria that we may use to obtain a maximum stepsize within the desired tolerance level `tol` use just one Taylor coefficient. The first one is just to consider

$$h = \left(\frac{\texttt{tol}}{\|\boldsymbol{y}^{[n]}(t_i)\|_\infty} \right)^{1/n}, \tag{5}$$

where $\boldsymbol{y}^{[n]}(t_i) = 1/n!\mathrm{d}^n\boldsymbol{y}(t_i)/\mathrm{d}t^n$ is the last nonzero term of our series approximation. The other stepsize criteria is based on the root criterion of convergence (see [21,22]):

$$h = \texttt{tol}^{1/(n+1)} \|\boldsymbol{y}^{[n]}(t_i)\|_\infty^{-1/n}. \tag{6}$$

Note that this estimator is similar but not the same as just considering the last Taylor coefficient because in (6) we assume the root criterion for the series and we sum the rest by considering $(\|\boldsymbol{y}^{[n]}(t_i)\|_\infty h^n)^{1/n} \leqslant k < 1$ so the rest is lower than $k^{n+1} + k^{n+2} + k^{n+3} + \cdots = k^{n+1}/(1-k)$ and the estimator is obtained by imposing $k^{n+1}/(1-k) = \texttt{tol}$ (for more details see [21,22]).

In this paper, we adopt a different approach in order to make more robust the stepsize estimator. Taking the Lagrange remainder of the Taylor series of degree $n-2$ (although we will consider the Taylor series up to degree n for approximating the solution), we have

$$\begin{aligned} R_{t_i,n-2}(t_{i+1}) = R_{t_i,n-2}(t_i + h) &= \frac{\boldsymbol{y}^{(n-1)}(t_i + \xi h)}{(n-1)!} h^{n-1} = \boldsymbol{y}^{[n-1]}(t_i + \xi h) h^{n-1} \\ &= \frac{h^{n-1}}{(n-1)!} \left(\boldsymbol{y}^{(n-1)}(t_i) + \boldsymbol{y}^{(n)}(t_i)(\xi h) + \frac{\boldsymbol{y}^{(n+1)}(t_i)}{2!} (\xi h)^2 + \cdots \right) \\ &= h^{n-1} \Bigg(\boldsymbol{y}^{[n-1]}(t_i) + n\boldsymbol{y}^{[n]}(t_i)(\xi h) \\ &\qquad + \cdots + \frac{(n)_k}{k!} \boldsymbol{y}^{[n+k-1]}(t_i)(\xi h)^k + \cdots \Bigg) \end{aligned}$$

with $\xi \in (0,1)$. Now, taking the two first terms on the series development of the remainder (note that ξh is small when the error tolerance is small and Taylor methods normally are used for high-precision computations) we have

$$R_{t_i,n-2}(t_{i+1}) \simeq R_2 \equiv h^{n-1} \left(\boldsymbol{y}^{[n-1]}(t_i) + n\boldsymbol{y}^{[n]}(t_i)(\xi h) \right)$$

and taking norms

$$\|R_2\|_\infty \leqslant h^{n-1} \left\{ \|\boldsymbol{y}^{[n-1]}(t_i)\|_\infty + h \cdot n \|\boldsymbol{y}^{[n]}(t_i)\|_\infty \right\}.$$

Therefore, we obtain the stepsize h for a given relative tolerance `TolRel` and absolute tolerance `TolAbs` by imposing

$$h^{n-1} \left\{ \|\boldsymbol{y}^{[n-1]}(t_i)\|_\infty + h \cdot n \|\boldsymbol{y}^{[n]}(t_i)\|_\infty \right\} = \texttt{Tol}$$

with $\mathtt{Tol} = \min\left\{\mathtt{TolRel}\cdot\max\{\|\boldsymbol{y}^{[0]}(t_i)\|_\infty, \|\boldsymbol{y}^{[1]}(t_i)\|_\infty\}, \mathtt{TolAbs}\right\}$. Now, by calling

$$A = \|\boldsymbol{y}^{[n-1]}(t_i)\|_\infty, \quad B = n\|\boldsymbol{y}^{[n]}(t_i)\|_\infty,$$

we have

$$h^{n-1}(A + hB) = \mathtt{Tol}$$

and so, applying the Newton method we have (usually just one or two iterations are enough)

$$h = h_0 - \frac{h_0^{n-1}(A + h_0B) - \mathtt{Tol}}{h_0^n((n-1)A + h_0nB)}.$$

If we take as initial value of the Newton process the classical stepsize control for Taylor methods

$$h_0 = \left(\frac{\mathtt{Tol}}{\|\boldsymbol{y}^{[n-1]}(t_i)\|_\infty}\right)^{1/(n-1)} = \left(\frac{\mathtt{Tol}}{A}\right)^{1/(n-1)},$$

we have that the first iteration will be

$$h = h_0\left(1 - \frac{h_0B}{(n-1)A + h_0nB}\right).$$

Another stepsize criteria may be just to use the information of the last two coefficients instead of only one as in (6) and imposing that both sets of coefficients are lower than the tolerance level, that is

$$h = \mathtt{fac}\cdot\min\left\{\left(\frac{\mathtt{Tol}}{\|\boldsymbol{y}^{[n-1]}(t_i)\|_\infty}\right)^{1/(n-1)}, \left(\frac{\mathtt{Tol}}{\|\boldsymbol{y}^{[n]}(t_i)\|_\infty}\right)^{1/n}\right\}, \tag{7}$$

where fac is a safety factor.

Note that, "a priori", in the Taylor methods there is no rejected step as occurs in any VS formulation for Runge–Kutta or multistep methods because we chose the stepsize once the series are generated in order to obtain a required precision level. But, in order to give more guarantee about the stepsize we may analyze the agreement between the tangent vector to the Taylor polynomial and the vector field at the end of the step (see [13]), that is, given the Taylor approximation of the solution on the interval $[t_i, t_{i+1}] = [t_i, t_i + h_{i+1}]$

$$\boldsymbol{y}(t) \simeq \sum_{k=0}^{n}\boldsymbol{y}^{[k]}(t_i)\cdot(t - t_i)^k, \quad \boldsymbol{y}'(t) \simeq \sum_{k=1}^{n}k\boldsymbol{y}^{[k]}(t_i)\cdot(t - t_i)^{k-1}$$

then evaluating at the end of the interval $\boldsymbol{y}'_{i+1} \equiv \sum_{k=1}^{n} k\boldsymbol{y}^{[k]}(t_i)\cdot(h_{i+1})^{k-1}$ and the criteria for rejecting the stepsize is

$$\text{if} \quad \|\boldsymbol{y}'_{i+1} - \boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})\|_\infty > \texttt{Tol} \quad \text{then} \quad \tilde{h}_{i+1} = \texttt{facr}\cdot h_{i+1}, \tag{8}$$

where `facr` is a factor that reduces the stepsize (we have taken $\texttt{facr} = 0.8$). It is important to remark that although the stepsize may be rejected we do not have to recalculate the Taylor coefficients, we only have to consider the new stepsize and enter again in the criteria for rejecting the stepsize. Therefore we cannot say that we reject a complete step, we just reject the estimation of the stepsize, and so the computational cost is not very high (in fact the cost of evaluating $\boldsymbol{y}'_{i+1}$ and $\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})$).

In Fig. 2, we present the precision-computer time diagram for the Lorenz problem (see the next section for a description) using the three different stepsize estimators: using one or two Taylor coefficients and using the Lagrange remainder. Besides, we also analyze the use of the criteria of rejection the stepsize. From the figure we observe as the worst performance is obtained for the stepsize estimator that uses only one coefficient, being much less accurate if

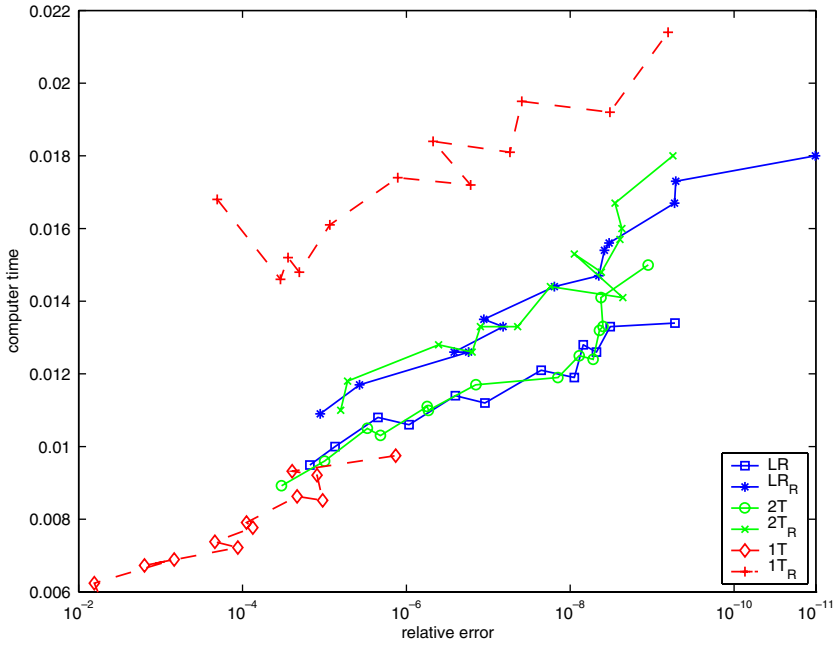

Fig. 2. Precision-computer time diagram for the Taylor methods for the Lorenz problem using the Lagrange remainder (LR) stepsize estimator, the one-term (1T) estimator (6) and the two-terms (2T) estimator (7), used with (subscript R) and without stepsize rejection (8).

we do not enter in the rejection criteria. On the contrary, when we use two coefficients or the Lagrange remainder the behaviour is similar in both cases and the rejection criteria do not improve the precision making the method slower. Therefore, the two-term estimators may be used without it in most of the problems (for solutions with series with a single real or a conjugate pair of primary singularities [7]). Note that in [13] the authors use a one-term stepsize estimator.

In order to obtain a more adaptive method we use a VO formulation of the Taylor method developed in [22]. To reach this goal it is necessary to know "a priori" an estimation of both, the computational time and the stepsize for a fixed-error tolerance `Tol`, for the different orders. On our own, we fix the order increment to p, that is, our possibilities are $n_i - p$, n_i or $n_i + p$, being n_i the order of the last step in the numerical integration. It is known that the computational complexity of each step of the Taylor method is $\mathcal{O}(n^2)$ for the nonlinear case, see [18,20]. Another information that we need to change the order is an estimation of the stepsize depending on the order n_i on the ith step. Obviously, we have to use the information of the previous step. Therefore, using the stepsize criteria (6) for simplicity, it is easy to obtain an estimation of the stepsize in the case of reducing the order to $n_i - p$:

$$h_{\text{est}}^{-} = \texttt{Tol}^{1/(n_i-p+1)} \cdot \|\boldsymbol{y}^{[n_i-p]}(t_i)\|_{\infty}^{-1/(n_i-p)}. \tag{9}$$

Much more complicated is the case of increasing the order to $n_i + p$ because now we do not know the coefficient $\boldsymbol{y}^{[n_i+p]}(t_i)$. Therefore, we have to estimate it. Assuming the root criterion on the Taylor series we first compute an approximation of the radius of convergence ρ. Using the information of the last coefficients of the previous step, we consider, in order to avoid problems with odd or even functions,

$$\rho \approx \rho_{\text{est}} = \min\left\{ \left\| \frac{\boldsymbol{y}^{[n_i-1]}(t_i)}{\boldsymbol{y}^{[n_i]}(t_i)} \right\|_{\infty}, \left\| \frac{\boldsymbol{y}^{[n_i-2]}(t_i)}{\boldsymbol{y}^{[n_i]}(t_i)} \right\|_{\infty}^{1/2}, \left\| \frac{\boldsymbol{y}^{[n_i-3]}(t_i)}{\boldsymbol{y}^{[n_i-1]}(t_i)} \right\|_{\infty}^{1/2} \right\},$$

where we denote $\|\boldsymbol{a}/\boldsymbol{b}\|_{\infty} := \max\{|a_1/b_1|, \ldots, |a_s/b_s|\}$ and so an estimation of the stepsize for a Taylor method of order $n_i + p$ will be

$$h_{\text{est}}^{+} = \texttt{Tol}^{1/(n_i+p+1)} \cdot \left(\frac{\|\boldsymbol{y}^{[n_i]}(t_i)\|_{\infty}}{\rho_{\text{est}}^{p}} \right)^{-1/(n_i+p)}. \tag{10}$$

Now we can analyze the advantage of changing the order in the integration process. We have considered that if in the previous step the order has increased or the step has decreased, then the order tends to increase, on the opposite case, we suppose that the order tends to decrease. So, we only have to compare two cases on each integration step: order n_i and $n_i - p$ or n_i and $n_i + p$. As the computational time is $\mathcal{O}(n_i^2)$ the criteria for changing the

order is (`facl` and `fac2` are factors to control the reduction or increment of the order)

if $n_i > n_{i-1}$ or $h_i < h_{i-1}$ then
  if $\left(\frac{n_i+p+1}{n_i+1}\right)^2 < \texttt{facl}\,\frac{h^+_{\text{est}}}{h_i}$ then $n_{i+1} = n_i + p$ else $n_{i+1} = n_i$
else
  if $\left(\frac{n_i-p+1}{n_i+1}\right)^2 < \texttt{fac2}\,\frac{h^-_{\text{est}}}{h_i}$ then $n_{i+1} = n_i - p$ else $n_{i+1} = n_i$
end if

As initial order n_0 we use the criteria given in [20], that is, $n_0 = -\frac{1}{2}\ln \texttt{Tol}$.

In Fig. 3, we show the precision-computer time diagrams for the Lorenz problem for the Taylor methods using variable-order or fixed-order strategies. The numerical tests have been done using quadruple precision (therefore, as the problem is chaotic the figures are different as the ones using double precision of the next section) in order to analyze the behaviour for calculations with very high precision. From the figure we observe as the VO scheme gives always a good performance. A fixed-order strategy may be useful only once we know in advance a good order for a particular problem and precision but the variable-

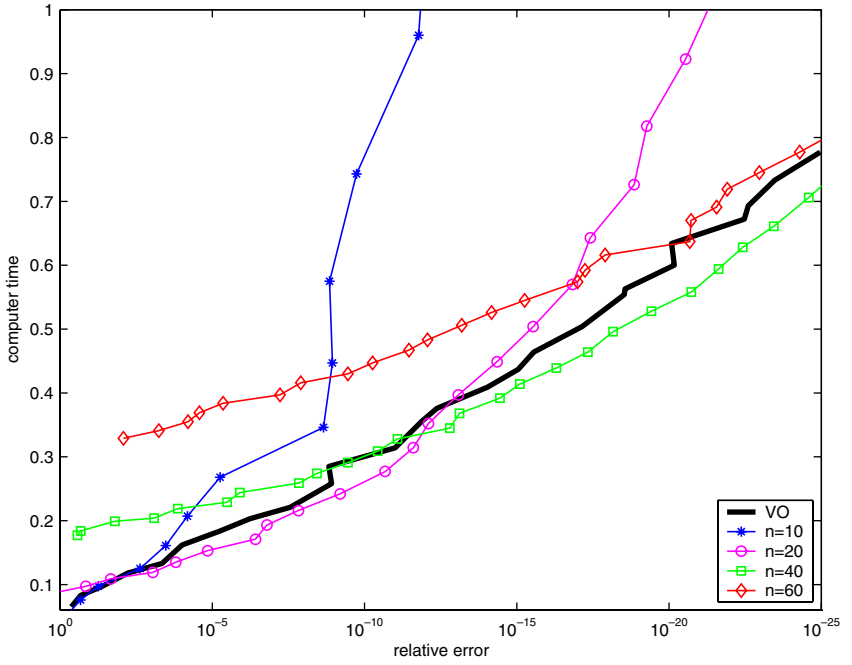

Fig. 3. Precision-computer time diagrams for the Taylor methods for the Lorenz problem using variable-order (VO) or fixed-order strategies.

Table 1
Number of steps using fixed- and variable-order Taylor methods for different error tolerances

	$\mathtt{Tol} = 10^{-5}$	10^{-10}	10^{-15}	10^{-20}	10^{-25}
$n = 10$	314	1111	3985	14,313	51,433
$n = 20$	142	257	467	853	1562
$n = 40$	098	130	175	234	314
$n = 60$	092	112	136	166	202
VO	279	429	486	604	616
Average order	10	14	19	22	27

In the variable-order (VO) case the average order is written on the bottom line.

order is useful for all cases. In Table 1, we present the number of steps in the integration of the Lorenz problem for different tolerance levels and the average order using the VO scheme.

Therefore the complete scheme of the Taylor series method for an ODE system will be

Taylor method for ODEs

1. Use of a pre-processor for the generation of the decomposition in elementary functions of the second member of the differential system.
2. On each step i,
   2.1. select the degree n_i using a VO scheme;
   2.2. `for` $k = 0$ `to` n_i, compute using the AD rules each Taylor coefficient;
   2.3. select the stepsize h_{i+1} using a VS criteria;
   2.4. use the rejection criteria to accept or not the stepsize:
   `while` (stepsize rejected) `do` $h_{i+1} = \mathtt{facr} \cdot h_{i+1}$.

3. ODEs: numerical tests

We have performed several numerical tests on a Windows PC-866MHz using `FORTRAN77` as programming language. We compare with the well-established code `dop853` developed by Hairer et al. [26] (it is based on an explicit Runge–Kutta of order 8(5,3) given by Dormand and Prince with stepsize control and dense output). The Taylor method has been implemented with the variable-stepsize and variable-order (VSVO) scheme using the two-term estimator (7) without the rejection criteria (8).

As model problems we have taken three classical ODE problems:

- *A Galactic dynamics model* [26,27]. This problem is a Hamiltonian problem with coordinates q_1, q_2, q_3 and moments p_1, p_2, p_3. The Hamiltonian function and parameter values (the initial conditions have been fixed to obtain $\mathscr{H} = 2$) are

$$\mathscr{H} = \frac{1}{2}(p_1^2 + p_2^2 + p_3^2) + \Omega(p_1 q_2 - p_2 q_1) + A \ln\left(C + \frac{q_1^2}{a^2} + \frac{q_2^2}{b^2} + \frac{q_3^2}{c^2}\right),$$

$$\begin{cases} a = 1.25, \quad b = 1, \quad c = 0.75 \quad A = 1, \quad C = 1, \quad \Omega = 0.25, \\ q_1(0) = 2.5, \quad q_2(0) = q_3(0) = 0, \\ p_1(0) = 0, \quad p_2(0) = \frac{1}{40}\left(25 + \sqrt{6961 - 3200 \ln 5}\right), \quad p_3(0) = 0.2 \end{cases}$$

and the integration time is [0, 1000].

- *The Lorenz model* [28] *with Saltzman's values*. This problem is a classical example of chaotic behaviour:

$$x' = -\sigma(x - y), \quad y' = -xz + rx - y, \quad z' = xy - bz.$$

$$\begin{cases} x(0) = -8, \quad y(0) = 8, \quad z(0) = r - 1, \\ b = 8/3, \qquad \sigma = 10, \qquad r = 28 \end{cases}$$

and the integration time is [0, 16].

- *The Kepler problem*, that describes the planar two body motion with eccentricity e:

$$x'' = -x/(x^2 + y^2)^{3/2}, \quad y'' = -y/(x^2 + y^2)^{3/2}$$

and initial conditions $x(0) = 1 - e$, $y(0) = 0, x'(0) = 0$ and $y'(0) = \sqrt{(1+e)/(1-e)}$. The integration time is $[0, 200 \cdot 2\pi]$ (that is, 200 periods). We have considered two values of the eccentricity, $e = 0.7$ and 0.99. Note that this problem is usually employed as test for VS strategies for high values of the eccentricity (near 1).

In Fig. 4, we present the precision-computer time diagrams using the `dop853` code and the Taylor method for the different problems. In all the numerical tests we observe as for high precision the Taylor method presents with difference the best behaviour. It is interesting to remark the very good performance for the Kepler problem with eccentricity 0.99, giving us an indicator of the correct behaviour of the stepsize estimator. Besides, we can observe the different evolution of both methods, the computer time increment for the Taylor methods is much slower (linear in the logarithmic scale of the figure) than for the RK method. This is due to the VO implementation of the Taylor methods. Note that although the performance of Taylor methods for these problems is quite encouraging this method may be used mainly for low dimensional problems (like many interesting problems in dynamical systems) due to memory requirements.

In Table 2, we show the scheme for the determination of the Taylor coefficients for two of the above problems: the Galactic and the Kepler problems. Any operation inside a box stands for an operation among Taylor series and $s^{[m]}$ stands for the mth Taylor coefficient of the Taylor series s (the mth coefficient of an operation among Taylor series is calculated by means of the AD

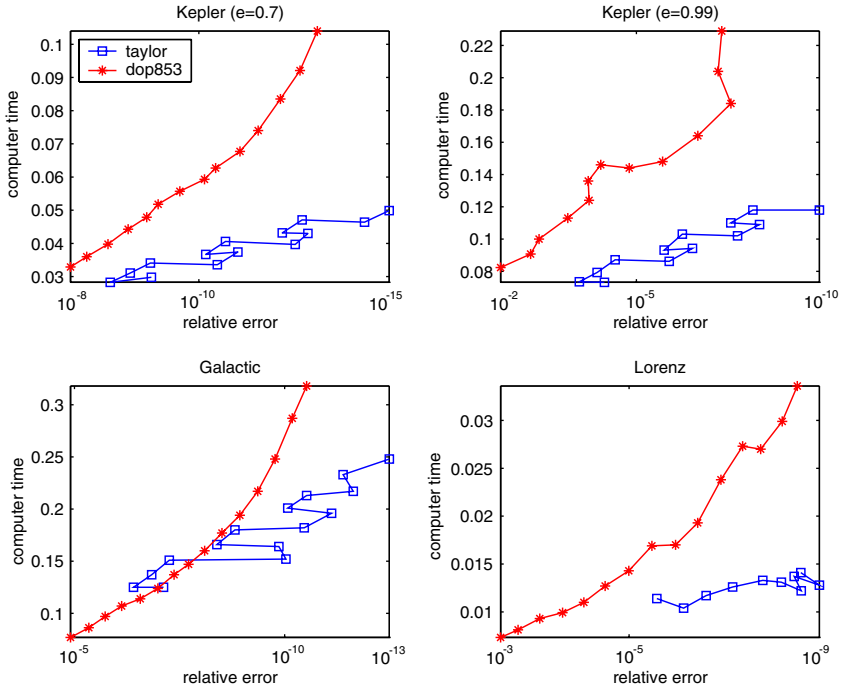

Fig. 4. Precision-computer time diagrams for the `dop853` and the Taylor methods for the Kepler (for two values of the eccentricity), Galactic and Lorenz problems.

Table 2
Computation of the Taylor coefficients for the Galactic and the Kepler problems (point 2.2) of the scheme of the Taylor method for ODEs)

Galactic problem	Kepler problem
for $m = 0$ to $n-1$ do	for $m = 0$ to $n-2$ do
$s_1^{[m]} = a^{-2}\boxed{q_1 \times q_1}^{[m]} + b^{-2}\boxed{q_2 \times q_2}^{[m]} + c^{-2}\boxed{q_3 \times q_3}^{[m]}$	$s_1^{[m]} = \boxed{x \times x}^{[m]} + \boxed{y \times y}^{[m]}$
if $(m = 0)$ then $s_1^{[0]} = s_1^{[0]} + C$	$s_2^{[m]} = \boxed{(s_1)^{-3/2}}^{[m]}$
$q_1^{[1+m]} = (p_1^{[m]} + \Omega q_2^{[m]})/(1+m)$	$c = (1+m)(2+m)$
$q_2^{[1+m]} = (p_2^{[m]} - \Omega q_1^{[m]})/(1+m)$	$x^{[2+m]} = -\boxed{x \times s_2}^{[m]}/c$
$q_3^{[1+m]} = p_3^{[m]}/(1+m)$	$y^{[2+m]} = -\boxed{y \times s_2}^{[m]}/c$
$p_1^{[1+m]} = (\Omega p_2^{[m]} - 2Aa^{-2}\boxed{q_1/s_1}^{[m]})/(1+m)$	end
$p_2^{[1+m]} = -(\Omega p_1^{[m]} + 2Ab^{-2}\boxed{q_2/s_1}^{[m]})/(1+m)$	
$p_3^{[1+m]} = -2Ac^{-2}\boxed{q_3/s_1}^{[m]}/(1+m)$	
end	

rules of Proposition 1). Note that for the Kepler problem we have used the second order formulation of the problem directly, without passing to a first order ODE system.

4. Taylor methods for DAEs

In the case of smooth DAEs the Taylor method was first applied by Chang and Corliss [6,7] and a systematic study was done by Pryce [10,11]. Let be a DAE system

$$\boldsymbol{F}\boldsymbol{y} := \boldsymbol{f}\left(t, \boldsymbol{y}, \boldsymbol{y}', \dots\right) = 0, \tag{11}$$

$$\boldsymbol{\Phi}\boldsymbol{y}_0 := \phi\left(t_0, \boldsymbol{y}_0, \boldsymbol{y}_0', \dots\right) = 0 \tag{12}$$

with $\boldsymbol{y}, \boldsymbol{y}', \dots \in \mathbb{R}^s$. The consistent initial conditions are given by (12). The Taylor series method for DAEs involves a pre-analysis stage where one solves an assignment problem (the Σ-*method* [11]).

The solvability of a DAE system by means of the Taylor series method is given by the following theorem [10]:

Theorem 3 (Pryce [10]). *Let the DAE* $\boldsymbol{F}\boldsymbol{y} = 0$ *be analytic in a neighbourhood of a consistent point. Then the Taylor series method applied at this point succeeds if the* $s \times s$ *matrix* J *is nonsingular at this point, where*

$$J_{ij} = \begin{cases} \frac{\partial f_i}{\partial y_j^{(d_j - c_i)}} & \text{if this derivative is present,} \\ 0, & \text{otherwise.} \end{cases}$$

The indices d_j and c_i give, respectively, the appropriate derivatives of the system variables y_j and the number of differentiations of the ith equation needed to convert to an explicit ODE system, see [10,11].

For a practical implementation of Taylor methods for DAEs first we enter in the pre-analysis stage [10,11] that gives us the new equations $\widehat{\boldsymbol{F}}\boldsymbol{y} := \hat{\boldsymbol{f}}(t, \boldsymbol{y}, \boldsymbol{y}', \dots, \boldsymbol{y}^{(\mathbf{d})}) = 0$, where we denote $\boldsymbol{y}^{(\mathbf{d})} = (y_1^{(d_1)}, \dots, y_s^{(d_s)})$. Then, we use AD in the generation of the Taylor coefficients of the function $\hat{\boldsymbol{f}}$, in a similar way as in the case of ODEs, and afterwards we solve for each Taylor coefficient a linear system with the same coefficient matrix J (defined in Theorem 3)

$$-J \cdot \left(\boldsymbol{y}^{(\mathbf{d})}\right)_k^{\mathrm{T}} = \left(\widehat{\boldsymbol{F}}\boldsymbol{y}\right)_{k0} := \left(\hat{\boldsymbol{f}}\left(t, \boldsymbol{y}, \boldsymbol{y}', \dots, \boldsymbol{y}^{(\mathbf{d})}\right)\right)_{k0},$$
$$\text{with} \quad \left(\boldsymbol{y}^{(\mathbf{d})}\right)_k = \left(\left(y_1^{(d_1)}\right)_k, \dots, \left(y_s^{(d_s)}\right)_k\right),$$

where $(g)_k$ stands for the kth Taylor coefficient of g and $\left(\widehat{\boldsymbol{f}}(t,\boldsymbol{y},\boldsymbol{y}',\ldots,\boldsymbol{y}^{(\mathbf{d})})\right)_{k0}$ is the kth Taylor coefficient of $\widehat{\boldsymbol{f}}$ considering $\left(\boldsymbol{y}^{(\mathbf{d})}\right)_k = 0$. Note that we may use the relation among the Taylor coefficients of a function and its derivatives (3) to obtain the series of the solution $\boldsymbol{y}$. We remark that for each Taylor coefficient we solve the above linear system but the matrix J is the same for any Taylor coefficient on each step, therefore a LU decomposition of the matrix J is recommended. On each step we also have to obtain the first Taylor coefficients in such a way that the solution is consistent with the DAE at the point of the development of the Taylor series (what we may call a pre-projection, instead of the classical projection at the end of each integration step, although strictly speaking it is not a projection at all, it is just the process to obtain the first Taylor coefficients). This involves to solve the nonlinear equation (12) with the constraints of the DAE. Note that apart from the initial conditions, the convergence of, for example, the Newton methods on this problem with the numerical values of the Taylor series at the end of the interval as initial guess is quite fast (in general one iteration is enough) because these values are very good approximations of the solution. For more details about the use of Taylor series in the numerical solution of DAEs, see [10,11].

In the case of DAEs we may also use a rejection criteria similar to that for ODEs

$$\text{if} \quad \|\widehat{\boldsymbol{F}}\boldsymbol{y}\|_\infty := \|\widehat{\boldsymbol{f}}(t,\boldsymbol{y},\boldsymbol{y}',\ldots,\boldsymbol{y}^{(\mathbf{d})})\|_\infty > \texttt{Tol} \quad \text{then} \quad \tilde{h}_{i+1} = \texttt{facr}\cdot h_{i+1}. \tag{13}$$

In order to clarify the Taylor method for DAEs, we develop in detail a very simple example (that has been also studied [10]).

Example. Let be the simple pendulum problem of length L in Cartesian coordinates. This is an index 3 problem (see [29] for details):

$$\begin{cases} E_1 \equiv x'' + x\lambda = 0, \\ E_2 \equiv y'' + y\lambda - g = 0, \\ C \equiv x^2 + y^2 - L^2 = 0. \end{cases}$$

Scheme of the solution (Taylor method for DAEs):

1. Structural analysis: by applying the Σ-method [11] we obtain

	x	y	λ
d_j	2	2	0

	E_1	E_2	C
c_i	0	0	2

   Therefore the equations $\widehat{\boldsymbol{F}}\boldsymbol{y} = 0$ that we use are E_1, E_2 and $\mathrm{d}^2C/\mathrm{d}t^2$ for the obtention of the general Taylor coefficients and C and $\mathrm{d}C/\mathrm{d}t$ for obtaining the first Taylor coefficients.

2. Enlarged system (system + hidden constraints):

$$\begin{cases} E_1 \equiv x'' + x\lambda = 0, \\ E_2 \equiv y'' + y\lambda - g = 0, \\ C \equiv x^2 + y^2 - L^2 = 0, \\ C' \equiv 2xx' + 2yy' = 0, \\ C'' \equiv 2xx'' + 2yy'' + 2(x')^2 + 2(y')^2 = 0. \end{cases}$$

3. Use of a pre-processor for the generation of the decomposition in elementary functions of the DAE system.
4. System Jacobian

$$J = \begin{array}{c} \begin{matrix} x'' & y'' & \lambda \end{matrix} \\ \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 2x & 2y & 0 \end{pmatrix} \end{array} \begin{matrix} E_1 \\ E_2 \\ C'' \end{matrix}.$$

5. $\det(J) = -2(x^2 + y^2) = -2L^2 \neq 0 \rightarrow$ Taylor series method works.
6. On each step i,
   6.1. select the degree n_i using a VO scheme;
   6.2. solve C for x_0, y_0 and C' for x_1, y_1 (pre-projection);
   6.3. `for` $k = 0$ `to` n_i, solve the linear system on each Taylor coefficient

$$\begin{aligned} -J \cdot (x_k'', y_k'', \lambda_k)^{\mathrm{T}} &= -J \cdot ((k+1)(k+2)x_{k+2}, (k+1)(k+2)y_{k+2}, \lambda_k)^{\mathrm{T}} \\ &= \boldsymbol{f}(x, x', x'', y, y', y'', \lambda)_{k0} \end{aligned}$$

   being x_k the kth Taylor coefficient of x and $\boldsymbol{f}_{k0}$ the kth Taylor coefficient of the right hand member of equations E_1, E_2 and C'' (obtained by using the AD rules of Proposition 1) considering $x_k'' = y_k'' = 0$ (so $x_{k+2} = y_{k+2} = 0$).
   6.4. select the stepsize h_{i+1} using a VS criteria;
   6.5. use the rejection criteria to accept or not the stepsize:
   `while` (stepsize rejected) `do` $h_{i+1} =$ `facr` $\cdot\, h_{i+1}$.

In the above example, Steps 1–5 are initial steps and they are done only once at the preparatory analysis of the DAE. Step 6 is the integration step.

5. DAEs: numerical tests

In these numerical tests we compare with the `dop853` code for nonstiff problems and `radau5` and `radau` [24] for the stiff ones. The codes `radau5` and `radau` are based on implicit RK schemes (Radau IIA), the first one of order 5 and the second one of variable orders (orders 5, 9 or 13) both with stepsize control and continuous output.

We have taken three classical DAE problems (two of them slightly stiff) as test problems:

- *Nonlinear simple pendulum*: a DAE system of index 3 (we have taken the values $L = g = 1$). We have considered the time interval [0, 100].

$$\begin{cases} x'' + x\lambda = 0, \\ y'' + y\lambda - g = 0, \\ x^2 + y^2 - L^2 = 0, \end{cases} \begin{vmatrix} x_0 = 1/2, \quad x_0' = 1, \\ y_0 = -\sqrt{1 - x(0)^2}, \quad y_0' = -x_0 x_0'/y_0, \\ \lambda_0 = y_0 + (x_0')^2 + (y_0')^2 = \frac{1}{6}(8 - 3\sqrt{3}). \end{vmatrix}$$

- *Transistor–amplifier problem* [24]: This problem is an example of a stiff DAE system of index 1. The term $U_e(t)$ is the entry voltage, $U_i(t)$ $(i = 1, \ldots, 5)$ the voltages at the nodes $1, \ldots, 5$ and $U_5(t)$ the output voltage. We have considered the time interval [0, 0.5].

$$M\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{U}), \quad \boldsymbol{U}(0) = (0, 3, 3, 6, 0)^{\mathrm{T}}$$

with

$$M = \begin{pmatrix} -C_1 & C_1 & 0 & 0 & 0 \\ C_1 & -C_1 & 0 & 0 & 0 \\ 0 & 0 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & -C_3 & C_3 \\ 0 & 0 & 0 & C_3 & -C_3 \end{pmatrix},$$

$$\boldsymbol{f}(\boldsymbol{U}) = \begin{pmatrix} -\frac{U_e(t)}{R_0} + \frac{U_1}{R_0} \\ -\frac{U_b}{R_2} + U_2\left(\frac{1}{R_1} + \frac{1}{R_2}\right) + (1 - \alpha)g(U_2 - U_3) \\ \frac{U_3}{R_3} - g(U_2 - U_3) \\ -\frac{U_b}{R_2} + \frac{U_4}{R_4} + \alpha g(U_2 - U_3) \\ \frac{U_5}{R_5} \end{pmatrix}$$

$$g(x) = 10^{-6}\left(\exp\left(\tfrac{x}{0.026}\right) - 1\right), \qquad U_e(t) = 0.4\sin(200\pi t),$$
$$R_0 = 1000, \quad R_1 = \cdots = R_5 = 9000, \quad U_b = 6, \quad C_1 = 10^{-6}, \quad C_k = kC_1.$$

- *The Chemical Akzo-Nobel problem* (DAE problem 1 of *Test Set for IVP Solvers* (release 2.2), see for a complete description http://hilbert.dm.uniba.it/~testset/): This IVP is a stiff system of 6 DAEs of index 1. It describes a chemical process in which two species are mixed, while carbon dioxide is continuously added. We have considered the time interval [0, 180].

$$M\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{y}), \quad \boldsymbol{y}(0) = (0.444, 0.00123, 0, 0.0017, 0, K_s \cdot y_1(0) \cdot y_4(0))^{\mathrm{T}},$$

with

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{y}) = \begin{pmatrix} -2r_1 + r_2 - r_3 - r_4 \\ -\frac{1}{2}r_1 - r_4 - \frac{1}{2}r_5 + F_{\text{in}} \\ r_1 - r_2 + r_3 \\ -2_2 + r_3 - 2r_4 \\ r_2 - r_3 + r_5 \\ K_s \cdot y_1 \cdot y_4 - y_6 \end{pmatrix},$$

$$r_1 = k_1 \cdot y_1^4 \cdot y_2^{1/2}, \quad r_2 = k_2 \cdot y_3 \cdot y_4, \quad r_3 = \frac{k_2}{K} \cdot y_1 \cdot y_5,$$

$$r_4 = k_3 \cdot y_1 \cdot y_4^2, \quad r_5 = k_4 \cdot y_6^2 \cdot y_2^{1/2}, \quad F_{\text{in}} = klA \cdot \left(\frac{p(\text{CO}_2)}{H} - y_2 \right).$$

The value of the parameters $k_1, k_2, k_3, k_4, K, klA, p(\text{CO}_2)$ and H are $k_1 = 18.7$, $k_2 = 0.58$, $k_3 = 0.09$, $k_4 = 0.42$, $K_s = 115.83$, $K = 34.4$, $p(\text{CO}_2) = 0.9$, $klA = 3.3$, $H = 737$.

Note that in the DAE problems the initial conditions have to be consistent with the problem.

We observe in Fig. 5 as, obviously, the explicit codes `dop853` and Taylor methods are the fastest methods, being the Taylor method the fastest one. In Figs. 6 and 7 the code `dop853` does not work as the problems are stiff, but the Taylor method of high order works and it is the fastest one. Note that although in these stiff problems the Taylor method may be used this is not true for highly stiff systems where we need implicit methods. In Figs. 5–7, we can appreciate the different slope of the VO methods (Taylor and `radau`) and the fixed-order ones, being clear as for high precision the VO schemes become the more competitive because they are more versatile.

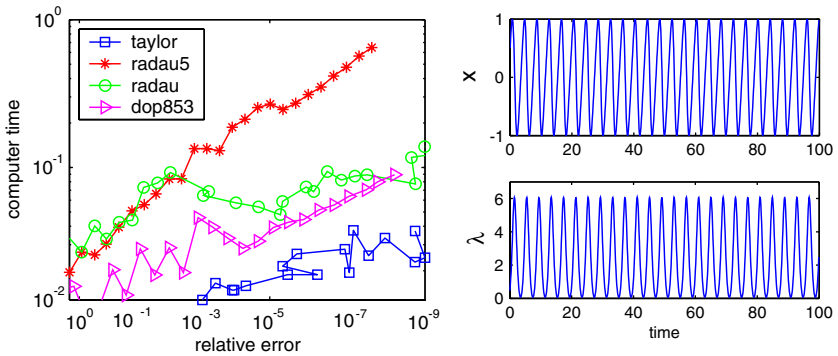

Fig. 5. Left: precision-computer time diagrams for the `dop853`, `radau5` and `radau` codes and the Taylor method for the pendulum problem. Right: evolution of the coordinates x and λ.

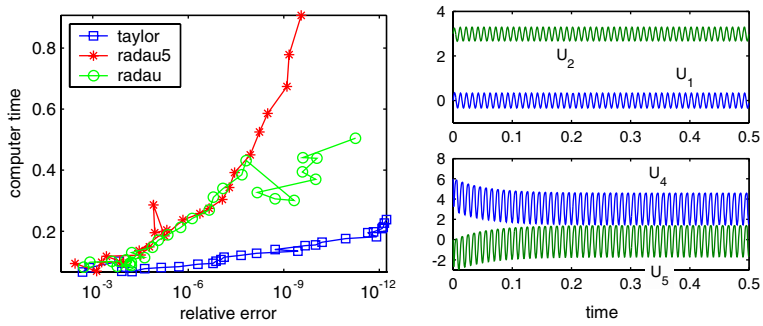

Fig. 6. Left: precision-computer time diagrams for the `radau5` and `radau` codes and the Taylor method for the transistor–amplifier problem. Right: evolution of the voltages U_1, U_2, U_4 and U_5.

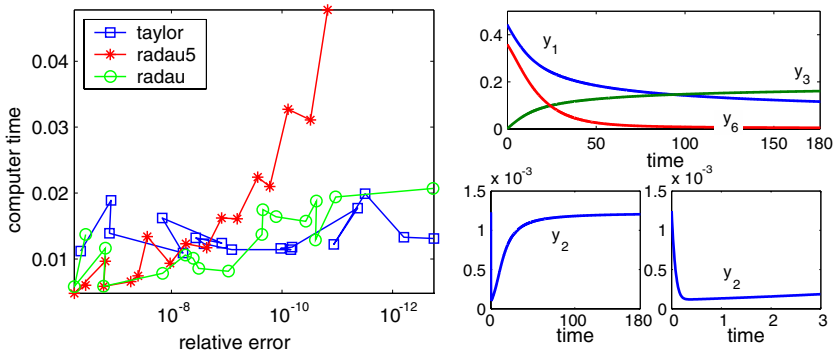

Fig. 7. Left: precision-computer time diagrams for the `radau5` and `radau` codes and the Taylor method for the Chemical Akzo-Nobel problem. Right: evolution of the coordinates y_1, y_2, y_3 and y_6.

6. Conclusions

In the present paper, we have done a comparison of the use of the Taylor series method for the numerical solution of ODE/DAE systems. From the numerical tests we observe as this method may be quite useful for low dimensional problems like in many interesting problems in dynamical systems [12,13], where we are interested also in high precision. For this kind of problems the use of the Taylor series method gives an interesting alternative to other schemes.

Acknowledgements

The author is supported by the Spanish Research Grant DGYCT BFM2003-02137.

References

[1] A. Deprit, J.F. Price, The computation of characteristic exponents in the planar restricted problem of three bodies, Astron. J. 70 (1965) 836–846.
[2] A. Deprit, R.V.M. Zahar, Numerical integration of an orbit and its concomitant variations by recurrent power series, Z. Angew. Math. Phys. 17 (1966) 425–430.
[3] A.E. Roy, P.E. Moran, W. Black, Studies in the application of recurrence relations to special perturbation methods I, Celestial Mech. 6 (1972) 468–482.
[4] J.F. Steffensen, On the differential equations of Hill in the theory of the motion of the Moon II, Acta Math. 95 (1956) 25–37.
[5] J.F. Steffensen, On the restricted problem of three bodies, Danske Vid. Selsk. Mat.-Fys. Medd. 30 (18) (1956), 17 pp.
[6] Y.F. Chang, G.F. Corliss, ATOMFT: solving ODEs and DAEs using Taylor series, Comput. Math. Appl. 28 (1994) 209–233.
[7] G.F. Corliss, Y.F. Chang, Solving ordinary differential equations using Taylor series, ACM Trans. Math. Software 8 (1982) 114–144.
[8] G.F. Corliss, A. Griewank, P. Henneberger, G. Kirlinger, F.A. Potra, H.J. Stetter, High-order stiff ODE solvers via automatic differentiation and rational prediction, in: Numerical Analysis and its Applications (Rousse, 1996), Lecture Notes in Comput. Sci., vol. 1196, Springer, Berlin, 1997, pp. 114–125.
[9] G.F. Corliss, G. Kirlinger, On implicit Taylor series methods for stiff ODEs, in: Computer Arithmetic and Enclosure Methods (Oldenburg, 1991), North-Holland, Amsterdam, 1992, pp. 371–379.
[10] J.D. Pryce, Solving high-index DAEs by Taylor series, Numer. Algor. 19 (1998) 195–211.
[11] J.D Pryce, A simple structural analysis method for DAEs, BIT 41 (2001) 364–394.
[12] W.G. Choe, J. Guckenheimer, Computing periodic orbits with high accuracy, Comput. Methods Appl. Mech. Engng. 170 (1999) 331–341.
[13] J. Guckenheimer, B. Meloon, Computing periodic orbits and their bifurcations with automatic differentiation, SIAM J. Sci. Comput. 22 (2000) 951–985.
[14] M. Lara, A. Deprit, A. Elipe, Numerical continuation of families of frozen orbits in the zonal problem of artificial satellite theory, Celestial Mech. Dynam. Astronom. 62 (1995) 167–181.
[15] N.S. Nedialkov, K.R. Jackson, G.F. Corliss, Validated solutions of initial value problems for ordinary differential equations, Appl. Math. Comput. 105 (1999) 21–68.
[16] J. Hoefkens, M. Berz, K. Makino, Computing validated solutions of implicit differential equations, Adv. Comput. Math. 19 (2003) 231–253.
[17] W. Tucker, A rigorous ODE solver and Smale's 14th problem, Found. Comput. Math. 2 (2002) 53–117.
[18] A. Griewank, Evaluating Derivatives, SIAM, Philadelphia, PA, 2000.
[19] L.B. Rall, Automatic Differentiation: Techniques and Applications, in: Lecture Notes in Computer Science, vol. 120, Springer-Verlag, Berlin, 1981.
[20] A. Jorba, M. Zou, A software package for the numerical integration of ODE by means of high-order Taylor methods, Technical Report, Department of Mathematics, University of Texas, TX, 78712-1082 (2001).
[21] M. Lara, A. Elipe, M. Palacios, Automatic programming of recurrent power series, Math. Comput. Simul. 49 (1999) 351–362.
[22] R. Barrio, F. Blesa, M. Lara, VSVO formulation of Taylor methods for the numerical solution of ODEs, Preprint 2003.
[23] R.A. Moore, Interval Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[24] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Different Algebraic Problems, in: Springer Ser. Comput. Math., vol. 14, Springer-Verlag, New York, 1991.

[25] D. Barton, On Taylor series and stiff equations, ACM Trans. Math. Software 6 (1980) 280–294.
[26] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, Springer Ser. Comput. Math., vol. 8, second ed., Springer-Verlag, New York, 1993.
[27] J. Binney, S. Tremaine, Galactic Dynamics, Princeton University Press, 1987.
[28] E.N. Lorentz, On the prevalence of aperiodicity in simple systems, in: M. Grmela, J.E. Marsden (Eds.), Global Analysis (Calgary 1978), Lecture Notes in Mathematics, vol. 755, 1979, pp. 53–75.
[29] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, in: Classics in Applied Mathematics, vol. 14, SIAM, Philadelphia, 1996.