An explicit one-step method for stiff problems

P. Novati
Università degli Studi dell'Aquila
Dipartimento di Matematica Pura ed Applicata
Via Vetoio, Coppito 67010 - L'Aquila - Italy
E-mail: novati@univaq.it

Abstract

In this paper we introduce an explicit one-step method that can be used for solving stiff problems. This method can be viewed as a modification of the explicit Euler method that allows to reduce the stiffness in some sense. Some numerical experiments on linear stiff problems and on the Van der Pol's equation show the effectiveness of the method.

1 Introduction

Given a function $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$, $x_0 \in \mathbb{R}$ and a vector $y_0 \in \mathbb{R}^N$, consider the initial value problem (IVP)

$$\left\{ \begin{array}{l} y'(x) = f(x, y(x)), \quad x > x_0, \\ y(x_0) = y_0. \end{array} \right. \tag{1}$$

As well known, if (1) is stiff, explicit methods generally provide a good approximation of the solution only if the integration step is chosen very small, and this choice is usually unfeasible from the computational viewpoint. For this reason, implicit methods are generally used to face such problems, but they require the solution of a nonlinear system of equation at each step. Obviously this represents a serious drawback if N is large.

When $f(x, y(x)) = Ay(x) + g(x)$, with $A \in \mathbb{R}^{N \times N}$ constant coefficients matrix, and $g : \mathbb{R} \to \mathbb{R}^N$, in order to overcome the drawbacks of the classical explicit and implicit methods, in recent years some authors introduced new kind of methods for (1), the so called exponential integrators, based on the polynomial approximation of the evolution operator $\exp(hA)$, where h is the step size, by means of a Krylov projection method (see e.g. [2], [3]), a series expansion method ([1], [5], [6], [10]), or an interpolation method ([4], [7]). Such methods generally performs better than classical explicit and implicit methods. Such improvement is substantially due to the fact that they are "problem dependent", in the sense that they provide a polynomial approximation $p_m(hA)$ of $\exp(hA)$ that depends on the spectral property of A. In particular, Krylov projection methods define p_m as the polynomial that interpolates the exponential function at the so called Ritz values of A. On the other hand, if Ω is a compact that approximates the convex hull of the spectrum of A, $\sigma(A)$, series expansion methods define p_m as the polynomial that approximates in some sense the exponential function in $h\Omega$, and interpolation methods define p_m as the polynomial that interpolates the exponential in a suitable set of points that depends on Ω. Anyway, these classes of methods present also some disadvantages. Indeed, Krylov projection methods are generally quite expansive, whereas series expansion methods and interpolation methods are cheaper but require the initial localization of $\sigma(A)$.

One of the most important features of these methods, that is of particular importance when they are applied to solve stiff problems, is that the coefficients of the polynomials p_m depend on the stepsize h. As consequence, the corresponding A-stability regions depend on h and the methods behave as they were A-stable (see e.g. [3]). For the method we are going to introduce, we want to maintain the above property of having the A-stability function depending on the stepsize and on the problem. This will constitute the relationship between our method and the exponential integrators.

Given $x_n > x_0$, $n = 1, 2, ...$, let y_n be the approximation of $y(x_n)$ provided by a certain discrete method. In the numerical implementation of any discrete method, one usually has to adopt a certain stepsize control procedure in order to maintain the local error less than a fixed tolerance δ, i.e.,

$$\|y_n - y(x_n)\| \leq \delta, \quad n = 1, 2, ... \tag{2}$$

This accuracy requirement obviously leads to a restriction of the stepsize $h_n = x_n - x_{n-1},$, that is, $h_n \leq h_{n,\delta}$, $n = 0, 1, ...$ If the problem is stiff, for any explicit method $h_{n,\delta}$ is forced to be less than a certain quantity h_s, i.e.,

$$h_{n,\delta} \leq h_s \tag{3}$$

that obviously depends on the stiffness and the method used. On the other side, working with an A-stable method one has more than what is necessary, because the condition (3) is not present, but, except for some special cases, the condition (2) does not allow to choose steps too large. Hence, the idea is to define an explicit method for which the stiffness does not force to maintain the step sizes less than a fixed quantity h_s. In other words, we want to create an explicit method where (3) is replaced by a restriction of the type

$$h_{n,\delta} \leq h_{n,s} \tag{4}$$

where $h_{n,s}$ can grow during the integration.

In order to do this, in this paper we introduce a new one-step explicit method that can be considered "problem dependent", designed in particular to face stiff IVPs. We consider a modification of the Euler method of the type

$$y_{n+1} = y_n + h_n \varphi(h_n, M_n) f(x_n, y_n), \tag{5}$$

where $M_n \in \mathbb{R}^{N \times N}$ and $\varphi : \mathbb{R} \times \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ is defined by

$$\varphi(h_n, M_n) := (1 + h_n) (1 + h_n M_n)^{-1}.$$

For a such kind of method, the A-stability region depends on h_n and M_n. As we shall see, the matrix M_n can be chosen in order to enlarge or also restrict the A-stability region dynamically during the integration in a close connection with the accuracy requirement (2). As already said, the aim is to create an explicit integrator for which the restriction (3) is remarkably relaxed. Indeed, we shall demonstrate that defining suitably the matrix sequence $\{M_n\}_{n\geq 0}$ the method (5) will be asymptotically stable under the stepsize condition

$$h_n \leq ch_{n-1}, \quad c > 1. \tag{6}$$

Because of (6), such a method can be considered as an intermediate approach, between the explicit and the backward Euler methods.

Regarding the construction of the matrix sequence $\{M_n\}_{n\geq 0}$, we want that such matrices have the properties of scaling the problem eliminating in some sense the stiffness, or, in other words, preconditioning the problem. As we shall see there are many ways to define such matrices, but most of them require additional work, such as approximating the inverse of the Jacobian of f, or approximating its eigenvalues. Since we want to avoid such computations, we shall create an automatic procedure that, starting from $M_0 = I_N$ (identity matrix of order N), dynamically update M_n using some informations on the local error. M_n will be maintained diagonal so that the computation of $\varphi(h, M_n)$ will not represent a significant additional cost. As we shall see, the only additional cost will be due to two further evaluation of f.

The paper is organized as follows. In Section 2 we define the scaled Euler method and in Section 3 we study the linear stability. In Section 4 we describe the algorithm that defines the matrix sequence $\{M_n\}_{n\geq 0}$ and study the asymptotic stability corresponding to this choice. In Section 5 we present some numerical details and the final algorithm that was used for the numerical experiments of Section 6. Finally, in Section 7 some ideas to extend the properties of the scaled Euler method to higher order one-step methods are presented.

2 The scaled Euler method

Consider the method (5)

$$y_{n+1} = y_n + h\varphi(h, M_n)f(x_n, y_n),$$

where h is the step size that, at the moment, we consider constant, $M_n \in \mathbb{R}^{N\times N}$ and $\varphi(h, M_n) \in \mathbb{R}^{N\times N}$. Clearly, depending on φ, such a method can be of order $p = 0$ or $p = 1$. In order to have $p = 1$, we must require that

$$\lim_{h\to 0} \varphi(h, M_n) = I_N, \tag{7}$$

where I_N denotes the identity matrix of order N. Moreover, with the aim of "scaling" the problem, we require that

$$\lim_{h\to \infty} \varphi(h, M_n) = M_n^{-1}. \tag{8}$$

The relations (7) and (8) lead to define

$$\varphi(h, M_n) := (1+h)\left(1+hM_n\right)^{-1}. \tag{9}$$

Due to the condition (8) we call *scaled Euler method* the corresponding method

$$y_{n+1} = y_n + h\,(1+h)\left(1+hM_n\right)^{-1} f(x_n, y_n). \tag{10}$$

Clearly, the choice (9) is only one among the possible choices of φ that satisfy the requirements (7) and (8).

3 Linear stability

In order to understand the reasons that lead us to introduce a method of type (10), and in particular, in order to understand the requirement (8), let us consider the linear stability properties of this method.

Consider the scalar test IVP

$$\begin{cases} y'(x) = \lambda y(x), & x > 0, \\ y(0) = 1, \end{cases} \tag{11}$$

where $\lambda \in \mathbb{C}^- := \{z \in \mathbf{C} : \operatorname{Re} z < 0\}$. Applying the scaled Euler method (10) to (11) with $M_n := M \in \mathbb{R}$, $M > 0$, and $\varphi : \mathbb{R}^2 \to \mathbb{R}$, the A-stability function is given by

$$R(h, \lambda, M) := 1 + h\lambda \frac{1+h}{1+hM}.$$

Hence, the corresponding A-stability region

$$S(h, M) := \{h\lambda \in \mathbf{C} : |R(h, \lambda, M)| \leq 1\}$$

depends on h and M, and it is easy to see that $S(h, M)$ is a circle of radius

$$\sigma(h, M) := \frac{1}{\varphi(h, M)} = \frac{1+hM}{1+h}$$

centered at the point $-\sigma(h, M)$. In this way, by (7), $S(h, M)$ tends to the A-stability region of the Euler method for $h \to 0$, and, for $h \to \infty$, $S(h, M)$ tends to the circle of radius M and centered in $-M$.

Just to have an example, if $\lambda = r\exp(i\theta)$, where $r > 0$, $\pi/2 < \theta < 3\pi/2$, defining

$$M := -\frac{r}{\cos\theta} > 0, \tag{12}$$

one gets a method that is asymptotically stable for each $h \leq 1$, because

$$\left| R(h, r\exp(i\theta), -\frac{r}{\cos\theta}) \right| \leq 1$$

for

$$h \leq h^*, \quad \text{with } \; h^* := \frac{1}{2}\left(1 + \sqrt{1 - \frac{8\cos\theta}{r}}\right) \geq 1,$$

for $r > 0$, $\pi/2 < \theta < 3\pi/2$. Hence, in a certain sense the stiffness has been eliminated, because we are not forced to use a small step even when r is very large, or, in other words, the choice of h has become independent of the problem.

Defining $M > -\frac{r}{\cos\theta}$ the A-stability region becomes larger, but the approximation of $\exp(h\lambda)$ becomes worse for small values of h. In fact, if we consider for instance the real case $\theta = \pi$, $\lambda = -r$, solving with respect to x the equation

$$\left|\exp(-hr) - \left(1 - hr\frac{1+h}{1+hx}\right)\right| = 0$$

we get

$$x = x(h,r) = 1 + \frac{r}{2} + \frac{1}{12}r(r+6)h + O(h^2).$$

Hence, defining

$$M = M(h) := 1 + \frac{r}{2} + \frac{1}{12}r(r+6)h, \tag{13}$$

that tends to $1 + r/2$ for $h \to 0$, the corresponding method is very efficient because it allows to get an approximation of the exponential of order 3

$$\exp(-hr) - \left(1 - hr\frac{1+h}{1+h\left(1+\frac{r}{2}+\frac{1}{12}r(r+6)h\right)}\right) = \frac{1}{12}c^3h^4 + O(h^5). \tag{14}$$

Moreover, with this choice of M depending on h, we obtain a method A_0-stable, because

$$\left|1 - hr\frac{1+h}{1+h\left(1+\frac{r}{2}+\frac{1}{12}r(r+6)h\right)}\right| \leq 1$$

for each $h \geq 0$, $r \geq 0$.

4 The definition of M_n

The expressions (12) and (13) given as possible definitions for M are interesting only from a theoretical viewpoint, because we were considering the simple scalar problem (11). For N-dimensional nonlinear problems (1), such expressions have no significance unless the problem is approximated using the Jacobian of f. In any case, even considering linear problems of the type

$$\begin{cases} y'(x) = Ay(x) + g(x), \\ y(x_0) = y_0, \end{cases} \tag{15}$$

where $A \in \mathbb{R}^{N\times N}$ and $\sigma(A) \subset \mathbb{C}^-$, the use of (12) or (13) would require the inversion of A, so that the method cannot be considered explicit anymore.

Of course, a possible alternative consists in using an approximation of A^{-1}, that, in the nonlinear case, would require the computation of the Jacobian of f and the approximation of its inverse at each step. Anyway, intuitively, such kind of approach should not provide an effective improvement with respect to any implicit method based on the same approximation. Another way consists in defining M equal to the diagonal matrix having $\|A\|$ (here $\|\cdot\|$ denotes a certain arbitrary natural matrix norm) as diagonal elements. This is equivalent to consider M scalar equal to $\|A\|$, and $\varphi : \mathbb{R}^2 \to \mathbb{R}$. If A is symmetric, by previous section we know that this choice guarantees asymptotic stability for $h \leq 1$, but can lead to poor results in the approximation of the components of the solution with respect to the smallest eigenvalues of A. In the nonlinear case this choice obviously requires the evaluation of the Jacobian. Another possible choice of M, consists in defining it as a diagonal matrix whose diagonal elements are approximation of the modulus of the eigenvalues of A. Also this choice could present a lot of problems for large systems, especially in the nonlinear case.

In order to avoid all these drawback, our idea is to define a sequence $\{M_n\}_{n\geq 0}$ of diagonal matrices without using any informations on f (or A in the linear case). Starting from M_0 equal to the identity matrix of order N (i.e., starting with the explicit Euler method), we want to define dynamically M_n by monitoring the local error.

Given y_n, h, M_n, let $\eta(x_n + h, h, M_n)$ and $\eta(x_n + h, h/2, M_n)$ be the approximations of $y(x_n + h)$ furnished by the method (10) with stepsizes h and $h/2$ respectively. As well known it is possible to estimate the local error by means of

$$e(x_n + h, h, M_n) := \eta(x_n + h, h, M_n) - \eta(x_n + h, h/2, M_n), \tag{16}$$

and we suppose that we are using a step size control technique that, by monitoring the quantities (16), allows to obtain h_n such that

$$\|e(x_n + h_n, h_n, M_n)\|_\infty \approx \delta \tag{17}$$

where δ is a fixed tolerance.

In order to define M_{n+1}, let $\gamma, \rho \in \mathbb{R}$ be such that $\gamma > 1$, $0 < \rho < 1$. We compute two further approximations $\eta(x_n+h_n, h_n, M')$ and $\eta(x_n+h_n, h_n/2, M')$ of $y(x_n + h_n)$ with $M' := M_n\gamma$. For $i = 1, 2, ..., N$, let $M_j^{(i)}$ be the i-th element of the diagonal of M_j, $j = 0, 1, ...,$ and let $v^{(i)}$ be the i-th element of a vector $v \in \mathbb{R}^N$. We define

$$M_{n+1}^{(i)} := \begin{cases} M_n^{(i)}\gamma & \text{if} \quad \left|e^{(i)}(x_n + h_n, h_n, M')\right| < \left|e^{(i)}(x_n + h_n, h_n, M_n)\right| \\ \max\left(1, M_n^{(i)}\rho\right) & \text{if} \quad \left|e^{(i)}(x_n + h_n, h_n, M')\right| > \left|e^{(i)}(x_n + h_n, h_n, M_n)\right| \end{cases} \tag{18}$$

In doing so, we create an automatic procedure that can be applied to both linear and nonlinear case, that requires at each step only two additional approximations, but no inversion nor eigenvalue estimation.

In order to understand how the definition (18) of the matrix sequence $\{M_n\}_{n\geq 0}$ reflects on the asymptotic stability of the method, examine the scalar test equation (11). Under the hypothesis that h_n has been chosen such that the relation (17) holds, if we define M_{n+1} as stated in (18), we want to understand how large it is possible to choose h_{n+1} in order that

$$\left|1 + h_{n+1}\lambda \frac{1 + h_{n+1}}{1 + h_{n+1}M_{n+1}}\right| \leq 1. \tag{19}$$

The following lemma can be demonstrate by direct computation.

Lemma 1 *Given $\delta, M_a, M_b > 0$, we have*

$$\left|1 + h\lambda\frac{1+h}{1+hM_a}\right| = \left|1 + h'\lambda\frac{1+h'}{1+h'M_b}\right| + \delta \tag{20}$$

for $h' > 0$ given by

$$h' := c(h, M_a, M_b)h - K\delta + O(\delta^2), \tag{21}$$

where $K > 0$ and

$$c(h, M_a, M_b) := \frac{1}{2h}\left(h\frac{1+h}{1+hM_a}M_b - 1 + \sqrt{\left(h\frac{1+h}{1+hM_a}M_b - 1\right)^2 + 4h\frac{1+h}{1+hM_a}}\right) \tag{22}$$

that satisfies the relations

$$c(h, M_a, M_b)\left\{\begin{array}{lll} \geq 1 & \textit{for} & M_b \geq M_a \\ < 1 & \textit{for} & M_b < M_a \end{array}\right.$$

and

$$\lim_{h\to\infty} c(h, M_a, M_b) = \frac{M_b}{M_a}.$$

Proposition 2 *For the (11) assume that at each step*

$$|e(x_k + h_k, h_k, M_k)| \leq \delta\,|y_k|\,, \quad k \geq 0.$$

If

$$\left|1 + h_n\lambda\frac{1+h_n}{1+h_nM_n}\right| \leq 1 \tag{23}$$

then

$$\left|1 + h_{n+1}\lambda\frac{1+h_{n+1}}{1+h_{n+1}M_{n+1}}\right| \leq 1$$

for

$$h_{n+1} := 2c(h_n, M_n, M_{n+1})h_n - \overline{K}\delta, \tag{24}$$

where $\overline{K} > 0$.

Proof. Since h_{n+1} is such that

$$|e(x_{n+1} + h_{n+1}, h_{n+1}, M_{n+1})| \leq \delta\,|y_{n+1}|\,,$$

we have

$$\begin{aligned}\left|\left(1 + h_{n+1}\lambda\frac{1+h_{n+1}}{1+h_{n+1}M_{n+1}}\right)y_{n+1}\right| &\leq |e(x_{n+1} + h_{n+1}, h_{n+1}, M_{n+1})| + \\ &\quad \left|\left(1 + \frac{h_{n+1}}{2}\lambda\frac{1+\frac{h_{n+1}}{2}}{1+\frac{h_{n+1}}{2}M_{n+1}}\right)y_{n+1}\right| \\ &\leq \left(\delta + \left|1 + \frac{h_{n+1}}{2}\lambda\frac{1+\frac{h_{n+1}}{2}}{1+\frac{h_{n+1}}{2}M_{n+1}}\right|\right)|y_{n+1}| \end{aligned} \tag{25}$$

Therefore, by (23) and Lemma 1 we get the thesis. ■

Hence, under the assumption that δ is chosen very small, without making any consideration about the values of γ and ρ, we can observe that if $M_{n+1} \geq M_n$, since $c(h_n, M_n, M_{n+1}) \geq 1$ for each value of h_n, we can double the stepsize maintaining the asymptotic stability requirement (19).

In order to define suitably the values of γ and ρ, it is important to investigate the behavior of the function $c(h, M_a, M_b)$. To this purpose, in the following two pictures we consider two plots of this function. In the first one (Fig.1) we consider $M_a > M_b$, and in the second one (Fig.2) $M_a < M_b$.

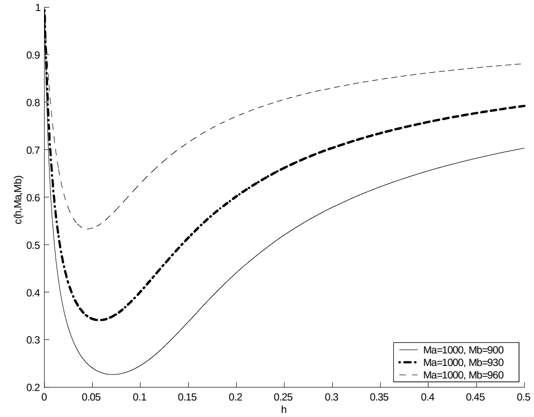

Fig.1

Looking at the case of $M_a > M_b$, we observe that the curve of $c(h, M_a, M_b)$ moves upward for $M_b \to M_a$. Since we want the restriction (24) not too strong, the idea is to define ρ and then M_{n+1} such that the curve remains above a certain value $1/2 < \alpha < 1$. In this way, by (24) we can maintain the asymptotic stability requirement with h_{n+1} not less than h_n. In order to do this, it is easy to show that the function

$$d(\psi) := c(h, M_a, M_a\psi), \quad \psi \geq 0,$$

is a monotone increasing function. Solving with respect to ψ the equation

$$d(\psi) - \alpha = 0,$$

we get

$$\psi = \psi(h, \alpha, M_a) = \frac{h^2\alpha^2 M_a + h\alpha M_a - 1 + \alpha - h + h\alpha^2}{h\alpha M_a(1+h)}. \tag{26}$$

This means that, given h_n, M_n, choosing ρ such that

$$\psi(h_n, \alpha, M_n) \leq \rho < 1 \tag{27}$$

and defining $M_{n+1} := M_n\rho$ allows to have $c(h_n, M_n, M_{n+1}) \geq \alpha$ and therefore the condition that ensure (19) becomes

$$h_{n+1} \leq 2\alpha h_n. \tag{28}$$

It is possible to demonstrate that, if for a given α and h, ψ is negative, then $c(h_n, M_n, M_{n+1}) \geq \alpha$ for each $\rho \geq 0$. Summing up the above arguments, in order to realize the restriction (28) with α fixed a priori, in general we cannot maintain ρ constant during the procedure, but we have to define it using (27).

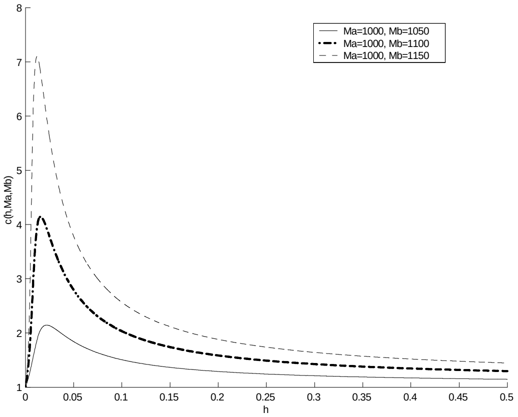

Fig.2

Regarding the case $M_a \leq M_b$, as stated in Lemma 1 we have $c(h, M_a, M_b) \geq 1$ and

$$\lim_{h \to \infty} c(h, M_a, M_b) = \frac{M_b}{M_a}.$$

As we can see in the picture above, the functions get a maximum value $m > \frac{M_b}{M_a}$ for small value of h. However, this does not happen when M_a is small, as we can see in the picture below (Fig.3).

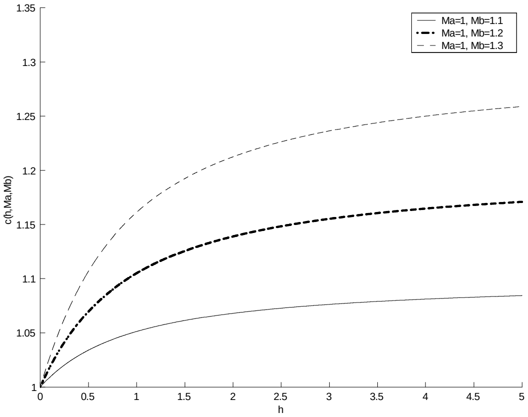

Fig.3

In fact, solving with respect to h the equation

$$c(h, M_a, M_b) = \frac{M_b}{M_a} =: \gamma,$$

we find

$$h = \frac{1}{\gamma(M_a - 1) - 1}. \tag{29}$$

Fixed γ, by (29) we must have

$$M_a \neq M_\gamma := \frac{\gamma + 1}{\gamma} > 1. \tag{30}$$

Hence, for each $h \geq 0$, we have that

$$c(h, M, M\gamma) < \gamma \quad \text{for } M \leq M_\gamma$$

and

$$\lim_{h \to \infty} c(h, M, M\gamma) = \gamma.$$

Since we start with $M_0 = 1$, for a given fixed γ, until M_n is no longer larger than M_γ we have $c > 1$, but not $c > \gamma$. If $M > M_\gamma$, since asymptotically $c(h, M, M\gamma) > \gamma$, we know that when the step is sufficiently large, the restriction (24) becomes

$$h_{n+1} \leq 2\frac{M_{n+1}}{M_n} h_n = 2\gamma h_n. \tag{31}$$

Alternatively, one can also require that

$$c(h_n, M_n, M_{n+1}) \geq \alpha > 1.$$

In order to obtain this, as before, one can chose $\gamma \geq \psi(h_n, \alpha, M_n)$ and define $M_{n+1} := M_n\gamma$ that lead to $h_{n+1} \leq 2\alpha h_n$. Anyway, this choice can be dangerous because it can lead to values for γ too large, that can determine instability.

5 Numerical implementation

Regarding the practical implementation of the scaled Euler method, the step size control procedure we use is the Richardson extrapolation as explained in [9]. In particular, let ε be the tolerance for the local error and let h_{n-1} be the previous stepsize used for computing y_n. On the basis of what stated in the previous section, we fix initially $h := 2\gamma h_{n-1}$ and compute $\eta(x_n + h, h, M_n)$ and $\eta(x_n + h, h/2, M_n)$. Then, we solve with respect to h' the equation

$$\frac{h}{h'} = \left[2\frac{\|e(x_n + h, h, M_n)\|_\infty}{\varepsilon} \right]^{\frac{1}{2}} \tag{32}$$

where $e(x_n + h, h, M_n)$ is defined by (16). If $h >> h'$ then we define $h := 2h'$ and recompute $\eta(x_n + h, h, M_n)$ and $\eta(x_n + h, h/2, M_n)$. We repeat the procedure until $h \approx 2h'$ and then define $h_n := h$ and the new approximation $y_{n+1} := \eta(x_n + h, h, M_n)$. Note that the last values $\eta(x_n + h_n, h_n, M_n)$ and $\eta(x_n + h_n, h_n/2, M_n)$ will be used to define M_{n+1}.

The following algorithm summarizes the practical implementation of the scaled Euler method with the adaptive construction of the matrix sequence $\{M_n\}_{n\geq 0}$.

```
Algorithm
1. given $\varepsilon$, $h_0$, $\gamma$, $\alpha$, $M_0$ equal to the identity matrix of order $N$, $n := 0$;
2. while $x_n \leq x_f$
    3. $n := n+1$, $h := 2\gamma h_{n-1}$;
    4. compute $\eta(x_n + h, h, M_n)$ and $\eta(x_n + h, h/2, M_n)$;
    5. compute $h'$ using (32);
    6. if $h >> 2h'$, then $h := 2h'$, go to 4;
    7. if $h \approx 2h'$, $y_{n+1} := \eta(x_n + h, h, M_n)$, $h_n := h$;
    8. define $M' := M_n\gamma$ and compute
           $\eta(x_n + h_n, h_n, M')$ and $\eta(x_n + h_n, h_n/2, M')$;
    9. for $i = 1, ..., N$, compute $\rho_i = \psi(h, \alpha, M_n^{(i)})$ and define
$$M_{n+1}^{(i)} := \begin{cases} M_n^{(i)}\gamma \quad \text{if} \quad \left|e^{(i)}(x_n + h_n, h_n, M')\right| < \left|e^{(i)}(x_n + h_n, h_n, M_n)\right| \\ \max\left(1, M_n^{(i)}\rho_i\right) \quad \text{if} \quad \left|e^{(i)}(x_n + h_n, h_n, M')\right| > \left|e^{(i)}(x_n + h_n, h_n, M_n)\right| \end{cases} ;$$
    10. $x_n := x_n + h_n$;
11. end
```

6 Numerical examples

In this section we want to test our method on some simple stiff problems. The aim is to put in evidence some of the characteristics of the scaled Euler method. Besides the trend of the step size, we want moreover to monitor the evolution of the matrix sequence $\{M_n\}_{n\geq 0}$.

Problem 1

In this first example we consider the scalar test equation (11) with $\lambda := -1000$. We want to integrate it from $x_0 = 0$ to $x_f = 400$, with initial condition $y(0) = 1$. Fixed the tolerance $\varepsilon = 10^{-5}$, in Fig.4 we can see the curve of the step size chosen by the method and the curve that represents the evolution of the sequence $\{M_n\}_{n\geq 0}$. The method is implemented with $\gamma = 1.1$ and $\alpha = 0.95$ and requires 124 steps.

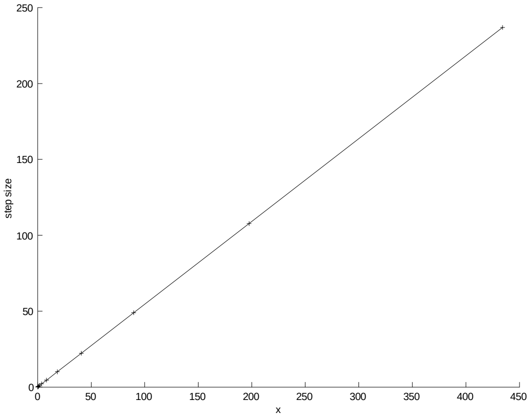

Fig.4-a Step size for Problem 1

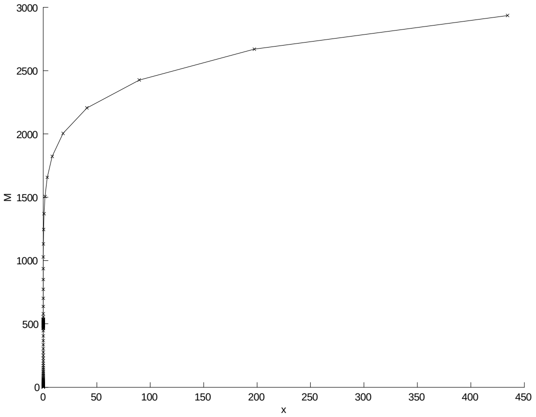

Fig.4-b $\{M_n\}_{n\geq 0}$ for Problem 1

As we can understand looking at Fig.4a, after the brief transient phase the step size become very large. In particular, during the stationary phase, the accepted steps grow with the relation $h_{n+1} = 2\gamma h_n$ (namely, with the maximum step size admitted), because $\{M_n\}_{n\geq 0}$ grows monotonically.

In the pictures below (Fig.5), we observe the accepted steps an the values of $\{M_n\}_{n\geq 0}$ during the brief transient phase. It is interesting to observe the condensation around the value 500 of the first values of the sequence $\{M_n\}_{n\geq 0}$. This is due to the approximation (14), and confirms that the way chosen to define such sequence is correct.

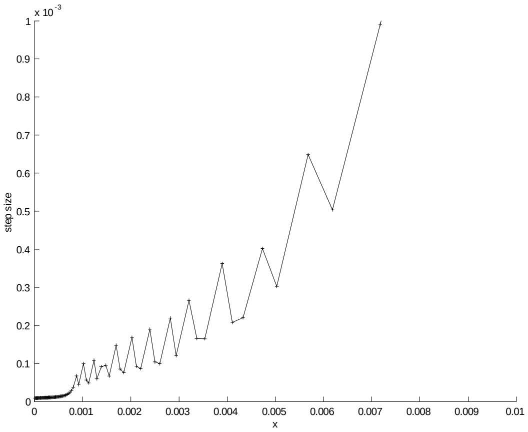

Fig.5-a Step size in the transient phase

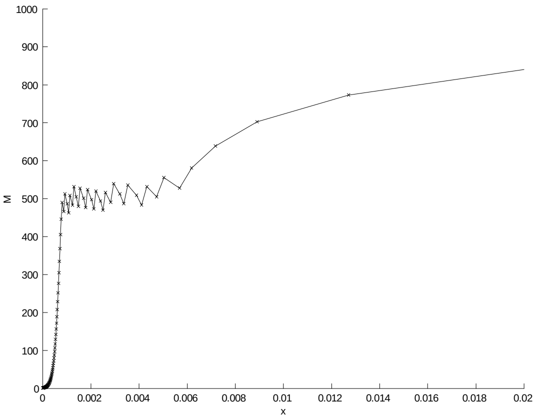

Fig.5-b $\{M_n\}_{n\geq 0}$ in the transient phase

In the complex case with $\lambda := -1000 + 500i$, integrating from 0 to 100 with the same initial condition and the same values of ε, γ and α, the Scaled Euler requires 234 steps, and the curve of the step size is very similar to that of the above real case. Indeed, in the stationary phase the accepted steps follow the relation $h_{n+1} = 2\gamma h_n$. During the transient phase, the evolution of $\{M_n\}_{n\geq 0}$ does not take into account of the imaginary part of λ, in the sense that there is a condensation around 500 as before.

Problem 2

Consider the problem

$$\left\{ \begin{array}{l} y'(x) = A(y(x) - vF(x)) + vF'(x), \\ y(x_0) = y_0, \end{array} \right.$$

where

$$A := \left[\begin{array}{cc} -1670 & 830 \\ 1660 & -840 \end{array} \right], \quad v := \left[\begin{array}{c} 1 \\ 1 \end{array} \right], \quad F(x) := \cos(x)\exp(-2x), \quad y_0 := \left[\begin{array}{c} 2 \\ 2 \end{array} \right].$$

The solution is given by $y(x) = \exp(Ax) + vF(x)$. The eigenvalues of A are $\lambda_1 = -2500$, $\lambda_2 = -10$. Integrating this equation from 0 to 100 with $\varepsilon = 10^{-5}$, in Fig.6-a we can see the step curve of the scaled Euler method implemented with $\gamma = 1.2$ and $\alpha = 0.95$.

Even if the last steps accepted are very large, the method requires 1293 steps to perform the integration. As we can understand looking at Fig.6-b, and remembering that we are working with a method of order 1, the large number of steps is due to long transient phase of the solution, caused by λ_2.

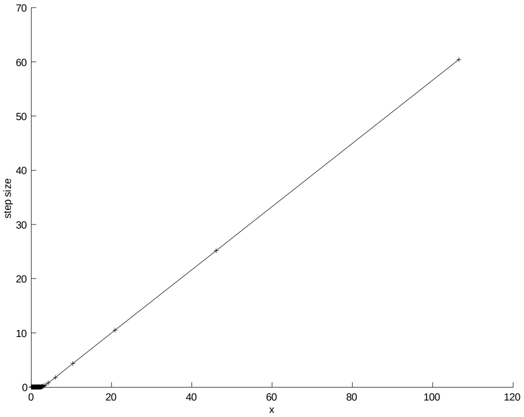

Fig.6-a Step size for Problem 2

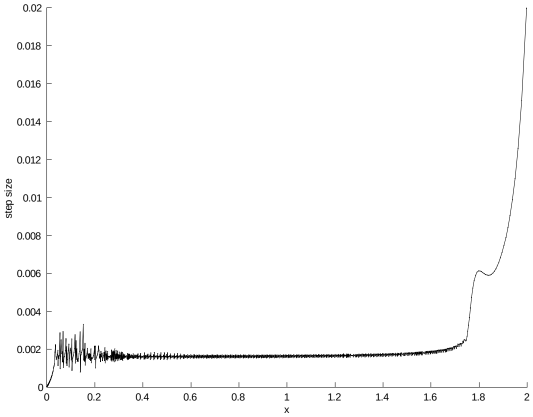

Fig.6-b Step size in the transient phase

Problem 3

Consider the heat equation

$$\begin{array}{rcl} \dfrac{\partial u}{\partial t} - \Delta u & = & 0 \quad in\ (0,T) \times \Omega, \\ u & = & 0 \quad on\ (0,T) \times \partial\Omega, \\ u|_{t=0} & = & u_0 \quad on\ \Omega, \end{array}$$

where Δ is the two dimensional Laplacian operator, $\Omega = (0,1)^2$, $\partial\Omega$ is the boundary of Ω and u_0 is a given function. Discretizing this equation with the method of lines using central differences on a uniform meshgrid of meshsize $h = 1/(n+1)$, we get an ordinary differential equation of the type

$$\begin{cases} y'(t) = Ay(t), \\ y(0) = y_0. \end{cases}$$

with $A \in \mathbb{R}^{n^2 \times n^2}$. As is well known, with the above discretization, A is a symmetric matrix and $\sigma(A)$ is contained in the real interval $[-4(n+1)^2(1 + \cos(\pi/(n+1)), -4(n+1)^2(1 - \cos(\pi/(n+1))]$. Choosing $n = 10$ (that implies $\lambda_{\min} \approx -948.4$ and $\lambda_{\max} \approx -19.6$) and initial condition $y_0 = (1,1,...,1)^T/n$ we integrate the above equation from 0 to 10. Fixed the tolerance $\varepsilon = 10^{-5}$ in Fig.7 is shown the curve of the step size chosen by the method, that is implemented with $\gamma = 1.05$ and $\alpha = 0.95$. The integration requires 314 steps.

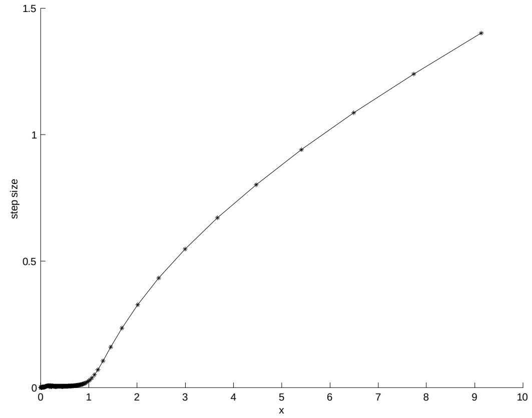

Fig.7 Step size for problem 3

As expected, the method is able to detect when all the components of the solution become smooth, allowing the growth of the step size.

Problem 4

In the last example we consider the van der Pol's equation

$$\begin{aligned} y_1'(x) &= y_2(x), \\ y_2'(x) &= \mu(1 - y_1^2(x))y_2(x) - y_1(x), \end{aligned}$$

with initial conditions $y_1(0) = 2$, $y_2(0) = 0$. As is well known, for large values of μ, the problem becomes very stiff. In our test we define $\mu = 500$. For this problem we choose $\varepsilon = 10^{-5}$, $\gamma = 1.05$ and $\alpha = 0.95$. Integrating this equation from 0 to 450, in Fig.8 is plotted the numerical solution of the equation. This integration requires about 9000 steps, but, as we shall see, this large number of steps is not due to the stiffness nor to the nonlinearity of the problem, but only to the order of the method.

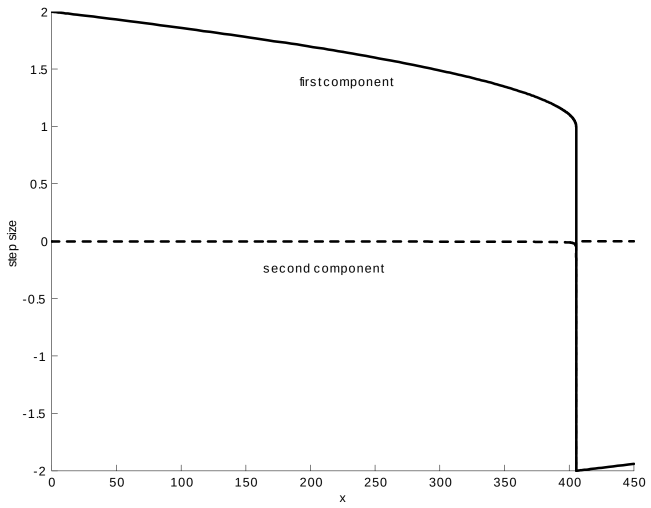

Fig.8 Numerical solution of Problem 4

In Fig.9 we can see the plots of the step size and of the sequence $\{M_n^{(2)}\}_{n\geq 0}$ (the sequence $\{M_n^{(1)}\}_{n\geq 0}$ is forced to stay closed to 1 on the whole interval). By comparing the solution with the pictures here below we can perfectly understand how the method works. As in the previous cases, we can observe that the stiffness is weakened by the method because when the solution is smooth, the method automatically allows the growth of the step size and the values of the sequence $\{M_n^{(2)}\}_{n\geq 0}$. On the other side, we can also observe that approaching the steepest part of the solution (around the point $x = 400$), the step size is drastically reduced and the sequence $\{M_n^{(2)}\}_{n\geq 0}$ is forced to go back to 1, so that the method behaves very similarly to the standard explicit Euler method.

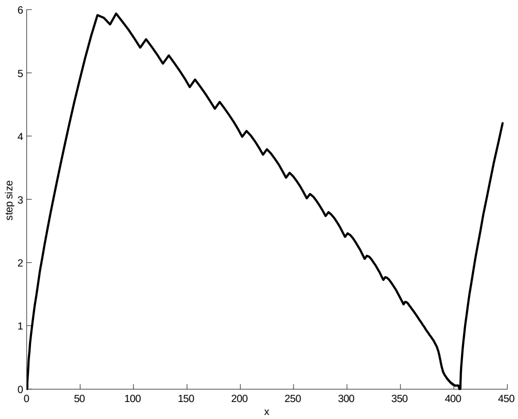

Fig.9-a Step size for Problem 4

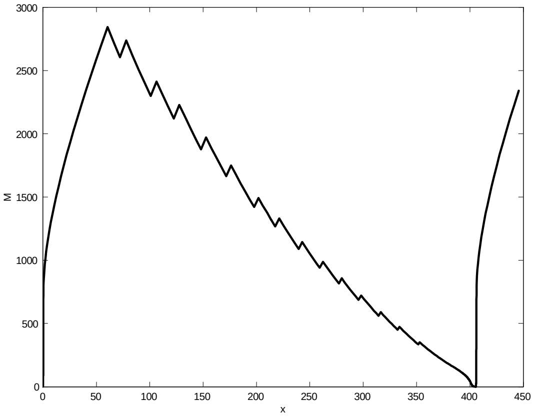

Fig.9-b $\{M_n^{(2)}\}_{n\geq 0}$ for Problem 4

7 Possible extension to higher order methods

The numerical experiments of previous section confirm that the scaled Euler method constitutes an interesting issue to face stiff problems. However, being a one-order method, it cannot be competitive with other higher order stiff solvers such as the well known Rosenbrock method, especially where the solution is highly varying. In order to overcome this drawback, in this section we want to propose an idea to extend the main features of the scaled Euler method to build higher order methods. The idea is to define a new class of explicit Runge-Kutta methods, or, to be more precise, to extend the class of Runge-Kutta methods.

As is well known, an explicit τ-stages Runge-Kutta method can be written in the following form

$$y_{n+1} = y_n + h \sum\nolimits_{j=1}^{\tau} c_j K_j, \tag{33}$$

where $c_j \in \mathbb{R}$, $j = 1, ..., \tau$, and, as usual,

$$\begin{aligned} K_1 &= f(x_n, y_n), \\ K_j &= f(x_n + a_j h, y_n + h \sum\nolimits_{s=1}^{j-1} b_{js} K_s), \quad j = 2, ..., \tau, \end{aligned} \tag{34}$$

with $a_j, b_{js} \in \mathbb{R},$, $j = 2, ..., \tau$, $s = 1, ..., j-1$. In order to face stiff problems, generalizing the construction of the scaled Euler method, the idea is to define $c_j = c_j(h, M)$, $j = 1, ..., \tau$, as suitable functions of h and M, where M is chosen to scale the problem and can also be dependent on n. The coefficients a_j and b_{js} can be maintained constant.

From now on we always assume $c_j = c_j(h, M)$, $j = 1, ..., \tau$. We want to define these functions such that for $\tau \leq 4$ the A-stability function of the corresponding method is of the form

$$\begin{aligned} \tau &= 2: \qquad R_2(h, \lambda, M) := 1 + h\lambda\varphi_2 + \frac{h^2\lambda^2}{2}\varphi_2^2 \\ \tau &= 3: \qquad R_3(h, \lambda, M) := 1 + h\lambda\varphi_3 + \frac{h^2\lambda^2}{2}\varphi_3^2 + \frac{h^3\lambda^3}{6}\varphi_3^3 \\ \tau &= 4: \qquad R_4(h, \lambda, M) := 1 + h\lambda\varphi_4 + \frac{h^2\lambda^2}{2}\varphi_4^2 + \frac{h^3\lambda^3}{6}\varphi_4^3 + \frac{h^4\lambda^4}{24}\varphi_4^4 \end{aligned}$$

where

$$\begin{aligned} \varphi_2 &= \varphi_2(h, M) := \frac{1 + h^2 M}{1 + h^2 M^2}, \\ \varphi_3 &= \varphi_3(h, M) := \frac{1 + h^3 M^2}{1 + h^3 M^3}, \\ \varphi_4 &= \varphi_4(h, M) := \frac{1 + h^4 M^3}{1 + h^4 M^4}. \end{aligned}$$

In this way, it is easy to prove that for $\tau \leq 4$ we obtain a method of order τ with an A-stability region that, if $M > 1$, is larger than the A-stability region of a τ-order explicit Runge-Kutta method.

Example 3 *Just to give an example, for $\tau = 2$ the order conditions become*

$$\begin{aligned} c_1 + c_2 &= \varphi_2, \\ a_2 c_2 &= \tfrac{1}{2}\varphi_2^2. \end{aligned}$$

With the above two conditions, we can create, for instance, the methods defined by the Butcher arrays

$$\begin{array}{ccc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ & \varphi_2\left(1-\varphi_2\right) & \varphi_2^2 \end{array} \quad , \quad \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ & \varphi_2\left(1-\frac{1}{2}\varphi_2\right) & \frac{1}{2}\varphi_2^2 \end{array}$$

that are obtained setting $a_2 = 1/2$ and $a_2 = 1$ and that can be viewed as a scaled generalized Euler method and as a scaled Heun method respectively.

A rigorous analysis of the ideas proposed in this section is now in progress. A number of numerical experiments in which the above two method were implemented replacing the constant M with the matrix sequence $\{M_n\}_{n\geq 0}$ defined using a stepsize control technique as for the scaled Euler method, have already revealed that these methods actually work very well.

8 Conclusions

The numerical experiments presented in the paper show that the scaled Euler method can be used to solve stiff problems. The method is able to detect where the solution is smooth, allowing the growth of the step size. Indeed, the A-stability region varies dynamically with the solution: it is large where the solution is smooth, and it collapses to the circle centered in -1 and radius 1 during the transient phases, in a close connection with the accuracy requirements. Since the method is explicit, the most important feature with respect to other stiff solvers regards the computational cost. Whenever one has to solve stiff initial value problems arising from the discretization of partial differential equations the use of a standard stiff solvers can be extremely expansive because of the dimension of the problem. If we consider, as example, the Rosenbrock method applied to problem 3 (as implemented in the ODE23S Matlab routine, see [8]) with the same accuracy requirement used above (i.e., the absolute tolerance $\varepsilon = 10^{-5}$), the integration requires 55 steps whereas the scaled Euler method 314. As is well known the computational cost of the Rosenbrock method is essentially that of an inversion (LU factorization) at each step. Hence, we have 55 inversions versus 314 matrix-vector multiplication; even considering the particular sparsity pattern of the matrix, the scaled Euler method is surely cheaper.

As mentioned in the introduction, in order to overcome the problem of the computational cost of the classical implicit methods, the exponential integrators based on Krylov projection techniques or other approximation techniques seems to be quite effective. However, contrary to the classical methods as well as the scaled Euler method, these methods are not able to face nonlinear problems unless a preliminary linearization is made.

Concluding, we want to say that when a rigorous analysis about the possible extension to higher order methods of the basic features of the scaled Euler method will be ready, such methods will constitute an effective alternative to the classical stiff solvers, especially for large dimensional problems. The basic point is that when solving a certain problem the accuracy requirement introduces an upper bound for the feasible step size. Hence, using a classical implicit method, the possibility of having arbitrary large step sizes (allowed by the fact that it makes inversions) is not so important. In the author's opinion, it is much more important to be able to solve a stiff problems efficiently, without solving one or more linear systems at each step.

Acknowledgement 4 *We would like to thank Stefano Maset and Lucio Torelli for many helpful discussions and comments.*

References

[1] L. Bergamaschi and M. Vianello, *Efficient computation of the exponential operator for large, sparse, symmetric matrices*, Numer. Linear Algebra Appl., 7 (2000), pp. 27-45 .

[2] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1236-1264.

[3] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911-1925.

[4] I. Moret and P.Novati, *An interpolatory approximation of the matrix exponential based on Faber polynomials*, J. Comput. App. Math., 131 (2001), pp. 361-380.

[5] I. Moret and P. Novati, *The computation of functions of matrices by truncated Faber series*, Numer. Func. Anal. and Optimiz., 22 (2001), pp. 697-719.

[6] P. Novati, *Solving linear initial valuue problems by Faber polynomials*, Numer. Linear Algebra Appl., 10 (2003), pp. 247-270.

[7] P. Novati, *A polynomial method based on Fejer points for the computation of functions of unsymmetric matrices*, Appl. Numer. Math., 44 (2003),.pp. 201-224.

[8] L. F. Shampine and M. W. Reichelt, *The MATLAB ODE Suite*, SIAM Journal on Scientific Computing, 18 (1997), pp. 1-22.

[9] J. Stoer and R. Bulirsch, *Introduction to numerical analysis.* Texts in Applied Mathematics. 12. New York: Springer-Verlag. xiii, 660 p. (1993).

[10] H. Tal-Ezer, *Spectral methods in time for parabolic problems*, SIAM J. Numer. Anal., 26 (1989), pp. 1-11.