On the starting algorithms for fully implicit Runge-Kutta methods

S. González–Pinto [1], J. I. Montijano [2] and S. Pérez–Rodríguez[1] *

[1] *Departamento Análisis Matemático, Universidad de La Laguna, Tenerife.*
38271, Spain. email: sgonzalez@ull.es

[2] *Departamento Matemática Aplicada, Universidad de Zaragoza.*
50009, Spain. email: monti@posta.unizar.es

Abstract

This paper is concerned with the behavior of starting algorithms to solve the algebraic equations of stages arising when fully implicit Runge-Kutta methods are applied to stiff initial value problems. The classical Lagrange extrapolation of the internal stages of the preceding step and some variants thereof that do not require any additional cost are analyzed.

To study the order of the starting algorithms we consider three different approaches. First we analyze the classical order through the theory of Butcher's series, second we derive the order on the Prothero and Robinson model [16] and finally we study the stiff order for a general class of dissipative problems. A detailed study of the orders of some starting algorithms for Gauss, Radau IA-IIA, Lobatto IIIA-C methods is also carried out. Finally, to compare the most relevant starting algorithms studied here, some numerical experiments on well known nonlinear stiff problems are presented.

AMS subject classification: 65L05

Keywords: Stiff Initial Value Problems; Implicit Runge-Kutta methods; Solution of Stage Equations; Starting algorithms

1 Introduction.

We consider the numerical solution of stiff initial value problems

$$y'(t) = f(t, y(t)), \quad y(0) = y_0 \in I\!R^m, \quad t \in [0, T], \tag{1.1}$$

where $f : [0, T] \times I\!R^m \to I\!R^m$ is assumed to be sufficiently smooth in a tubular neighborhood of the unique solution $y(t),\ t \in [0, T]$ of (1.1).

For the solution of (1.1) we consider implicit Runge–Kutta methods in which the time stepping from (t_0, y_0) to $(t_1 = t_0 + h, y_1)$ is given by

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f(t_0 + c_i h, X_i),$$

* This work was supported by project DGES PB97–1018

where the internal stages X_i are calculated from the system

$$X_i = y_0 + h\sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, X_j) \quad (i = 1, \ldots, s). \tag{1.2}$$

We will denote by $c = (c_1, \ldots, c_s)^T$ the node vector, by $b = (b_1, \ldots, b_s)^T$ the weight vector associated to the corresponding quadrature formula with s nodes and by $A = [a_{ij}] \in I\!R^{s\times s}$ the coefficient matrix of the Runge–Kutta method. As usual, it will be tacitly assumed that $Ae = c$, where $e = (1, \ldots, 1)^T \in I\!R^s$.

Once the algebraic system (1.2) has been solved, typically by some modified Newton iteration, we want to compute good approximations Y_i^0 to the internal Runge–Kutta stages Y_i of the next step $(t_1 = t_0 + h, y_1) \to (t_2 = t_1 + rh, y_2)$ to start the iteration process for the new system

$$Y_i = y_1 + \bar{h}\sum_{j=1}^{s} a_{ij} f(t_1 + c_j \bar{h}, Y_j) \quad (i = 1, \ldots, s), \tag{1.3}$$

where $r = \bar{h}/h$ is supposed to be of moderate size, usually $r \in (0, 2]$.

To this end we will consider the following two types of starting methods, that can be implemented with little or practically no additional computational cost:

Starting algorithms of type I

$$Y_i^0 = \gamma_i y_0 + \sum_{j=1}^{s} \alpha_{ij} X_j, \;\; i = 1, ..., s. \tag{1.4}$$

Starting algorithms of type II

$$Y_i^0 = y_0 + h\delta_i f(t_0, y_0) + h\sum_{j=1}^{s} \beta_{ij} f(t_0 + c_j h, X_j), \;\; i = 1, ..., s. \tag{1.5}$$

The coefficients $\{\gamma_i, \alpha_{ij}\}$ and $\{\delta_i, \beta_{ij}\}$ must only depend on the stepsize ratio r and on the coefficients of the RK method.

We will say that a starting algorithm (1.4) or (1.5) is of order q if this is the largest integer such that

$$\max_{1\le i\le s} |Y_i - Y_i^0| = \mathcal{O}(h^{q+1}).$$

Usually, the term $\mathcal{O}(h^{q+1})$ will depend on the successive elementary differentials of f, on the stepsize ratio r and on the coefficients of the RK method. On the other hand, we will say that the starting algorithm has stiff order q if $\max_{1\le i\le s} |Y_i - Y_i^0| = \mathcal{O}(h^{q+1})$ where the term $\mathcal{O}(h^{q+1})$ can depend on the successive derivatives of the exact solution $y(t)$ of (1.1), but it is independent of the elementary differentials of $f(t, y(t))$ when they are considered separately, that is, it does not depend on the stiffness of the problem.

Let us note that from (1.2), starting algorithms of type I are in fact a particular case of algorithms of type II with $\delta_i = 0, i = 1, \ldots, s$ if

$$\gamma_i + \sum_{j=1}^{s} \alpha_{ij} = 1. \tag{1.6}$$

In this case, denoting $\mathcal{A} = [\alpha_{ij}]$ and $\mathcal{B} = [\beta_{ij}]$, the coefficients of the corresponding algorithms are related by $\mathcal{B} = \mathcal{A}A$. Moreover, if A is nonsingular, all algorithm of type II with $\delta_i = 0, i = 1, \dots, s$ can also be written as an algorithm of type I with $\mathcal{A} = \mathcal{B}A^{-1}$. On the other hand, since $Y_i = y_0 + \mathcal{O}(h)$ and $X_i = y_0 + \mathcal{O}(h)$, condition (1.6) is required in order Y_i^0 to have at least order 0. Then, we will consider only starting algorithms of type I satisfying (1.6).

If matrix A is nonsingular, for computational purposes it is better to use an algebraically equivalent version of (1.5) which consists in eliminating, by using (1.2), the derivatives of the preceding step $f(t_0 + c_j h, X_j)$, and putting Y_i^0 as a linear combination of X_i, y_0 and $f(t_0, y_0)$. Moreover if the method is stiffly accurate (i. e., $b_j = a_{sj}, j = 1, \dots, s$) it is convenient to put $f(t_0, y_0)$ as a linear combination of the internal stages of the previous step with the purpose of reducing the amplification of the accumulated errors when we evaluate the derivative function. Nevertheless, for analysis purposes we prefer to keep the algorithm in the present form.

Starting algorithms of these types have already been considered by Nørsett and Thomsen in [14]. Likewise in [11, 12], similar algorithms have been studied, mainly to start the iterations of the RK Gauss methods applied to nonstiff hamiltonian problems. Also, in [11, 12] the author proposes some variants that consist in adding one or two additional evaluations of the derivative function with the purpose of gaining order and so reducing the number of iterations of the iterative process employed. On the other hand, Sand in [18] studied starting algorithms of type I for stiff problems, considering together with the usual order conditions (obtained by comparing the expansions in powers of h of the starting algorithms and those corresponding to the internal stages, by using the Butcher's series theory), what he calls inverse order conditions, i.e., expansions of the internal stages in powers of h^{-1}, which can make sense as recognized by Sand, when the spectral radius of the matrix $h^{-1}J^{-1}(t_0, y_0)$ is small (here $J = \partial f / \partial y$ denotes the Jacobian matrix). However, although he gives some suggestions concerning what conditions of order to choose by considering an SDIRK method taken from [14], it is not clear if those order conditions will be appropriate for fully implicit Runge-Kutta methods, such as those based on Gauss, Radau, Lobatto points, or collocation RK methods in general. Moreover the theory of the inverse order is not fully justified and the theory of the direct order (based on comparing the expansions in powers of h) can not be satisfactory for stiff problems.

In [10] several kinds of starting algorithms, not only polynomial extrapolation but also other implicit algorithms, are considered as predictor of certain iterated Runge-Kutta methods (oriented to a parallel computation) based on Gauss, Radau or Lobatto points, and the order of the resulting predictor-corrector methods is analyzed, as well as some relevant properties of linear stability. Our objective in this paper is quite different, because we are interested in using the starting algorithms in sequential computation.

On the other hand, in [9] Chap. IV.8 the authors recommend as starting values for the simplified Newton iteration the polynomial interpolation of Lagrange on the internal stages X_i and y_0, of the preceding step evaluated at the points $t = 1 + rc_i$, rather than taking $Y_i^0 = y_1$ (both cases are starting algorithms of *Type I*). The authors base their considerations on the performance of such a starting algorithm on a great variety of stiff problems as showed in the integrations carried out by their code RADAU5 (which uses the three–stage Radau IIA formula). It is widely recognized that the starting algorithms based on the polynomial interpolation of Lagrange of the internal stages of the preceding step X_i and y_0 yield good starting algorithms in general when the RK method is a collocation method. In this paper we pursue to give a theoretical support to this good behavior and moreover to analyze other starting algorithms of possible interest for the integration of stiff problems.

A study of starting algorithms for differential algebraic problems has also been done in [17] and [15].

The paper is organized as follows, in section 2 we study the classical order of the proposed starting algorithms through the Butcher series theory, in section 3 we consider the stiff order, by analyzing first the Prothero and Robinson model and concluding with some order results for nonlinear dissipative problems. In section 4 some numerical experiments are carried out in order to confirm the theoretical results obtained in previous sections and to gain some insight by comparing the most interesting starting algorithms.

2 Nonstiff order of the starting algorithms.

In this section we study the maximum attainable order for nonstiff problems of the starting algorithms defined in the previous section.

Here and in the rest of the paper we will use the typical simplifying order conditions

$$B(p): \;\; b^T c^{j-1} = 1/j, \; (1 \le j \le p) \quad C(q): \;\; A c^{j-1} = c^j/j, \; (1 \le j \le q),$$

where $c^j = (c_1^j, \ldots, c_s^j)^T$, and we will call the *stage order* of a RK (A, b) the largest integer τ such that $B(\tau)$ and $C(\tau)$ hold.

To simplify the study of the order we will first give the following Lemmas:

Lemma 2.1. *Let $1 \le q \le s$ and suppose that the a s-stage non-confluent RK (A,b) satisfies $B(q)$, $C(q)$. Then a starting algorithm of type II has order $\ge q{+}1$ if and only if for all $i = 1, \ldots, s$*

$$\delta_i + \beta_i^T e = (1 + rc_i), \tag{2.1}$$

$$\beta_i^T c^{j-1} = \Delta_{ij}(r), \quad j = 2, ..., q+1, \tag{2.2}$$

where $\beta_i^T = (\beta_{i1}, ..., \beta_{is})$ and $\Delta_{ij}(r) = b^T c^{j-1} + r A_i^T (e + rc)^{j-1}$ being $A_i^T = (a_{i1}, \ldots, a_{is})$ the i^{th}–row of matrix A.

Proof. If we consider in (1.2) y_0 as an additional stage $X_0 \equiv y_0$, the starting algorithm Y_i^0 can be seen as the approximation given by a Runge–Kutta method with Butcher tableau

$$\begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline Y_i^0 & \tilde{p}_i^T \end{array} \quad = \quad \begin{array}{c|cc} 0 & 0 & 0^T \\ c & 0 & A \\ \hline Y_i^0 & \delta_i & \beta_i^T \end{array} \tag{2.3}$$

Similarly, the solution Y_i of the implicit system (1.3) can be seen as the solution provided by a Runge–Kutta

$$\begin{array}{c|c} \bar{c} & \bar{A} \\ \hline Y_i & p_i^T \end{array} \quad = \quad \begin{array}{c|cc} c & A & 0 \\ e + rc & eb^T & rA \\ \hline Y_i & b^T & rA_i^T \end{array} \tag{2.4}$$

Now, comparing the Butcher's series for Y_i and Y_i^0 and taking into account conditions $B(q)$ and $C(q)$, the conditions for having order $q + 1$ reduce to (2.1), (2.2). □

Remark 2.1. Since starting algorithms of type I (satisfying (1.6)) are a particular case of algorithms of type II, conditions (2.1), (2.2) reduce in this case to

$$\gamma_i + \alpha_i^T e = 1, \quad \alpha_i^T A c^{j-1} = \Delta_{ij}(r), \quad 1 \le j \le q+1, \tag{2.5}$$

where $\alpha_i^T = (\alpha_{i1}, ..., \alpha_{is})$.

LEMMA 2.2. *Let us assume that an s-stage RK(A, b) fulfills $B(p)$ and $C(q)$. Then,*

$$\Delta_{ij}(r) = \frac{(1 + rc_i)^j}{j}, \quad i = 1, ...s, \; j = 1, ..., u.$$

with $u = \min\{p, q\}$.

PROOF. It follows immediately from the definition of $\Delta_{ij}(r)$ after using $B(p)$ and $C(q)$. □

LEMMA 2.3. *Let a s-stage non-confluent RK (A,b) be satisfying $B(p)$ and $C(q)$ with $p, q \geq s - 1$. Let us denote $V = [e, c, ..., c^{s-1}]$ and $e_1^T = (1, 0, ..., 0)$. Then*

$$A \textit{ is singular } \;\Leftrightarrow\; e_1^T V^{-1} A c^{s-1} = 0.$$

PROOF. If $s = 1$ it is trivial. For $s > 1$ we denote $K = e_1^T V^{-1} A c^{s-1}$. Since all knots c_i are different the matrix V is nonsingular, hence there exist real numbers λ_j; $j = 0, \ldots, s$ such that

$$Ac^{s-1} = \sum_{j=0}^{s-1} \lambda_j c^j.$$

On the other hand, $V^{-1} c^j = e_{j+1}^T$ where $e_j^T = (0, ..., \overset{(j)}{1}, ..., 0) \in \mathbb{R}^s$. From here it follows that

$$K = e_1^T \sum_{j=0}^{s-1} \lambda_j V^{-1} c^j = e_1^T \sum_{j=0}^{s-1} \lambda_j e_{j+1} = \lambda_0.$$

Now, using $C(s - 1)$ we can put

$$A[e, c, \ldots, c^{s-1}] = [c, c^2, \ldots, c^{s-1}, e] \left[\begin{array}{ccccc} 1 & 0 & \cdots & 0 & \lambda_1 \\ 0 & 1/2 & 0 & \cdots & \lambda_2 \\ & & & & \\ 0 & & 0 & 1/(s-1) & \lambda_{s-1} \\ 0 & & & 0 & \lambda_0 \end{array} \right],$$

and taking determinants we conclude the proof. □

THEOREM 2.4. *Let us assume that a s-stage non-confluent RK method satisfies $B(s - 1)$, $C(s - 1)$. Then, for the starting algorithms of type I, we have that*

(a) *There exists a s-parameter family of order $s - 1$ (with γ_j, $j = 1, \ldots, s$ as parameters) which is determined by the linear system*

$$(2.6) \qquad \alpha_i^T e = 1 - \gamma_i; \quad \alpha_i^T c^j = (1 + rc_i)^j; \quad 1 \leq i \leq s, \;\; 1 \leq j \leq s - 1.$$

In particular, by choosing $\gamma_j = 0; j = 1, \ldots, s$, the coefficients of the resulting starting algorithm of order $s - 1$ are given by,

$$(2.7) \qquad \alpha_{ij} = \bar{l}_j(1 + rc_i), \quad 1 \leq i, j \leq s,$$

where $\{\bar{l}_j(t); j = 1, \ldots, s\}$ is the Lagrange basis associated to the knots $\{c_1, \ldots, c_s\}$.

(b) If A is singular, the starting algorithm cannot reach order s.

(c) If A is nonsingular, there exists a unique starting algorithm of maximal order s. Moreover, if $B(s)$ and $C(s)$ are fulfilled such a starting method is given by

$$\begin{aligned} \gamma_i &= l_0(1+rc_i), \quad 1 \le i \le s, \\ \alpha_{ij} &= l_j(1+rc_i), \quad 1 \le i,j \le s, \end{aligned} \tag{2.8}$$

where $\{l_0(x), l_1(x), ..., l_s(x)\}$ is the Lagrange basis associated to the knots $\{c_0 = 0, c_1, ..., c_s\}$.

Proof.
(a) It is an immediate consequence of lemma 2.1, remark 2.1 and lemma 2.2, together with the condition $c_i \neq c_j$ for $i \neq j$.

Moreover, if we choose $\gamma_i = 0, i = 1, \dots, s$ the linear system (2.6) reduces to

$$\alpha_i^T c^j = (1+rc_i)^j, \; j = 0, \dots, s-1, \;\; (i = 1, \dots, s). \tag{2.9}$$

On the other hand, the Lagrange basis associated to the knots $\{c_1, \dots, c_s\}$

$$\bar{l}_j(t) = \pi(t)/((t-c_j)\pi'(c_j)), \quad j = 1, \dots, s; \quad \pi(t) = (t-c_1)\cdots(t-c_s) \tag{2.10}$$

satisfies

$$t^j = \sum_{k=1}^{s} \bar{l}_k(t) c_k^j, \quad j = 0, 1, ..., s-1. \tag{2.11}$$

Then, by taking $t = 1 + rc_i;\; i = 1, ..., s$ it follows by the uniqueness of solution of (2.9) that $\alpha_{ik} = \bar{l}_k(1 + rc_i);\; i, k = 1, \dots, s$.
(b) We proceed by contradiction. Let us assume that A is singular and that the order s is reached for the starting algorithm, then the parameters $\{\alpha_{ij}, \gamma_i\}$ must satisfy the linear system (2.6) and the equation in (2.5) corresponding to $j = s$ in remark 2.1, i.e.

$$\alpha_i^T A c^{s-1} = \Delta_{is}(r). \tag{2.12}$$

Denoting $V = [e, c, \dots, c^{s-1}]$, $R_i^T = \left(1, (1+rc_i), \dots, (1+rc_i)^{s-1}\right)$ and $e_1^T = (1, 0, \dots, 0)$, conditions (2.6) can be written in the form

$$\alpha_i^T V = R_i^T - \gamma_i e_1^T, \quad i = 1, \dots, s \tag{2.13}$$

Now, multiplying these equations by the vector $V^{-1}Ac^{s-1}$ and substituting in (2.12) we get

$$R_i^T V^{-1} A c^{s-1} - \gamma_i e_1^T V^{-1} A c^{s-1} = \Delta_{is}(r). \tag{2.14}$$

By lemma 2.3, since A is singular, it follows that $e_1^T V^{-1} A c^{s-1} = 0$, hence

$$R_i^T V^{-1} A c^{s-1} = b^T c^{s-1} + r A_i^T (e + rc)^{s-1}, \;\; \forall r > 0. \tag{2.15}$$

Since the left side of this equation is a polynomial in r of degree at most $s-1$, we get that $A_i^T c^{s-1} = 0$ for all $i = 1, ..., s$, i.e., $Ac^{s-1} = 0$. This implies from (2.15) that $b^T c^{s-1} + r A_i^T (e + rc)^{s-1} \equiv 0$, which leads to $A \equiv 0$. But this is impossible if $s > 1$ because $Ae = c$, and for $s = 1$ it is also impossible because in this case $X_1 = y_0$ which implies from (2.6) that $Y_i^0 = y_0, i = 1, \dots, s$, resulting a starting algorithm of order $s = 0$, but not of order $s = 1$.

(c) Since A is nonsingular it follows from lemma 2.3 that $e_1^T V^{-1} A c^{s-1} \neq 0$, then the only starting algorithm of order s (at least) satisfies (2.13)-(2.14), for each $i = 1, \dots, s$. Thus, it is explicitly determined by

$$\gamma_i = \frac{R_i^T V^{-1} A c^{s-1} - \Delta_{is}(r)}{e_1^T V^{-1} A c^{s-1}}, \tag{2.16}$$

and by the linear system in (2.13). Let us see now that order $s+1$ is impossible to reach (in general). To get order $s+1$ the method must satisfy, together with (2.13) and (2.16) the order condition

$$\alpha_i^T A c^s = b^T c^s + r A_i^T (e + rc)^s.$$

This implies that $Ac^s = 0$, since the left side in the above equation is a polynomial in r of degree $\leq s$, whereas the second side is a polynomial of degree $s+1$. Then A must vanish, which is impossible.

On the other hand if we also assume $B(s)$ and $C(s)$, we have by lemma 2.2 that $\Delta_{is}(r) = (1 + rc_i)^s / s$. Then, the conditions to reach order s reduce to the linear systems (for each $i = 1, \dots, s$) of dimension $s+1$

$$(\gamma_i, \alpha_i^T) W = v_i^T, \quad i = 1, \dots, s \tag{2.17}$$

where

$$W = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ e & c & \cdots & c^s \end{bmatrix}; \quad v_i^T = (1, (1 + rc_i), \cdots, (1 + rc_i)^s). \tag{2.18}$$

Proceeding in a similar way as in part *(a)* of the proof, it is easy to conclude that the Lagrange basis associated to the knots $\{c_0 = 0, c_1, \dots, c_s\}$ satisfies the equations (2.17)-(2.18), and from the uniqueness of solution of (2.17) the proof is complete. □

Theorem 2.5. *Let us suppose that an s-stage non-confluent RK method satisfies $B(s-1)$, $C(s-1)$. Then, for the starting algorithms of type II, we have that*

(a) There exists a s-parameter family of order s (with δ_j $j = 1, \dots, s$ as parameters), which is determined by the linear system

$$\begin{aligned} \beta_i^T e &= (1 + rc_i) - \delta_i, \\ \beta_i^T c^{j-1} &= \frac{(1 + rc_i)^j}{j}, \quad j = 2, ..., s-1, \\ \beta_i^T c^{s-1} &= \Delta_{is}(r). \end{aligned} \tag{2.19}$$

Moreover, if $B(s)$ and $C(s)$ are fulfilled and we choose $\delta_j = 0$; $j = 1, \dots, s$, the only starting algorithm of maximal order s, is given by

$$\beta_{ij} = \int_0^{1 + rc_i} \bar{l}_j(t) dt, \quad 1 \leq i, j \leq s,$$

where $\{\bar{l}_j(t); j = 1, \dots, s\}$ is the Lagrange basis associated to the knots $\{c_1, \dots, c_s\}$. If A is nonsingular such a starting algorithm coincides with the one of type I given in theorem 2.4-(c)

(b) Assuming $B(s)$ and $C(s)$, there exists a unique starting algorithm of order $s+1$ if and only if A is nonsingular or the considered Runge–Kutta is the explicit Euler method.

PROOF.

(a) The first part is an immediate consequence of lemma 2.1 and lemma 2.2. Moreover, if $B(s)$ and $C(s)$ are satisfied then conditions (2.19) can be written

$$\beta_i^T V = (1+rc_i, (1+rc_i)^2/2, ..., (1+rc_i)^s/s); \quad i = 1, \dots, s. \tag{2.20}$$

The Lagrange basis (2.10) associated to the knots $\{c_j;\ j = 1, \dots, s\}$ satisfies (2.11), hence integration of this last equation on $[0, 1+rc_i]$ yields

$$\sum_{k=1}^{s} c_k^{j-1} \int_0^{1+rc_i} \bar{l}_k(t)dt = (1+rc_i)^j/j; \quad i, j = 1, \dots, s.$$

Now, from the uniqueness of solution of the linear system (2.20) it follows that

$$\beta_{ik} = \int_0^{1+rc_i} \bar{l}_k(t)dt; \quad 1 \le i, k \le s.$$

If the RK matrix A is nonsingular, we have that this starting algorithm is equivalent to that one of *type I* given in theorem 2.4–*(c)*, since both starting algorithms can be expressed in the form

$$Y_i^0 = y_0 + h \sum_{j=1}^{s} \beta_{ij} f(t_0 + hc_j, X_j); \quad 1 \le i, j \le s,$$

for adequate parameters β_{ij} and both are of order s. Hence the equivalence follows immediately from the uniqueness of solution of (2.20).

(b) The conditions to reach order $s+1$ for the starting algorithm are given (bearing in mind $B(s)$, $C(s)$ and lemma 2.1) by

$$\delta_i e_1^T + \beta_i^T V = \bar{R}_i^T \equiv (1+rc_i, (1+rc_i)^2/2, ..., (1+rc_i)^s/s); \; 1 \le i \le s, \tag{2.21}$$

and by the equation $\beta_i^T c^s = \Delta_{i,s+1}$. This last equation can be replaced (by using (2.21)) by

$$\delta_i e_1^T V^{-1} c^s + \Delta_{i,s+1} = \bar{R}_i^T V^{-1} c^s. \tag{2.22}$$

Then, from lemma 2.3 if A is nonsingular, $\bar{K} = e_1^T V^{-1} c^s \neq 0$ and the coefficients of the only starting algorithm of order $\ge s+1$ are given by,

$$\left\{ \begin{array}{l} \delta_i = \dfrac{\bar{R}_i^T V^{-1} c^s - \Delta_{i,s+1}(r)}{e_1^T V^{-1} c^s}, \\ \beta_i^T = \bar{R}_i^T V^{-1} - \delta_i e_1^T V^{-1} \end{array} \right. \qquad 1 \le i \le s. \tag{2.23}$$

Conversely, if the starting algorithm has order $s+1$, (2.21) and (2.22) are satisfied. Now, if A is singular then the equation (2.22) becomes,

$$\Delta_{i,s+1} = \bar{R}_i^T V^{-1} c^s, \quad 1 \le i \le s.$$

By comparing the coefficient of r^{s+1} in the polynomials appearing in both sides of the last equation we conclude that $Ac^s = 0$. Since the matrix V is nonsingular, there exist real numbers λ_i such that,

$$c^s = \sum_{j=1}^{s} \lambda_j c^{j-1}. \tag{2.24}$$

¿From here we can put,

$$0 = Ac^s = \sum_{j=1}^{s} \lambda_j A c^{j-1} = \sum_{j=1}^{s} \frac{1}{j} \lambda_j c^j,$$

which after replacing c^s gives,

$$0 = \lambda_1 \lambda_s e + \sum_{j=1}^{s-1} (\frac{s}{j} \lambda_j + \lambda_s \lambda_{j+1}) c^j.$$

It follows that $\lambda_j = 0,\ (j = 1, \ldots, s)$, hence $c^s = 0$ or equivalently $c = 0$. Now, if $s > 1$ this is impossible, and if $s = 1$, we have that $A = c = 0$ and since $B(1)$ is satisfied, the RK method must be the Euler explicit method.

To see that the order $s+2$ cannot be reached when A is nonsingular, we will show that if we add the order conditions corresponding to the power $s+2$ of h, we arrive to a contradiction. Observe that this would imply

$$\beta_i^T A c^s = b^T A c^s + r A_i^T \left(e b^T c^s + r A (e + rc)^s \right), \quad i = 1, \ldots, s. \tag{2.25}$$

By (2.23) the left side of (2.25) is a polynomial of degree $\leq s+1$, which implies that $A^2 c^s = 0$ or equivalently $c = 0$, which is impossible. □

Remark 2.2. For non-confluent s-stage RK methods satisfying $C(s),\ B(s)$ and with matrix A nonsingular the starting algorithm of type II with maximal order $s+1$, can be calculated by the alternative formula (more practical than (2.23))

$$Y_i^0 = y_1 + rh \sum_{j=1}^{s} a_{ij} \mathcal{F}_j, \quad i = 1, ..., s, \tag{2.26}$$

where

$$\mathcal{F}_j = P(1 + rc_j),\ j = 1, ..., s$$

being $P(x)$ the interpolation polynomial of the derivatives of the preceding step, i.e.,

$$P(0) = f(t_0, y_0);\ \ P(c_j) = f(t_0 + c_j h, X_j),\ j = 1, \ldots, s.$$

This can be easily verified by putting the starting algorithm (2.26) as one of type II and showing that it fulfills the order equations given by (2.21) and $\beta_i^T c^s = \Delta_{i,s+1}$.

3 Stiff order of the starting algorithms.

The order results derived for the starting algorithms in the preceding section are based on the Butcher's series theory and are in principle relevant only for nonstiff problems. It is well known that Runge–Kutta methods can give certain order reduction phenomenon when they are applied to stiff systems and therefore a different analysis of the order must be done in this case. As a first approach to the stiff case, we will study the behavior of the starting algorithms when the RK method is applied to the linear Prothero and Robinson equation (see e.g. Hairer & Wanner [9], chapter IV.15). After that, we will study the order of the algorithms on a more general class of nonlinear stiff problems.

3.1 The Prothero and Robinson model.

Let us consider the Prothero and Robinson model,

$$y' = f(t,y) = \lambda(y-\phi(t)) + \phi'(t), \quad y(0)=\phi(0), \quad Re(\lambda) \le 0, \tag{3.1}$$

where $\phi(t)$ is assumed to be smooth enough in $[0,T]$ (for instance, we can consider that $\phi(t)$ is an analytical function). If we apply a RK $(\hat{A},\hat{b})$ to advance from the point (t_0,y_0) to $t_1 = t_0+h$, we get (see e.g. [13]):

$$y_{RK}(t_0+h) \equiv y_1 = \hat{R}(z)(y_0-\phi(t_0)) + \phi(t_0) + \sum_{j\ge1} \frac{\phi^{(j)}(t_0)}{j!} v_j(z) h^j$$

where $z=\lambda h$, $\hat{R}(z)$ is the linear stability function of the RK method, i.e.

$$\hat{R}(z) = 1 + z\hat{b}^T(I-z\hat{A})^{-1}e, \tag{3.2}$$

and the coefficients $v_j(z)$ are given by

$$\left\{ \begin{array}{l} v_j(z) = -z\mu_j + j\mu_{j-1}, \quad j=1,2,... \\ \mu_j = \mu_j(z) = \hat{b}^T(I-z\hat{A})^{-1}\hat{c}^j, \quad j=0,1,... \end{array} \right. \tag{3.3}$$

Then, since the solution of the problem at t_0+h can be expressed as

$$y(t_0+h) = \phi(t_0+h) = \phi(t_0) + \sum_{j\ge1} \frac{\phi^{(j)}(t_0)}{j!} h^j$$

the error is given by

$$y(t_1) - y_1 = \hat{R}(z)(\phi(t_0)-y_0) + d_h(t_0)$$

where $d_h(t_0) = \sum_{j\ge1} \phi^{(j)}(t_0)(1-v_j(z))h^j/j!$ is the local error and the term $\hat{R}(z)(\phi(t_0)-y_0)$ represents the propagation of the previous error.

If the RK method $(\hat{A},\hat{b})$ is AS-stable, that is, $\sup_{Re(z)\le0} |z\hat{b}^T(I-z\hat{A})^{-1}|_2 < \infty$, then it is clear that the coefficients v_j are uniformly bounded in $\mathrm{Re}\, z \le 0$ and the order of the local error $d_h(t_0)$ is independent of the stiffness of the problem. In the following we will assume that the Runge–Kutta method (A,b) considered is AS and ASI–stable, i.e., $\sup_{Re(z)\le0} |zb^T(I-zA)^{-1}|_2 < \infty$ and $\sup_{Re(z)\le0} |(I-zA)^{-1}|_2 < \infty$ (to see some aspects concerning AS and ASI–stability, see for instance [1], [2]).

On the other hand, each internal stage Y_i in (1.3) can be seen as a RK method with Butcher tableau given by (2.4). Thus, for the Prothero & Robinson model we have

$$\left. \begin{array}{c} Y_i = \bar{R}_i(z)(y_0-\phi(t_0)) + \phi(t_0) + \displaystyle\sum_{j\ge1} \frac{\phi^{(j)}(t_0)}{j!} v_{i,j}(z) h^j, \\ v_{i,j}(z) = -z\mu_{i,j} + j\mu_{i,j-1}, \quad j\ge1, \\ \mu_{i,k} = p_i^T(I-z\bar{A})^{-1}\bar{c}^k, \quad k\ge0 \\ \bar{R}_i(z) = 1 + zp_i^T(I-z\bar{A})^{-1}e. \end{array} \right\} \tag{3.4}$$

If the RK (A,b) method is AS and ASI–stable, then $\bar{R}_i(z)$ is uniformly bounded on $\mathrm{Re}\, z \le 0$. In fact, a direct calculation gives

$$\begin{array}{c} (I-z\bar{A})^{-1} = \begin{pmatrix} (I-zA)^{-1} & 0 \\ u(z) & (I-zrA)^{-1} \end{pmatrix}, \\ \\ u(z) = z(I-zrA)^{-1}eb^T(I-zA)^{-1} \end{array} \tag{3.5}$$

and from here we can get after some calculations that

$$\bar{R}_i(z) = R(z)(1 + zrA_i^T(I - zrA)^{-1}e),$$

where $A_i^T = e_i^T A$ and since $1 + zA_i^T(I - zA)^{-1}e = e_i^T(I + zA(I - zA)^{-1})e = e_i^T(I - zA)^{-1}e$, then

$$\bar{R}_i(z) = R(z)R_i(rz) \tag{3.6}$$

with

$$R(z) = 1 + zb^T(I - zA)^{-1}e, \qquad R_i(z) = e_i^T(I - zA)^{-1}e,$$

that are uniformly bounded.

Moreover, under the same conditions we can assure that,

$$\sup_{Re(z)\le 0} \max_{1\le i\le s} |v_{ij}| \le K_j < \infty, \quad (j \ge 1).$$

To prove this, it is enough to see that $|z\mu_{ij}|$ is uniformly bounded on $Re(z) \le 0$ for any $1 \le i \le s$ and any fixed $j \ge 1$. To this end, it suffices to show that the euclidean norm of the following $s \times 2s$ matrix

$$M(z) \equiv z[eb^T, rA](I - z\bar{A})^{-1}$$

is uniformly bounded on $Re(z) \le 0$. After some manipulations and by using (3.5) we get,

$$M(z) = [e(zb^T(I - zA)^{-1}) + (zrA(I - zrA)^{-1})e(zb^T(I - zA)^{-1})\,,\, zrA(I - zrA)^{-1}].$$

Now the AS and the ASI-stability of the RK (A,b) yield the desired bound.

On the other hand, the approximation Y_i^0 given by the starting algorithm (1.5) can be considered as the approximation given by the RK method (2.3) and therefore

$$\left.\begin{array}{c} Y_i^0 = \tilde{R}_i(z)(y_0 - \phi(t_0)) + \phi(t_0) + \displaystyle\sum_{j\ge 1} \frac{\phi^{(j)}(t_0)}{j!}\tilde{v}_{i,j}(z)h^j, \\ \tilde{v}_{i,j}(z) = -z\tilde{\mu}_{i,j} + j\tilde{\mu}_{i,j-1}, \;\; j \ge 1, \\ \tilde{\mu}_{i,k} = \tilde{p}_i^T(I - z\tilde{A})^{-1}\tilde{c}^k, \;\; k \ge 0 \\ \tilde{R}_i(z) = 1 + z\tilde{p}_i^T(I - z\tilde{A})^{-1}e \end{array}\right\} \tag{3.7}$$

Taking into account the value of $\tilde{p}_i$ in (2.3), the coefficients $\tilde{\mu}_{i,j}$ can be expressed in terms of the coefficients of the starting algorithm as

$$\begin{array}{l} \tilde{\mu}_{i,0} = \delta_i + \beta_i^T(I - zA)^{-1}e \\ \tilde{\mu}_{i,j} = \beta_i^T(I - zA)^{-1}c^j, \text{ for } j \ge 1 \end{array}$$

and the amplifying error function as

$$\tilde{R}_i(z) = 1 + z\delta_i + z\beta_i^T(I - zA)^{-1}e. \tag{3.8}$$

¿From (3.7) and (3.4), the error $Y_i - Y_i^0$ can be written

$$Y_i - Y_i^0 = \big(\bar{R}_i(z) - \tilde{R}_i(z)\big)(y_0 - \phi(t_0)) + \sum_{j\ge 1} \frac{\phi^{(j)}(t_0)}{j!}\big(v_{i,j} - \tilde{v}_{i,j}(z)\big)h^j.$$

To analyze the behavior of the starting algorithms, we will focus on three particular points: the boundedness of the coefficients $\tilde{v}_{i,j}(z)$, the order of the algorithm, that is, the maximum $j \geq 1$ such that $\bar{v}_{i,j}(z) - \tilde{v}_{i,j}(z) = 0$ for $i = 1, \ldots, s$ and for all z with $\operatorname{Re} z \leq 0$, and the boundedness of the difference $\bar{R}_i(z) - \tilde{R}_i(z)$.

We will use the following lemma in the proofs of the next theorems.

Lemma 3.1. *If the s–stage RK (A, b) satisfies the conditions $C(q)$, then*

(a) $(I - zA)^{-1}(-zc^j + jc^{j-1}) = jc^{j-1}, \ 1 \leq j \leq q.$

(b) $(I - zrA)^{-1}\left[-z(e + rc)^j + j(e + rc)^{j-1}\right] = -z(I - zrA)^{-1}e + j(e + rc)^{j-1}$, $1 \leq j \leq q$.

Proof.

The first statement es equivalent to

$$-zc^j + jc^{j-1} = (I - zA)jc^{j-1}, \ 1 \leq j \leq q,$$

which follows immediately from $C(q)$.

In the same way, the second statement is equivalent to

$$-z(e + rc)^j + j(e + rc)^{j-1} = -ze + j(I - zrA)(e + rc)^{j-1}, \ 1 \leq j \leq q,$$

which after some simplifications is equivalent to

$$(e + rc)^j = e + jrA(e + rc)^{j-1}, \ 1 \leq j \leq q.$$

The last equation follows from $C(q)$ by expanding the powers of the binomials. □

Theorem 3.2. *Let us suppose that a s-stage RK (A,b) is ASI–stable. Then the coefficients $\tilde{v}_{i,j}$ in (3.7) satisfy*

$$\sup_{Re(z) \leq 0} \max_{1 \leq i \leq s} |\tilde{v}_{ij}| \leq \tilde{K}_j < \infty, \quad \forall j \geq 1$$

if some of the following conditions are fulfilled:

a) The coefficients β_{ij} of the starting algorithm satisfy $\beta_i^T = \alpha_i^T A$ $(i = 1, \ldots, s)$ for some vectors α_i^T.

b) The RK method is non-confluent and satisfies $C(s-1)$ with $c_i = 0$ for some $1 \leq i \leq s$.

Proof. If $\beta_i^T = \alpha_i^T A$, then $z\tilde{\mu}_{ij} = z\beta_i(I - zA)^{-1}c^j = z\alpha_i A(I - zA)^{-1}c^j$ and since $zA(I - zA)^{-1} = \big((I - zA)^{-1} - I\big)$, the ASI–stability gives the boundedness. If $c_i = 0$ for some i, then for all $j \geq s$, c^j can be written as a linear combination of $c, c^2, \ldots, c^{s-1}$ and consequently, $\tilde{\mu}_{ij}$ can be written as a linear combination of $\tilde{\mu}_{i1}, \ldots, \tilde{\mu}_{i,s-1}$. But by $C(s-1)$, $c^j = jAc^{j-1}$ for $1 \leq j \leq s-1$ and

$$z\tilde{\mu}_{ij} = j\beta_i((I - zA)^{-1} - I)c^{j-1}, \quad 1 \leq j \leq s-1,$$

which are uniformly bounded on $Re(z) \leq 0$ by the ASI-stability. Now it is straightforward to prove the uniform boundedness on $Re(z) \leq 0$ of $|z\tilde{\mu}_{ij}|$ for $1 \leq i \leq s$ and any fixed $j \geq 1$. □

Remark 3.1.

- Starting algorithms of type I are a particular case of algorithms of type II with $\beta_i^T = \alpha_i^T A$. Then for this kind of algorithms the coefficients $\tilde{v}_{i,j}$ are uniformly bounded for all $1 \leq i \leq s$ and $j \geq 1$.

- If the matrix A of the method is nonsingular, then for each β_i^T there exists a vector α_i^T such that $\beta_i^T = \alpha_i^T A$ and in consequence each starting algorithm of Type II has its coefficients $\tilde{v}_{i,j}$ uniformly bounded.

Lemma 3.3. *If a s-stage non-confluent RK (A,b) satisfies $B(q)$ and $C(q)$ $(1 \leq q \leq s)$. Then, a starting algorithm of type II with coefficients $\tilde{v}_{i,j}$ uniformly bounded has order stiff q for the Prothero and Robinson equation if and only if*

$$\begin{aligned} &\delta_i + \beta_i^T e = (1 + rc_i), \\ &\beta_i^T c^{j-1} = \Delta_{ij}(r) = (1 + rc_i)^j / j, \quad j = 2, \ldots, q; \;\; i = 1, \ldots, s. \end{aligned} \tag{3.9}$$

Proof. Since the coefficients $\tilde{v}_{i,j}$ are uniformly bounded, the algorithm will have order stiff q if

$$v_{i,j}(z) - \tilde{v}_{i,j}(z) = 0, \; \forall Re(z) \leq 0, \quad j = 1, ..., q, \tag{3.10}$$

with $v_{i,j}(z)$ and $\tilde{v}_{i,j}(z)$ given by (3.4) and (3.7) respectively. On the other hand, it is not difficult to see that

$$v_{i,j} - \tilde{v}_{i,j} = d_i^T (I - z\bar{A})^{-1}(-z\bar{c}^j + j\bar{c}^{j-1}) - \delta_i \delta_{j1}, \;\; j \geq 1,$$

with $d_i^T = (b^T - \beta_i^T, rA_i^T)$ and $\delta_{j1} = 1$ if $j = 1$ and 0 otherwise.

Now, from (3.5)

$$\begin{aligned} v_{i,j} - \tilde{v}_{i,j} \;\; &= (b^T - \beta_i^T)(I - zA)^{-1}(-zc^j + jc^{j-1}) \\ &+ rA_i^T u(z)(-zc^j + jc^{j-1}) \\ &+ rA_i^T (I - zrA)^{-1}\left[-z(e + rc)^j + j(e + rc)^{j-1}\right] - \delta_i \delta_{j1}. \end{aligned} \tag{3.11}$$

and by using lemma 3.1 we get after some calculations that (3.10) is fulfilled if and only if (3.9) is satisfied. □

Theorem 3.4. *If a s-stage non-confluent RK (A, b) fulfills*
(i) the simplifying conditions $B(s-1), C(s-1)$,
(ii) the method is ASI-stable and AS-stable.
Then,

(a) The s–parameter family of starting algorithms of type I of order $s-1$ in part (a) of theorem 2.4 has also order stiff $s-1$ for the Prothero and Robinson equation.

(b) If A is nonsingular there exists a unique starting algorithm of type I of order stiff $s-1$ that reaches order s for quadratures, i.e. for the case $\lambda = 0$ (its coefficients are given by (2.6) and (2.16)). If moreover, $B(s)$ and $C(s)$ are fulfilled such a starting algorithm reaches the maximal order s (for any $Re(z) \leq 0$) and its coefficients are given by,

$$\gamma_i = l_0(1 + rc_i), \quad \alpha_{ij} = l_j(1 + rc_i), \quad 1 \leq i, j \leq s,$$

where $\{l_0(t), \ldots, l_s(t)\}$ is the Lagrange basis associated to the knots $\{c_0 = 0, c_1, ..., c_s\}$.

Proof.
(a) It is an immediate consequence of theorem 3.2, remark 3.1, and lemma 3.3 with $q = s - 1$.

(b) If the matrix A is nonsingular the equations to reach order $s-1$ for all $Re(z) \leq 0$ and order s for $z=0$ are given by (2.6) and (2.16). If we moreover assume $B(s)$ and $C(s)$, then by using lemma 3.3 with $q=s$ and proceeding as in theorem 2.4–*(c)* we complete the proof. □

THEOREM 3.5. *Let us suppose that we are in the same conditions as in theorem 3.4 and let us also assume that one of the conditions a) or b) in theorem 3.2 is satisfied. Then for the starting algorithms of type II we have:*

(a) There exists a $2s$*-parameter family of order stiff* $s-1$ *for the Prothero and Robinson equation. If moreover we ask for order* s *for quadratures (i.e. for* $z=0$*) we get the* s*-parameter family given in theorem 2.5-(a).*

(b) If $C(s)$ *and* $B(s)$ *are assumed then the above* s*-parameter family reaches order stiff* s*. Moreover, if we choose* $\delta_i = 0$ $(i=1,\ldots,s)$*, the coefficients of the starting algorithm of maximal order* s *are given by,*

$$\beta_{ij} = \int_0^{1+rc_i} \bar{l}_j(t)dt, \quad 1 \leq i,j \leq s,$$

where $\{\bar{l}_j(t); j=1,\ldots,s\}$ *is the Lagrange basis associated to the knots* $\{c_1,\ldots,c_s\}$*, as in theorem 2.5.*

(c) By assuming $C(s)$*,* $B(s)$ *and that the matrix* A *is nonsingular. Then the order* $s+1$ *cannot be reached in general for all* $Re(z) \leq 0$*. However, if we require order* s *in general, and order* $s+1$ *at the particular points* $z=0$ *or well at* $z=\infty$*, we get a unique starting algorithm.*

PROOF.

(a) ¿From lemma 3.3 the order $s-1$, independent of the stiffness, is equivalent to (3.9) with $q=s-1$ (which is also equivalent to (2.1)-(2.2) with q = s-2). Thus, the order conditions reduce to a linear system of $s-1$ equations with $s+1$ unknowns (for each $i=1,\ldots,s$), which can be written as follows

$$\beta_i^T V^* = R_i^{*T} - \delta_i e_{s-1,1}^T, \quad V^* = [e, c, ..., c^{s-2}],$$

$$R_i^{*T} = (1+rc_i, ..., (1+rc_i)^{s-1}/(s-1)), \quad e_{s-1,1}^T = (1,0,...,0) \in I\!R^{s-1}.$$

Since the range of matrix V^* is $s-1$, we have a $2s$-parameter family of starting algorithms.

On the other hand, if we require order s for quadratures, i.e. for $z=0$, we have from (3.5) that,

$$v_{i,s}(0) - \tilde{v}_{i,s}(0) = s\left(b^T c^{s-1} + rA_i^T(e+rc)^{s-1} - \beta_i^T c^{s-1}\right) = 0.$$

Hence the equations (2.1)–(2.2) for $q=s-1$ are fulfilled. This drive us to the same starting algorithms as in theorem 2.5–*(a)*.

(b) If we assume $B(s)$ and $C(s)$ then the conditions for classical order s are equivalent to (3.9) with $q=s$. From here we get a s-parameter family, with δ_i $(i=1,\ldots,s)$ as parameters. Hence, if we choose $\delta_i=0$ $(i=1,\ldots,s)$, we have the starting algorithm given by theorem 2.5- *(a)* that of course has order s on the Prothero and Robinson model.

(c) If moreover the matrix A is nonsingular, to get order $s+1$ in $z=0$ it is necessary that

$$v_{i,s+1}(0) - \tilde{v}_{i,s+1}(0) = (s+1)(b^T c^s + rA_i^T(e+rc)^s - \beta_i^T c^s) = 0$$

which drive us to the order equations (2.1)-(2.2) with $q = s$. The existence and uniqueness of solution, which is given by (2.23), is assured by the non-singularity of the matrix A.

In order to study the order $s+1$ for $z = \infty$, after some calculations we get that $v_{i,s+1}(\infty) - \tilde{v}_{i,s+1}(\infty) = 0$ if and only if

$$\beta_i^T A^{-1} c^{s+1} = b^T A^{-1} c^{s+1} - A_i^T A^{-1} e b^T A^{-1} c^{s+1} + A_i^T A^{-1} (e + rc)^{s+1}$$

which, bearing in mind that $A_i^T A^{-1} = (0, ..., 0, \overset{(i}{1}, 0, ..., 0)$, is equivalent to

$$\beta_i^T A^{-1} c^{s+1} = (1 + rc_i)^{s+1}, \quad 1 \leq i \leq s, \tag{3.12}$$

Hence, the order s in general and the order $s+1$ at the infinity point is equivalent to (3.9) with $q = s$ and (3.12). It is not difficult to see that these equations have a unique solution given by

$$\begin{aligned} \delta_i &= \frac{\bar{R}_i^T V^{-1} A^{-1} c^{s+1} - (1 + rc_i)^{s+1}}{e_1^T V^{-1} A^{-1} c^{s+1}}, & \beta_i^T &= \bar{R}_i^T V^{-1} - \delta_i e_1^T V^{-1}, \\ V &= [e, c, \ldots, c^{s-1}], & \bar{R}_i^T &= (1 + rc_i, \ldots, (1 + rc_i)^s / s). \end{aligned} \tag{3.13}$$

In order to see that the order $s+1$ cannot be reached on the Prothero and Robinson model, we proceed by contradiction. If there exists a starting algorithm of *type II* of order $s+1$, since it reaches order $s+1$ at $z = 0$ and at $z = \infty$, we get after equaling the expressions of δ_i in the equations (3.13) and (2.23) the following (we denote $K_1 = e_1^T V^{-1} c^s$ and $K_2 = e_1^T V^{-1} A^{-1} c^{s+1}$)

$$K_2(\bar{R}_i^T V^{-1} A^{-1} c^{s+1} - (1 + rc_i)^{s+1}) = K_1(\bar{R}_i^T V^{-1} c^s - \Delta_{i,s+1}),$$

that is,

$$\begin{aligned} \bar{R}_i^T V^{-1}(K_2 A^{-1} c^{s+1} - K_1 c^s) = \ & K_2 (1 + rc_i)^{s+1} \\ & - K_1 (b^T c^s + r A_i^T (e + rc)^s), \ \forall r. \end{aligned} \tag{3.14}$$

Now comparing the coefficients of the leading terms in the polynomials (on r) of the last equation we get $K_2 c_i^{s+1} - K_1 A_i^T c^s = 0$ for every $i = 1, ..., s$, i.e. $K_2 c^{s+1} = K_1 A c^s$. Which gives $K_2 A^{-1} c^{s+1} = K_1 c^s$, hence the left side of (3.14) is null. So we can put

$$K_2 (1 + rc_i)^{s+1} - K_1 (b^T c^s + r A_i^T (e + rc)^s) = 0, \ \ \forall r > 0,$$

or equivalently

$$(K_2 - K_1 b^T c^s) + \sum_{j=1}^{s} \binom{s+1}{j} r^j c_i^j \left(K_2 - \frac{K_1}{s+1} \right) + r^{s+1} (K_2 c_i^{s+1} - K_1 A_i^T c^s) \equiv 0.$$

This would imply $B(s+1)$ and $C(s+1)$, since K_1 must be nonzero (see lemma 2.3 and recall that the matrix A is nonsingular). This fact yields a contradiction because $B(s+1)$ and $C(s+1)$ cannot be satisfied for any s-stage Runge-Kutta method. □

Comparing the amplifying error functions.

The amplifying functions of the exact solution Y_i, $\bar{R}_i(z)$, and the approximation Y_i^0, $\tilde{R}_i(z)$, provided by the starting algorithm are given by (3.6) and (3.8) respectively:

$$\begin{aligned}\bar{R}_i(z) &= R(z)R_i(z) = \big(1 + zb^T(I - zA)^{-1}e\big)\big(e_i^T(I - zA)^{-1}e\big)\\ \tilde{R}_i(z) &= 1 + z\delta_i + z\beta_i^T(I - zA)^{-1}e.\end{aligned}$$

For the particular case of starting algorithms of type I ($\delta_i = 0$ and $\beta_i^T = \alpha_i^T A$), the amplifying function $\hat{R}_i(z)$ is

$$\hat{R}_i(z) = 1 + z\alpha_i^T A(I - zA)^{-1}e$$

If the RK method is AS and ASI–stable, it is clear that $\bar{R}_i(z)$ and $\hat{R}_i(z)$ are uniformly bounded on Re $z \leq 0$. However, the amplifying function of type II algorithms can not be bounded if some $\delta_i \neq 0$ and A is nonsingular (If the matrix A were singular it is possible the uniform boundedness of $\tilde{R}_i(z)$, but in this case δ_i and β_i must satisfy a certain equation, we see more about this in the next theorem for the case of Lobatto IIIA methods).

This fact may lead us to discard this kind of algorithms but if the RK(A,b) is stiffly accurate, A-stable and its matrix A is nonsingular, the global errors for the Prothero and Robinson equation satisfy $\phi(t_n) - y_n = \mathcal{O}(z^{-1})$ (see [9], pp. 225–227) and consequently the product $\tilde{R}(z)(\phi(t_n) - y_n)$ is bounded when z goes to infinity. Thus, starting algorithms of type II with unbounded amplifying functions can be suitable for practical purposes when they are used with stiffly accurate methods, such as Radau IIA, Lobatto IIIC and Lobatto IIIA.

The next theorem give us the behavior of the amplifying error functions evaluated at the infinity point for different RK methods based on quadratures of high order.

Theorem 3.6. *(a) For the RK Gauss, Radau IA, Radau IIA and Lobatto IIIC of s stages (and in general for any RK with A nonsingular) we have for* $1 \leq i \leq s$ *that,*

$$\bar{R}_i(\infty) = 0, \quad \hat{R}_i(\infty) = \gamma_i, \quad \tilde{R}_i(\infty) = \begin{cases} 1 - \beta_i^T A^{-1}e & \text{if } \delta_i = 0 \\ \infty & \text{if } \delta_i \neq 0. \end{cases}$$

(b) For Lobatto IIIA methods of s stages and by denoting

$$A = \begin{pmatrix} 0 & 0^T \\ w & \bar{A} \end{pmatrix}, \quad w^T = (a_{21}, \ldots, a_{s1}), \quad \bar{A} = [a_{ij}]_{i,j=2}^s,$$

$$\bar{b}^T = (b_2, ..., b_s), \quad \bar{c} = (c_2, ..., c_s)^T,$$

$$e_{s-1,j}^T = (0, ..., \overset{(j)}{1}, ..., 0) \in I\!R^{s-1}, \quad \beta_i^T = (\beta_{i1}, \bar{\beta}_i^T)$$

we have for every $2 \leq i \leq s$ *that*

$$\bar{R}_i(\infty) = (-1)^s e_{s-1,i-1}^T \bar{A}^{-1}w, \quad \hat{R}_i(\infty) = 1 - \bar{\alpha}_i^T \bar{A}^{-1}\bar{c},$$

and

$$\tilde{R}_i(\infty) = \begin{cases} 1 - \bar{\beta}_i^T \bar{A}^{-2}\bar{c} & \text{if } \delta_i = -\beta_{i1} + \bar{\beta}_i^T \bar{A}^{-1}w \\ \infty & \text{otherwise.} \end{cases}$$

Proof.
(a) It follows by taking limits when $z \to \infty$ in the expressions of the amplifying functions.

(b) For the Lobatto IIIA methods, since A is singular, we need some extra calculations, so we can put

$$(I - zA)^{-1} = \begin{pmatrix} 1 & 0^T \\ z(I - z\bar{A})^{-1}w & (I - z\bar{A})^{-1} \end{pmatrix},$$

$$(I - zA)^{-1}e = \begin{pmatrix} 1 \\ (I - z\bar{A})^{-1}(zw + e) \end{pmatrix}.$$

Since $\bar{A}$ is nonsingular and $\bar{A}e + w = \bar{c}$, it follows that

$$(I - z\bar{A})^{-1}(zw + e) = -\bar{A}^{-1}w - z^{-1}\bar{A}^{-2}\bar{c} + \mathcal{O}(z^{-2}). \tag{3.15}$$

On the other hand, for these methods is well known that $R(\infty) = (-1)^{s-1}$, where $R(z)$ is the linear stability function of the method. Moreover,

$$R_i(z) = e_{s,i}^T(I - zA)^{-1}e = e_{s-1,i-1}^T(I - z\bar{A})^{-1}(zw + e), \quad 2 \leq i \leq s.$$

¿From here it immediately follows that

$$\bar{R}_i(\infty) = R(\infty)R_i(\infty) = (-1)^s e_{s-1,i-1}^T\bar{A}^{-1}w, \quad 2 \leq i \leq s.$$

For the starting algorithms of *type I* we have, by putting $\alpha_i^T = (\alpha_{i1}, \bar{\alpha}_i^T)$, that

$$\hat{R}_i(z) = 1 + z\alpha_i^T A(I - zA)^{-1}e = 1 + z\bar{\alpha}_i^T\left(w + \bar{A}(I - z\bar{A})^{-1}(zw + e)\right),$$

which after using (3.15) gives $\hat{R}_i(\infty) = 1 - \bar{\alpha}_i^T\bar{A}^{-1}\bar{c}, \quad 2 \leq i \leq s$. For the starting algorithms of *type II* we have

$$\tilde{R}_i(z) = 1 + z\delta_i + z(\beta_{i1}, \bar{\beta}_i^T)(I - zA)^{-1}e = 1 + z\delta_i + z\beta_{i1} + z\bar{\beta}_i^T(I - z\bar{A})^{-1}(zw + e).$$

Now by using (3.15) it follows

$$\tilde{R}_i(z) = z(\delta_i + \beta_{i1} - \bar{\beta}_i^T\bar{A}^{-1}w) + (1 - \bar{\beta}_i^T\bar{A}^{-2}\bar{c}) + \mathcal{O}(z^{-1}), \quad 2 \leq i \leq s,$$

and from here we conclude the proof. □

3.2 Stiff order on dissipative problems.

Now, we are going to study the stiff order of the starting algorithms of type I and II on a more general class of nonlinear differential equations that are typically considered in the study of the convergence of Runge–Kutta or multistep methods (see e.g. [5], [3], [9]).

We will assume that the derivative function f satisfies a one-sided Lipschitz condition (the norm considered will be the euclidean norm associated to an inner product, $|v|^2 = \langle v, v\rangle$),

$$\langle y - z, f(t, y) - f(t, z)\rangle \leq \nu|y - z|^2, \quad \forall t \in [0, T], \quad \forall y, z \in I\!R^m. \tag{3.16}$$

where $\nu = \mathcal{O}(1)$, and we will pay special attention to the case $\nu = 0$, i.e., when the problem is dissipative. Here and throughout the paper $\mathcal{O}(1)$ will mean a quantity which is independent of the stiffness of the problem.

We will also consider that the derivatives of the exact solution $y(t)$ satisfy

$$|y^{(l)}(t)| \leq M_l = \mathcal{O}(1), \quad \forall t \in [0, T], \quad l = 1, \ldots, q + 1 \tag{3.17}$$

where q denotes the stage order of the RK considered.

The next theorems confirm that the main order results obtained in theorems 3.4 and 3.5 for the Prothero and Robinson equation remain valid also for the nonlinear family of problems satisfying (3.16)-(3.17). There, the RK method was supposed to be AS– and ASI–stable, stability properties that appear in a natural way in the study of the convergence of the methods with linear stiff systems. To prove the order results with the nonlinear stiff equations, we will suppose that the RK method is diagonally stable, i.e., there exist a positive definite diagonal matrix D such that $DA+A^TD$ is definite positive. Let us note that this condition implies that the matrix A is nonsingular and therefore the method is AS-, ASI–stable and condition a) of theorem 3.2 is always satisfied.

Theorem 3.7. *Let a non-confluent RK (A, b) of s stages satisfying*
(i) B(s-1) and C(s-1) and
(ii) it is diagonally stable.

Then, by considering the local assumption $y_0 = y(t_0)$, the s–parameter family of starting algorithms of type I of order $s-1$ in part (a) of theorem 2.4 has order stiff $s-1$. Moreover, if $B(s)$ and $C(s)$ are fulfilled, the unique starting algorithm (of type I) of order s given by the polynomial extrapolation of Lagrange, as indicated in statement (c) of theorem 2.4, has also order stiff s.

Proof. Since this study cannot be carried out through the Butcher's series theory we consider a different approach that makes use of the assumptions (3.17) and (3.16). Thus, let us denote

$$\hat{X}_i = y(t_0 + c_i h), \quad \hat{Y}_i = y(t_0 + (1 + rc_i)h), \quad 1 \le i \le s, \tag{3.18}$$

being $y(t)$ the exact solution of the initial value problem, and let us define the vectors

$$Z_i = \gamma_i y_0 + \sum_{j=1}^{s} \alpha_{ij} \hat{X}_j, \quad 1 \le i \le s. \tag{3.19}$$

With these notations we can put,

$$Y_i - Y_i^0 = \underbrace{Y_i - \hat{Y}_i}_{(1)} + \underbrace{\hat{Y}_i - Z_i}_{(2)} + \underbrace{Z_i - Y_i^0}_{(3)}. \tag{3.20}$$

Now, we analyze separately every addendum.

(1) From $B(s-1)$, $C(s-1)$ and the diagonal stability of the RK method, proceeding for instance as in theorem 14.3, page 219 of [9], we can ensure that $Y_i - \hat{Y}_i = \mathcal{O}(h^s)$.

(3) $Z_i - Y_i^0 = \sum_{j=1}^{s} \alpha_{ij}(\hat{X}_j - X_j) = \mathcal{O}(h^s)$, since α_{ij} are uniformly bounded for $r \in (0, r_0]$, with $r_0 = \mathcal{O}(1)$ and $(\hat{X}_j - X_j) = \mathcal{O}(h^s)$ (see theorem 14.3 in [9]).

(2) By expanding in powers of h around t_0 we have,

$$\begin{aligned}\hat{Y}_i - Z_i &= y(t_0 + (1+rc_i)h) - \gamma_i y_0 - \sum_{j=1}^{s} \alpha_{ij} y(t_0 + c_j h) \\ &= \gamma_i (y(t_0) - y_0) + (1 - \gamma_i - \sum_{j=1}^{s} \alpha_{ij}) y(t_0) \\ &\quad + \sum_{k\geq 1} \frac{y^{(k)}(t_0)}{k!} \left((1+rc_i)^k - \sum_{j=1}^{s} \alpha_{ij} c_j^k \right) h^k.\end{aligned}$$

Hence the order conditions (2.6) together the local assumption $y(t_0) = y_0$ give us $\hat{Y}_i - Z_i = \mathcal{O}(h^s)$. Moreover, assuming $B(s)$, $C(s)$, if the starting algorithm has order s, the three addenda in (3.20) are $\mathcal{O}(h^{s+1})$ and we have order stiff s. On the other hand, the stiff order $s+1$ cannot be reached because for the particular case of the Prothero and Robinson model such fact is impossible as we saw in theorem 3.4. □

Theorem 3.8. *If we consider an s–stage non-confluent Runge-Kutta (A, b) and assume (i)-(ii) as in theorem 3.7, then by taking $y_0 = y(t_0)$ we have for the starting algorithms of type II that,*
(a) The $2s$-parameter family considered in theorem 3.5-(a) possesses order stiff $s-1$.
(b) If $B(s)$ and $C(s)$ are also fulfilled, there exists a s-parameter family with stiff order s, which is given by the equations (2.21) (take for instance, δ_i $(i = 1, \ldots, s)$ as parameters). If we choose $\delta_i = 0\,(i = 1, \ldots, s)$, then the coefficients β_{ij} are given by the polynomial extrapolation of Lagrange as indicated in theorem 3.5-(b).

Proof. By using the notations given into (3.18) and replacing Z_i in (3.19) by

$$Z_i = y_0 + h\delta_i f(t_0, y_0) + h \sum_{j=1}^{s} \beta_{ij} f(t_0 + c_j h, \hat{X}_j). \tag{3.21}$$

We can put

$$Y_i - Y_i^0 = \underbrace{Y_i - \hat{Y}_i}_{(1)} + \underbrace{\hat{Y}_i - Z_i}_{(2)} + \underbrace{Z_i - Y_i^0}_{(3)}.$$

By using the same arguments as in the proof of theorem 3.7 we have

(1) $Y_i - \hat{Y}_i = \mathcal{O}(h^s)$.

Now, by using the uniform boundedness of β_{ij} and the theorem 14.3, page 219 in [9] (recall that A is nonsingular by the assumption *(ii)*) it is easy to see that,

(3) $Z_i - Y_i^0 = h \sum_{j=1}^{s} \beta_{ij} (f(t_0 + c_j h, \hat{X}_j) - f(t_0 + c_j h, X_j)) = \mathcal{O}(h^s)$.

(2) By considering the Taylor expansions of $y'(t_0 + c_j h)$ and $y(t_0 + (1+rc_i)h)$ around t_0, we

arrive at,

$$\begin{aligned}\hat{Y}_i - Z_i = \quad & y(t_0) - y_0 + h\delta_i(y'(t_0) - f(t_0, y_0)) + \\ & (1 + rc_i - \delta_i - \sum_{j=1}^{s} \beta_{ij})hy'(t_0) \\ & + \sum_{k\geq 2} \frac{y^{(k)}(t_0)}{k!} \left((1 + rc_i)^k - k \sum_{j=1}^{s} \beta_{ij} c_j^{k-1} \right) h^k.\end{aligned}$$

Again with the local assumption $y(t_0) = y_0$, if the starting algorithm has order $s - 1$, then $\hat{Y}_i - Z_i = \mathcal{O}(h^s)$ and it also has order stiff $s - 1$.

(b) The proof is similar to the proof of part (a). □

REMARK 3.2. In previous results we have supposed that $y_0 = y(t_0)$. If we consider $y_0 - y(t_0)$ as the global error of the preceding step and we suppose that the RK method is B–convergent of order p, then $y_0 - y(t_0) = \mathcal{O}(h^p)$ and consequently for the starting algorithms of type I

$$Y_i - Y_i^0 = \mathcal{O}(h^q), \quad 1 \leq i \leq s,$$

being $q = \min\{p, s-1\}$ or $q = \min\{p, s\}$ depending respectively on if $B(s-1), C(s-1)$ or $B(s), C(s)$ are satisfied. This fact cannot be guaranteed at all for the starting algorithms of type II when some $\delta_i \neq 0$, which says in some sense that the algorithms of type II have in general worse stability properties than those of type I (this fact was already reflected in the analysis carried out on the Prothero and Robinson model). However, if the method is stiffly accurate, then $h\delta_i(y'(t_0) - f(t_0, y_0)) = \mathcal{O}(h^p)$ and the starting algorithm has again order stiff q.

3.3 Some relevant starting algorithms.

In this section we summarize the order results of some relevant starting algorithms of *types I-II* for the following Runge-Kutta methods: Gauss, Radau IA, Radau IIA, Lobatto IIIA and Lobatto IIIC. We give a brief guide for practical purposes of use, by indicating the equations that the starting algorithms must fulfill, as well as a special notation for them.

Type I.

$\mathcal{L}_s^0$: denotes de polynomial extrapolation of Lagrange based on the internal stages $X_1, \ldots, X_s$ and on y_0, as indicated in theorem 2.4 by (2.8). As proved in previous sections, this algorithm has order (stiff and nonstiff) s for Gauss and RadauIIA, that satisfy conditions $C(s)$, and order $s-1$ for Radau IA and Lobatto IIIC, that satisfy $C(s-1)$. For Lobatto IIIA, even though it satisfies $C(s)$, its matrix A is singular and the algorithm has order $s-1$. Concerning the amplifying functions, they are uniformly bounded in all the cases, and the values at $z = \infty$ are $\gamma_i \neq 0$.

$\mathcal{L}_s^1$: denotes de polynomial extrapolation of Lagrange of the internal stages $X_1, \ldots, X_s$ ($\gamma_i = 0$), as indicated in theorem 2.4 by (2.7). It has order $s-1$ in all the cases and the amplifying functions (uniformly bounded) satisfy $\bar{R}_i(\infty) = 0$, except for Lobatto IIIA. Let us not that for this particular method, $y_0 = X_1$ and this algorithm is equivalent to $\mathcal{L}_s^0$.

$\mathcal{M}_s^I$: denotes the starting algorithm of type I of (classical) order s when A is nonsingular and the stage order of the RK method is $s-1$ (see theorem 2.4). It is given by (2.6) and (2.16).

Table 3.1: Gauss and Radau IIA (A nonsingular, $C(s)$)

	nonstiff order	stiff order	P-R order at $z = 0$	P-R order at $z = \infty$	$\tilde{R}_i(\infty)$
$\mathcal{L}_s^1$	$s-1$	$s-1$	$s-1$	$s-1$	0
$\mathcal{L}_s^0$	s	s	s	s	bounded
$\mathcal{M}_{s+1}^{II,1}$	$s+1$	s	$s+1$	s	∞
$\mathcal{M}_{s+1}^{II,2}$	s	s	s	$s+1$	∞

Table 3.2: Radau IA and Lobatto IIIC (A nonsingular, $C(s-1)$)

	nonstiff order	stiff order	P-R order at $z = 0$	P-R order at $z = \infty$	$\tilde{R}_i(\infty)$
$\mathcal{L}_s^1$	$s-1$	$s-1$	$s-1$	$s-1$	0
$\mathcal{L}_s^0$	$s-1$	$s-1$	$s-1$	$s-1$	bounded
$\mathcal{M}_s^I$	s	$s-1$	$s-1$	$s-1$	∞

It makes sense for Radau IA and Lobatto IIIC (for the other methods, it coincides with $\mathcal{L}_s^0$). It has order stiff $s-1$ and the amplifying functions are bounded.

Type II

$\mathcal{M}_{s+1}^{II,1}$: for Gauss and Radau IIA, denotes the starting algorithm of type II and classical order $s+1$ given by (2.23). The order stiff is s and the amplifying function is not bounded when z goes to infinity (see theorem 3.6).

$\mathcal{M}_{s+1}^{II,2}$: for Gauss and Radau IIA, denotes the starting algorithm of classical order s and order $s+1$ at the particular point $z = \infty$ (on the Prothero and Robinson model). Its governing equations are given by (3.13) (see theorem 3.5-(c)). As in the previous case, the order stiff is s and the amplifying functions are not bounded when z goes to infinity.

$\mathcal{M}_{s+1}^{II,3}$: for Lobatto IIIA, denotes the starting algorithm of classical and stiff order s given by theorem 3.5-(b). For this algorithm the amplifying functions are not bounded when z goes to infinity (see theorem 3.6).

Remark 3.3. For Radau IA and lobatto IIIC methods the stage order is $s-1$, hence the maximal order (classical) is s and we have a s-parameter family of this order. Moreover, this family only achieves order $s-1$ on the Prothero and Robinson model and the stiff order is $s-1$ (in the case of Lobatto IIIC a special analysis must be carried out because these methods are not diagonally stable for $s > 2$, and their stiff order would be only $s-2$ for problems satisfying (3.16) with $\nu < 0$).

Table 3.3: Lobatto IIIA (A singular, $C(s)$)

	Nonstiff order	stiff order	P-R order at $z = 0$	P-R order at $z = \infty$	$\tilde{R}_i(\infty)$
$\mathcal{L}_s^1 \equiv \mathcal{L}_s^0$	$s-1$	$s-1$	$s-1$	$s-1$	bounded
$\mathcal{M}_{s+1}^{II,3}$	s	s	s	s	∞

In tables 3.1, 3.2 and 3.3 we summarize the properties of the starting algorithms considered.

4 Numerical experiments.

In this section we present some numerical experiments showing the performance of the starting algorithms studied in previous sections.

As Runge–Kutta method we have chosen the fifth–order, three–stage RadauIIA formula that has been implemented in the code RADAU5 [9]. This method satisfies conditions $C(3)$ and is stiffly accurate so that we can compare the efficiency of the starting algorithms $\mathcal{L}_3^1$, $\mathcal{L}_3^0$, $\mathcal{M}_4^{II,1}$and $\mathcal{M}_4^{II,2}$ whose properties are given in table 3.1 for $s = 3$. For comparison purposes we have also included the starting values $Y_i^0 = y_1$, $i = 1, 2, 3$. They are in fact type I algorithms with $\gamma_i = 0$ and $\alpha_i^T = (0, 0, 1)$, their order (stiff and classical) is 0 and their amplifying function $\tilde{R}_i(z) = R(z)$ equal the amplifying function of the RadauIIA method.

In order to do our experiments, we have modified the code RADAU5 so that it can use these starting algorithms and we have integrated a number of stiff problems, including those of the well known DETEST package [4].

To measure the efficiency of the algorithms, we have computed for each problem and error tolerance the global error (GE) of the numerical solution at the end of the integration interval and the number of evaluations of the derivative function (NFN) required in the integration. This can not be an adequate measure of the final computational cost in implicit methods, mainly if the dimension of the differential system is large. However, in most cases a larger value of NFN corresponds with a larger value of the number of LU matrix factorizations and evaluations of the Jacobian matrix, and we have considered this value NFN useful because it gives us a good idea of the cost involved in the iterations of the modified Newton scheme.

In figure 4.1 we have plotted the pairs (NFN, Log(GE)) obtained integrating with absolute and relative error tolerances 10^{-2}, 10^{-3}, ..., 10^{-9} the Van der Pol problem:

Problem 1.- The Van der Pol oscillator (see e. g. [9] pp. 144)

$$\begin{array}{ll} y_1' = y_2 & y_1(0) = 2 \\ y_2' = ((1 - y_1^2)y_2 - y_1)/\varepsilon & y_2(0) = 0, \\ t \in [0, 2], \qquad \varepsilon = 10^{-6}. & \end{array}$$

In figure 4.2 we have plotted the corresponding pairs of values obtained integrating the Oregonator problem:

Problem 2.- The Oregonator (see e. g. [9] pp. 144).

$$\begin{array}{ll} y_1' = 77.27(y_2 + y_1(1 - 8.375 \times 10^{-6} y_1 - y_2)) & y_1(0) = 1 \\ y_2' = (y_3 - (1 + y_1)y_2)/77.27 & y_2(0) = 2 \\ y_3' = 0.161(y_1 - y_3) & y_3(0) = 3, \\ t \in [0, 360]. & \end{array}$$

In the figures we can observe how for low tolerances starting algorithms with higher order perform better than the others and in particular $\mathcal{M}_4^{II,1}$, that has the highest classical order, is the most efficient. For large tolerances, there are no clear differences between the algorithms. Such kind of results have been obtained in most of the problems we have integrated, thus they represent in some way the general behavior of these starting algorithms for stiff problems.

RTOL	INI	GE	NSTEP	NRC	NLU	NSOL	NFN	NITER
10^{-2}	y_1	0.7644E-03	128	49	169	523	1698	2.74
	$\mathcal{L}_3^1$	0.8244E-03	119	36	149	392	1296	2.58
	$\mathcal{L}_3^0$	0.5429E-04	122	28	144	367	1224	2.43
	$\mathcal{M}_4^{II,2}$	0.3064E-03	121	22	138	417	1372	2.84
	$\mathcal{M}_4^{II,1}$	0.6045E-03	122	27	145	377	1254	2.52
10^{-4}	y_1	0.8551E-05	302	44	316	1036	3410	2.76
	$\mathcal{L}_3^1$	0.3426E-05	285	16	272	756	2553	2.43
	$\mathcal{L}_3^0$	0.9128E-05	283	8	258	674	2305	2.25
	$\mathcal{M}_4^{II,2}$	0.8832E-05	287	3	261	745	2522	2.39
	$\mathcal{M}_4^{II,1}$	0.5668E-05	285	6	234	607	2106	2.02
10^{-6}	y_1	0.2672E-06	900	0	521	2486	8359	2.74
	$\mathcal{L}_3^1$	0.5884E-07	856	1	695	1881	6500	2.18
	$\mathcal{L}_3^0$	0.2050E-07	848	0	595	1589	5616	1.85
	$\mathcal{M}_4^{II,2}$	0.3933E-07	859	0	694	1879	6497	2.17
	$\mathcal{M}_4^{II,1}$	0.7227E-07	845	0	556	1286	4704	1.50
10^{-8}	y_1	0.1467E-08	2901	0	2035	8894	29584	3.06
	$\mathcal{L}_3^1$	0.3013E-09	2719	1	2059	5587	19481	2.05
	$\mathcal{L}_3^0$	0.8068E-09	2699	0	1327	4286	15558	1.59
	$\mathcal{M}_4^{II,2}$	0.8254E-10	2719	0	1956	5283	18569	1.94
	$\mathcal{M}_4^{II,1}$	0.4708E-09	2657	0	1049	3378	12792	1.27

Table 4.1: Van der Pol problem.

In table 4.1 we present for the Van der Pol problem a more detailed set of values obtained in the integrations. In particular, for each starting algorithm and error tolerance, we give the number of successful steps (NSTEP), the number of rejected steps due to a fail in the convergence (NRC), the number of LU factorizations (NLU), the number of triangular systems solved (NSOL), the average number of iterations required to reach convergence at each step (NITER), the number of evaluations of the derivative function NFN and the global error at the end point of the integration interval GE.

It can be seen in the table that in general, all the algorithms give a similar number of steps and LU factorizations, except perhaps the algorithm $Y_i^0 = y_1$ that has just order 0. Regarding the number of iterations required to get convergence of the iterative scheme, it decreases in general as the order of the algorithm increases and therefore the same happens with the number of evaluations of the derivative function and the number of triangular systems solved. In particular, $\mathcal{M}_4^{II,1}$ is in this sense the most efficient.

With some problems we have observed that iterations started with algorithms with unbounded amplifying functions ($\mathcal{M}_4^{II,1}$ and $\mathcal{M}_4^{II,2}$) present more difficulties to converge than the others, mainly for large tolerances. In these cases, the algorithms for which the amplifying function vanishes when z goes to infinity turned out to be more robust than the others. An example of such a kind of problem is the E5 of the DETEST package:

Problem 3.- E5 of DETEST package (see e. g. [9] pp. 145).

$$\begin{aligned}
&y_1' = -7.89 \cdot 10^{-10} y_1 - 1.1 \cdot 10^7 y_1 y_3 && y_1(0) = 1.76 \cdot 10^{-3} \\
&y_2' = 7.89 \cdot 10^{-10} y_1 - 1.13 \cdot 10^9 y_2 y_3 && y_2(0) = 0 \\
&y_3' = 7.89 \cdot 10^{-10} y_1 - 1.1 \cdot 10^7 y_1 y_3 && y_3(0) = 0 \\
&\qquad -1.13 \cdot 10^9 y_2 y_3 + 1.13 \cdot 10^3 y_4 && \\
&y_4' = 1.1 \cdot 10^7 y_1 y_3 - 1.13 \cdot 10^3 y_4 && y_4(0) = 0, \\
&t \in [0, 1000]. &&
\end{aligned}$$

In figure 4.3 we have plotted the pairs (NFN, Log(GE)) obtained integrating the E5 problem. In this case, due to the particular values of the solution of this problem, we have used different relative and absolute error tolerances. More precisely, we have taken relative error tolerances RTOL from 10^{-2} to 10^{-9} and for each of them we have taken as absolute error tolerance ATOL = RTOL $\times 10^{-3}$.

As we can see, the algorithms $\mathcal{M}_4^{II,1}$ and $\mathcal{M}_4^{II,2}$ present an anomalous behavior. Besides that, with these algorithms the solution obtained with RTOL = 10^{-2} and 10^{-3} was not valid and we have not include the corresponding values in the plot. The algorithms $\mathcal{L}_3^0$ and $\mathcal{L}_3^1$ present a more regular behavior and $\mathcal{L}_3^1$ performs even better than the other, due probably to the fact that its amplifying function vanishes at infinity.

In Table 4.2 we present for the E5 problem, the same kind of data as in Table 4.1 for the Van der Pol problem.

It can be seen in the table that in general, for low error tolerances all the algorithms give a similar number of steps and LU factorizations, except again the algorithm $Y_i^0 = y_1$ that has only order 0. However, the algorithms with unbounded amplifying functions provide a numerical solution with a higher error which make them less efficient. For large tolerances, the algorithms $\mathcal{M}_4^{II,1}$ and $\mathcal{M}_4^{II,2}$ make the method integrate the problem with more steps than with the other starting algorithms and in particular for tolerances 10^{-2} and 10^{-3} the code was not able to complete the integration.

We must mention that integrating this problem with the same relative error tolerances and taking as absolute error tolerances ATOL = RTOL $\times 10^{-10}$, that is, an almost pure relative error control, the behavior of the algorithms was similar to that of Van der Pol or Oregonator problems.

In conclusion, from our experiments we can deduce that increasing the order of the starting algorithm can lead to more efficient integrations, but a special attention must be paid to the amplifying error functions if we want to have also a robust integrator. Some more effort must be done in the search of new starting algorithms according to these requirements and such a research is the subject of current work.

References

[1] K. Burrage, W. H. Hundsdorfer and J. G. Verwer, *A study of B-convergence of Runge-Kutta methods*, Computing 36 (1986), pp. 17–34.

[2] M. Calvo, J. I. Montijano and S. Gonzalez-Pinto, *Runge-Kutta methods for the numerical solution of stiff semilinear problems*, submitted to BIT (1998).

[3] K. Dekker and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North Holland, Amsterdam, 1984.

[4] W. H. Enright, T. E. Hull and B. Lindberg, *Comparing numerical methods for stiff systems of ODEs*, BIT, 15 (1975), pp. 10–48.

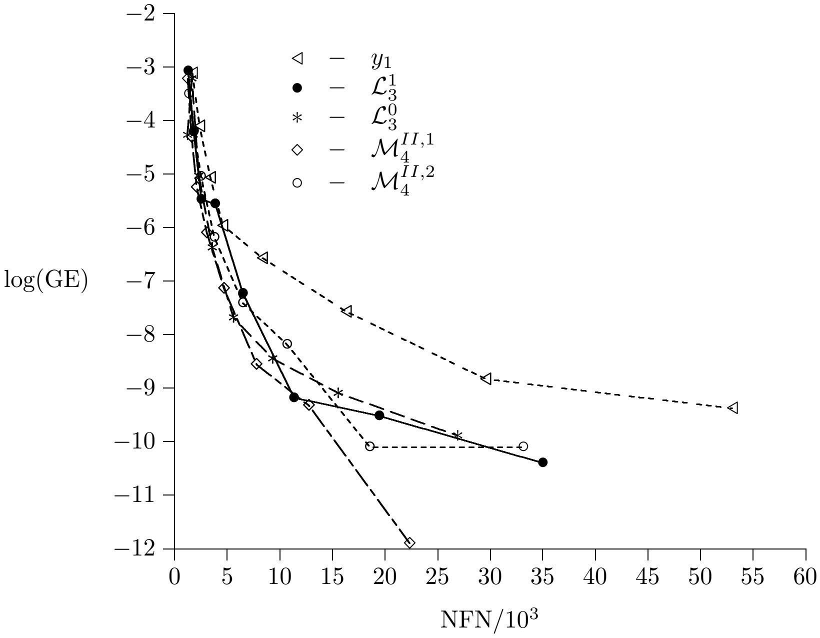

Fig. 4.1 Van der Pol problem

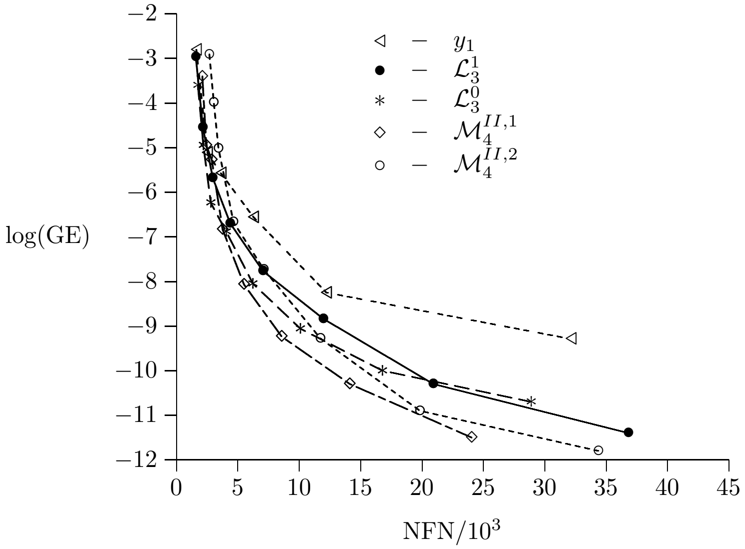

Fig. 4.2 Oregonator problem

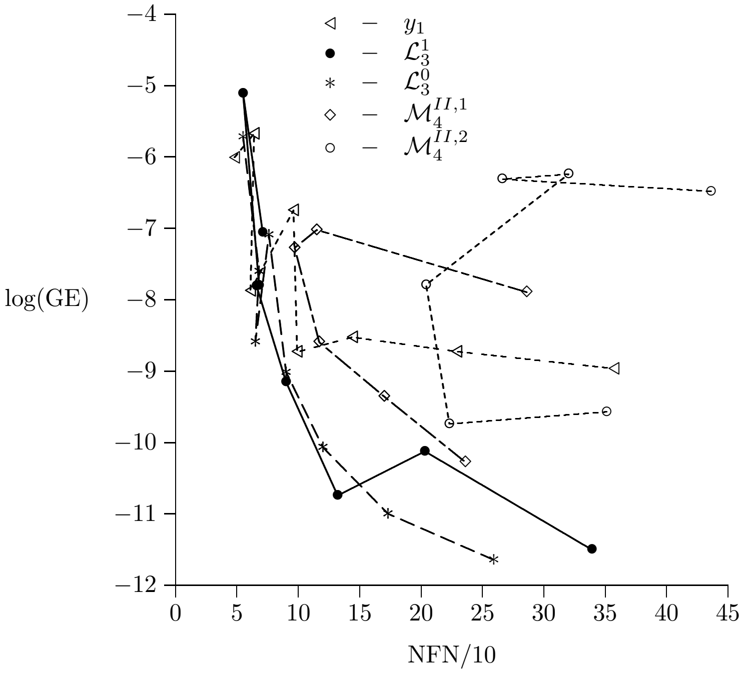

Fig. 4.3 E5 problem

RTOL	INI	GE	NSTEP	NRC	NLU	NSOL	NFN	NITER
10^{-2}	y_1	0.9853E-06	9	1	10	13	48	1.22
	$\mathcal{L}_3^1$	0.8701E-07	11	3	13	20	71	1.18
	$\mathcal{L}_3^0$	0.1922E-05	10	2	11	15	55	1.10
	$\mathcal{M}_4^{II,2}$	***	**	**	**	**	**	****
	$\mathcal{M}_4^{II,1}$	***	**	**	**	**	**	****
10^{-3}	y_1	0.2132E-05	10	1	11	18	64	1.40
	$\mathcal{L}_3^1$	0.7734E-05	10	1	11	15	55	1.20
	$\mathcal{L}_3^0$	0.2519E-07	11	2	13	19	68	1.36
	$\mathcal{M}_4^{II,2}$	***	**	*	**	**	**	***
	$\mathcal{M}_4^{II,1}$	***	**	*	**	**	**	***
10^{-4}	y_1	0.1341E-07	10	0	10	17	61	1.70
	$\mathcal{L}_3^1$	0.1585E-07	11	1	12	19	68	1.36
	$\mathcal{L}_3^0$	0.2593E-08	11	1	12	18	65	1.45
	$\mathcal{M}_4^{II,2}$	0.3314E-06	31	45	74	135	436	1.42
	$\mathcal{M}_4^{II,1}$	0.1277E-07	22	29	51	88	286	1.36
10^{-6}	y_1	0.1855E-08	15	0	15	28	99	1.73
	$\mathcal{L}_3^1$	0.7028E-09	15	0	16	25	90	1.53
	$\mathcal{L}_3^0$	0.9694E-09	15	0	16	25	90	1.47
	$\mathcal{M}_4^{II,2}$	0.5708E-06	32	22	55	96	320	1.59
	$\mathcal{M}_4^{II,1}$	0.5335E-07	16	1	17	27	97	1.56
10^{-8}	y_1	0.1881E-08	31	1	20	66	229	1.87
	$\mathcal{L}_3^1$	0.7518E-10	32	0	24	57	203	1.72
	$\mathcal{L}_3^0$	0.1003E-10	32	0	23	47	173	1.41
	$\mathcal{M}_4^{II,2}$	0.1799E-09	34	0	25	63	223	1.74
	$\mathcal{M}_4^{II,1}$	0.4389E-09	32	0	24	46	170	1.31

Table 4.2: E5 problem.

[5] R. Frank, J. Schneid and C. W. Ueberhuber, *The concept of B-convergence*, SIAM J. Numer.Anal., 18 (1981), pp. 753–780.

[6] S. Gonzalez-Pinto, J. I. Montijano and L. Randez, *Iterative schemes for implicit Runge-Kutta methods*, Appl. Numer. Math., 17 (1995), pp. 363-382.

[7] E. Hairer, Ch. Lubich and M. Roche, *The numerical solution of differential algebraic systems by Runge-Kutta methods*, Lecture Notes in Math. 1409, Springer Verlag (1989).

[8] E. Hairer, S. P. Nørsett and G. Wanner , *Solving ordinary differential equations I*, Springer Verlag, Berlin, 1993.

[9] E. Hairer and G. Wanner, *Solving ordinary differential equations II*, Springer Verlag, Berlin, 1996.

[10] P. J. van der Houwen and B. P. Sommeijer, *Preconditioning in parallel Runge-Kutta methods for stiff initial value problems*, J. Comput. Math. Appl. 28, No. 10-12, 17-31 (1994), pp. 17-31.

[11] M. P. Laburta, *Starting algorithms for IRK methods*, J. Comput. Appl. Math., 83, (1997), pp. 269-288.

[12] M. P. Laburta, *Construction of starting algorithms for the RK-Gauss methods*, J. Comput. Appl. Math., 90, (1998), pp. 239-261.

[13] C. Mac Donald and W. Enright, *Implications of order reduction for implicit Runge–Kutta methods*, Numerical Algorithms, 2 (1992), pp. 351-370.

[14] S. P. Nørsett and P. G. Thomsen, *Local error control in SDIRK-methods*, BIT, 26 (1986), pp. 100-113.

[15] H. Olsson and G. Söderlind, *Stage value predictors and efficient Newton iterations in implicit Runge–Kutta methods*, SIAM Journal on Scientific Computing, 20, 1 (1999), pp. 185–202.

[16] A. Prothero and A. Robinson, *On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations*, Math. Comp., 28, (1974), pp. 145-162.

[17] T. Roldan and I. Higueras, *IRK methods for DAE: starting algorithms*, Dept. of Mathematics and Computer Science, Univ. Public Navarra, Spain, 1998.

[18] J. Sand, *Starting methods for the iteration of IRK's*, Dept. of Computer Science, Univ. Copenhagen, Denmark, 1989.