Construction of starting algorithms for the RK–Gauss methods

M.P. Laburta*
Departamento de Matemática Aplicada, Universidad de Zaragoza, Centro Politécnico Superior, María de Luna 3, 50015-Zaragoza, Spain

Received 30 June 1997; received in revised form 23 October 1997

Abstract
In this paper starting algorithms for the numerical solution of stage equations in Runge–Kutta–Gauss formulae with 2, 3 and 4 stages are constructed. For each of these formulae, three types of starting algorithms are given according to their requirement of none, one or two additional function evaluations per step. Numerical experiments with Hamiltonian systems are presented to show the superior performance of the new starting algorithms of high order. © 1998 Elsevier Science B.V. All rights reserved.

AMS classification: 65L06; 65L05

Keywords: Starting algorithms; Gauss methods; Implicit methods; Runge–Kutta; Symplectic methods; Hamiltonian systems

1. Introduction
Consider initial value problems in ordinary differential equations

\[ y'(t) = F(y(t)), \quad y(t_0) = y_0 \in \mathbb{R}^D, \tag{1} \]

where \( F : \mathbb{R}^D \to \mathbb{R}^D \) is a given sufficiently smooth function. In order to simplify the notation, only autonomous problems will be considered throughout this paper, although the whole discussion can be extended to nonautonomous systems.

An \( s \)-stage Runge–Kutta (RK) method applied to the system in (1) advances the numerical solution from \( t_n \) to \( t_{n+1} = t_n + h \) through the equation

\[ y_{n+1} = y_n + h \sum_{i=1}^{s} b_i F(Y_i), \]

* E-mail: laburta@posta.unizar.es.

1 Partially supported by the Spanish DGICYT (Pb 93-0305).
where the internal stages $Y_i$ are determined by the relations

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij} F(Y_j), \quad i = 1, \ldots, s. \quad (2)$$

As usual, we shall assume that $c_i = \sum_{j=1}^{s} a_{ij}$ $\forall i$, and we will denote by $A$ the $s \times s$ matrix $(a_{ij})$ and by $b$ and $c$ the column vectors $(b_1, \ldots, b_s)^T$ and $(c_1, \ldots, c_s)^T$, respectively.

Physical systems from mechanics, optics, etc., are often described through a set of Hamilton equations with $d$ degrees of freedom

$$\frac{dp_i}{dt} = \frac{\partial H}{\partial q_i}, \quad \frac{dq_i}{dt} = -\frac{\partial H}{\partial p_i}, \quad i = 1, \ldots, d, \quad (3)$$

where $H = H(p_1, \ldots, p_d, q_1, \ldots, q_d)$ is the Hamiltonian function. Clearly, (3) is a particular case of the system in (1) with $y = (p_1, \ldots, p_d, q_1, \ldots, q_d)^T$, $D = 2d$.

When Hamiltonian systems of differential equations are numerically integrated, it is often advantageous to use symplectic methods [13]. Recently, different methods of type RK which are explicit and symplectic for certain classes of Hamiltonian systems have been constructed. Thus, there are explicit partitioned Runge–Kutta methods (PRK) (see, e.g., [16, 12, 8]), and explicit Runge–Kutta–Nyström methods (RKN) (see, e.g., [1, 8, 2, 9]), which are symplectic when they are applied, respectively, to Hamiltonian problems with separable Hamiltonian $H(p, q) = T(p) + V(q)$, or of the form

$$H(p, q) = \frac{1}{2} p^T M^{-1} p + V(q),$$

where $M$ is often a diagonal matrix (mass matrix). However, they cannot be applied to general Hamiltonians $H(p, q)$.

Explicit Runge–Kutta methods are not symplectic. In [11] it was shown that an $s$-stage RK method is symplectic for general Hamiltonians if its coefficients satisfy

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \quad 1 \leq i, j \leq s.$$ 

In particular, the RK schemes based on Gauss–Legendre quadrature, due to Butcher, are symplectic. Furthermore, these formulae possess maximal and arbitrarily high order $(2s)$ and they also have good stability properties. All these advantages have carried us to use them for the numerical integration of Hamiltonian systems.

However, the Gauss methods have a difficulty: They are implicit and therefore the cost of the required linear algebra for the resolution of (2) may be high. Usually, this nonlinear system is solved by means of some iterative scheme and its efficiency depends on the number of iterations needed in each step. In general, the more accurate the initial values $Y_i^0$ which approximate to the stage vectors $Y_i$, $i = 1, \ldots, s$, are, the lower this number of iterations will be. In order to reduce that computational cost and by applying the theory developed in [7], several starting algorithms are constructed for the 2-, 3- and 4-stage RK–Gauss methods in this paper.

Each initial value $Y_i^0$ to start the iterations to solve (2) will be said to have order $p$ if this is the largest integer for which

$$\| Y_i - Y_i^0 \| = O(h^{p+1}). \quad (4)$$

Furthermore, a starting algorithm that produces the initial values $Y_i^0$, $i = 1, \ldots, s$, will be said to have order $p$ if this is the largest integer such that (4) is satisfied for all $i = 1, \ldots, s$. 

There are very few papers in the literature concerning starting algorithms [10, 7]. In [7], three types of starting algorithms were studied according to their additional computational cost per step. There, it was proved that, under some conditions on the coefficients of the RK method, starting algorithms with one additional function evaluation per step can be obtained whose order exceeds by one those without additional cost. Moreover, by adding another additional evaluation per step, the order again increases by one. For each of these three cases, a method for the construction of a starting algorithm of maximal order was given.

The coefficients of the s-stage RK–Gauss formula satisfy the conditions that were imposed in the above-mentioned results and therefore we can construct starting algorithms for them of order s, s + 1 or s + 2, according to these not requiring additional cost or needing one or two additional function evaluations per step, respectively.

Taking into account the large class of problems to which the Gauss formulae can be efficiently applied, the possibility of making some changes in the step size to obtain the initial values has been considered. However, by simplicity, for the high-order algorithms we will only give the expressions corresponding to constant step sizes.

In order to evaluate the final computational cost, we will not only take into account the order of the starting algorithm and its additional cost per step, but also the iterative scheme that we are going to use. Since we are going to apply the starting algorithms to nonstiff Hamiltonian problems, if the iterative scheme is functional iteration (FI), we have for the kth iteration [5]

\[ ||Y^k - Y|| = O(h) \ ||Y^{k-1} - Y||, \quad k = 1, 2, \ldots \]

(5)

and if the iterations are computed by means of quasi-Newton (QN) method, we have

\[ ||Y^k - Y|| = O(h^2) \ ||Y^{k-1} - Y||, \quad k = 1, 2, \ldots . \]

(6)

So, if we use a predictor with one more order than another one, we can say (roughly speaking) that the iterative scheme FI will require one less iteration, and therefore, s evaluations of the derivative function will be saved per step. This same saving will appear if we use QN and we compare two predictors whose order differ by 2. Therefore, if we compare two starting algorithms whose additional cost and order differ by one, the theoretical saving per step on the number of evaluations of the derivative function is

\[
\text{FI: } s - 1 \quad (s > 1) \quad \text{QN: } \frac{1}{2}s - 1 \quad (s > 2).
\]

By these rough considerations, we expect, in general, a superior performance of the starting methods of high order, and that they give better results when the stage number s grows larger. Moreover, their advantages will be more remarkable when we use functional iteration.

When nonstiff problems are considered, important work about the order of RK methods which use simple iteration, modified Newton iteration or full Newton iteration to compute their internal stages values has been done in [5, 6]. Further results on the behaviour of modified Newton schemes in the solution of stage equations of implicit RK methods for stiff systems have been obtained by Spijker [14].

The paper is organized as follows: In Section 2 the three types of starting algorithms that will be constructed are described, and asymptotic expressions for their errors are given. In Sections 3–5, we expose, respectively, the construction of different starting algorithms for the 2-, 3- and 4-stage
RK–Gauss methods. Finally, in Section 6 we present some numerical experiments comparing some of the starting methods constructed for the 3- and 4-stage RK–Gauss formulae applied to several Hamiltonian problems.

2. Description of the starting methods

In the sequel we will assume that the first step of the RK method from $t_0$ to $t_1 = t_0 + h$ with internal stages $Y_i$, $i = 1, \ldots, s$, has been already completed. We look for initial approximations $\hat{Y}_i^0$ to the internal vectors $\hat{Y}_i$ of the following step from $t_1$ to $t_1 + rh$. For the other steps we may proceed in an analogous way.

The starting methods without additional cost that we are going to consider here have the form

$$\hat{Y}_i^0 = y_0 + h \sum_{j=1}^{s} b_{i,j} F(Y_j), \quad i = 1, \ldots, s, \quad (7)$$

where the values of the coefficients $b_{i,j}$ for which the order is the highest possible will be obtained.

Starting algorithms with at most one function evaluation per step will have the form

$$\hat{Y}_i^0 = y_0 + h \left[ b_{i,0} F(y_0) + \sum_{j=1}^{s} b_{i,j} F(Y_j) \right], \quad i = 1, \ldots, s, \quad (8)$$

and if, moreover, another function evaluation per step is added, we will compute

$$\hat{Y}_i^0 = y_0 + h \left[ b_{i,0} F(y_0) + \sum_{j=1}^{s} b_{i,j} F(Y_j) + b_{i,s+1} F(y_{i+1}) + b_{i,s+2} F(Y_{s+2}) \right], \quad i = 1, \ldots, s, \quad (9)$$

where the new function evaluation corresponds to the arbitrary stage $Y_{s+2} = y_0 + h \left[ \mu_0 F(y_0) + \sum_{j=1}^{s} \mu_j F(Y_j) \right]$.

Let us notice that (8) does not require any additional function evaluation if the underlying RK method is a Lobatto method.

In [7] was shown, by using the B-series theory, that if $p$ is the largest integer such that

$$b_{i,\rho}^- \Phi(\tau) = b_{i,\rho}^- \Phi(\tau) = 0, \quad \forall \tau \in T, \quad 1 \leq \rho(\tau) \leq p, \quad \forall i = 1, \ldots, s, \quad (10)$$

then the starting method has order $p$. Here $T$ is the set of the rooted trees, $\rho(\tau)$ the order of $\tau$, $b_{i,\rho}^-$ is the vector whose components are the coefficients $b_{i,j}$, $j = 0, 1, 2, \ldots$, of the initial value $\hat{Y}_i^0$, and the elementary weight functions $\Phi(\tau)$ and $\Phi(\tau)$ are computed from certain tableaux that are associated, respectively, with $\hat{Y}_i$ (tableau (3) in [7]) and $\hat{Y}_i^0$. Eq. (10) will be referred to as the order equations.

Taking into account those B-series, the error of a starting algorithm of order $p$ can be written as

$$\hat{Y}_i - \hat{Y}_i^0 = \left[ \sum_{j=1}^{N_{p+1}} C_{p+1,j} F(\tau_{p+1,j})(y_0) \right] h^{p+1} + O(h^{p+2}), \quad i = 1, \ldots, s,$$
where $\tau_{p+1,j}$ represents each of the $N_{p+1}$ rooted trees with exactly $p+1$ vertices, $F(\tau_{p+1,j})(\gamma_0)$ is the elementary differential of the derivative function $F$ corresponding to the tree $\tau_{p+1,j}$ evaluated at $\gamma_0$, and $C'_{p+1,j}$ are the constants

$$C'_{p+1,j} = \frac{\pi(\tau_{p+1,j})y(\tau_{p+1,j})}{(p+1)!}(\hat{b}'_j \hat{\Phi}(\tau_{p+1,j}) - \hat{b}'_j \hat{\Phi}(\tau_{p+1,j})),$$

with $\pi(\tau_{p+1,j})$ the number of the monotonic labellings of $\tau_{p+1,j}$ and $y(\tau_{p+1,j})$ its density. We will denote

$$C'_{p+1} = (C'_{p+1,1}, \ldots, C'_{p+1,N_{p+1}})^T, \quad i = 1, \ldots, s, \quad \text{and} \quad C_{p+1} = ((C'_{p+1})^T, \ldots, (C'_{p+1})^T)^T,$$

the vectors whose components are the error constants for $\hat{Y}_0$ and $\hat{Y}_0 = (\hat{Y}_0)^T$, respectively. The ordering used for the rooted trees $\tau_{i,j}$ throughout this paper is the same as in [3, p. 147], (where they are denoted by $t_{i,j}$). We will denote by $\tau_{1,1}$ the unique tree with one node which in the mentioned reference is denoted by $\tau$.

As usual [3, II.7], we will denote by $B(p)$ and $C(q)$ the following simplifying assumptions:

$$B(p): \sum_{i=1}^{s} b_i C_i^{k-1} = \frac{1}{k}, \quad k = 1, \ldots, p,$$

$$C(q): \sum_{j=1}^{s} a_{ij} C_j^{k-1} = \frac{c_i^j}{k}, \quad i = 1, \ldots, s, \quad k = 1, \ldots, q.$$

It is well known that the $s$-stage collocation RK methods satisfy $C(s)$ and so, this condition is, in particular, satisfied by the $s$-stage RK–Gauss formula, which also satisfies the condition $B(2s)$. As a consequence, it is easy to check [7] that the tableaux used to compute $\hat{\Phi}(\tau)$ and $\hat{\Phi}(\tau)$ in (10) satisfy the condition $C(s)$. Therefore, some order equations turn out to be scalar multiples of some others and so it is not necessary to consider them. This is the reason why some reductions in the number of the order equations will be considered throughout this paper.

### 3. Starting algorithms for the 2-stage RK–Gauss

We consider the Gauss method with 2 stages given in [3, p. 203] for which we have constructed the following starting algorithms:

(a) **Starting method of order 2**: The starting algorithm of type (7) and maximum order is the one given by the collocation polynomial and it is of order 2. Its coefficients can be computed in the way exposed in Theorem 2.7 in [7] and they turn out to be

$$b_{1,1}, b_{2,2} = \frac{3(2 + 2r + r^2) \mp 2\sqrt{3}r(2+r)}{12}, \quad b_{1,2}, b_{2,1} = \frac{3(2 - r^2) \pm 2\sqrt{3}r(1+r)}{12}.$$

This notation will be used throughout this paper to simplify the presentation. If the $\mp$ symbol is used in an expression, the top sign ($-$) corresponds to the left most coefficient ($b_{1,1}$), and the bottom sign ($+$) to the right most coefficient ($b_{2,2}$). An analogous interpretation has to be done with the $\pm$ symbol.
The error constants are

\[
C_{3,1}^1, C_{3,1}^2 = \frac{r(1 + r)}{72} [3(1 + r) \pm \sqrt{3}(1 + 2r)], \quad C_{3,2}^1 = C_{3,1}^1, \quad C_{3,2}^2 = C_{3,1}^2.
\]

By computing the norm of the vectors \(C_i = (C_i,1, C_i,2) \in \mathbb{R}^2, i = 1, 2\), we have an estimation of the magnitude of those errors, since the elementary differentials depend on the differential problem. With the euclidean norm it is obtained for \(r \in [0.1, 2]\):

\[
0.263 \cdot 10^{-2} < \|C_1^1\| < 0.432 \cdot 10^{-1}, \quad 0.116 \cdot 10^{-1} < \|C_2^2\| < 2.082.
\]

(b) Starting methods of order 3: According to Theorem 3.3 in [7], there exists a two-parameter family of order 3. If order 4 is required for quadratures we have the only starting algorithm of type (8) given by

\[
b_{1,0}, b_{2,0} = \frac{r^2(1 + r)}{6} \left[ -1 - 2r \pm \sqrt{3} \frac{1 + 3r}{2} \right],
\]

\[
b_{1,1}, b_{2,2} = \frac{2 - 5r^2 - 7r^3 - 3r^4}{4} \pm \sqrt{3} \frac{r^2(9 + 11r + 4r^2)}{12},
\]

\[
b_{1,2}, b_{2,1} = \frac{2 + r^2 + 5r^3 + 3r^4}{4} \mp \sqrt{3} \frac{r^2(1 + r)(3 + 4r)}{12},
\]

\[
b_{1,3}, b_{2,3} = \frac{r(1 + r)}{6} \left[ 3 + 4r + 2r^2 \mp \sqrt{3} \frac{(1 + r)(2 + 3r)}{2} \right].
\]

The order equation for \(\hat{Y}_1^0\) corresponding to the tree \(T_{4,3}\) (and so to \(T_{4,4}\)), is only satisfied for \(r = \frac{1}{2} (\sqrt{3} - 1)\) when \(r \in [0.1, 2]\), whereas the equation for \(\hat{Y}_2^0\) corresponding to the same tree is never satisfied in that interval.

The euclidean norms of the vectors \(C_i = (0, 0, C_i,3, C_i,4) \in \mathbb{R}^4, i = 1, 2\), whose components are the corresponding coefficients of the error \(\hat{Y}_i - \hat{Y}_i^0\), satisfy for \(0.1 \leq r \leq 2\)

\[
0 \leq \|C_i^3\| < 0.141, \quad 4.32 \cdot 10^{-4} < \|C_i^4\| < 1.09.
\]

If we had only considered the function evaluation corresponding to the final point of the previous step (and not the initial point), it can be checked that it is also possible to obtain initial values of order 3, but in this case no order equation for trees with four nodes are satisfied for \(r \in [0.1, 2]\).

By using \(\hat{Y}_1^0\) in the way described in [7, Remark 3.5], an approximation \(\hat{Y}_2^0\) of order 4 can be obtained in the form

\[
\hat{Y}_2^0 = y_0 + h[b_{2,0}F(y_0) + b_{2,1}F(Y_1) + b_{2,2}F(Y_2) + b_{2,3}F(Y_1) + b_{2,4}F(\hat{Y}_1^0)].
\]
In this case tableau (18) in [7] is required to compute \( \hat{\Phi}(\tau) \) in the order equations (10) for \( i = 2 \) and \( \hat{b}_2 = (b_{2,j})_{j=0}^3 \). The coefficients turn out to be \((r \neq 0, -1, 1 \pm \sqrt{3})\)

\[
\hat{b}_2 = \left( -\frac{r^3(1 - r)(1 + r)(1 + 2r + \sqrt{3}(1 + r))}{4(2 + 2r - r^2)} \right),
\]

\[
\frac{4 + 4r - 2r^2 + 2r^3 + 2r^4 - 4r^5 - 3r^6 + 2\sqrt{3}r^3(1 - r)(1 + r)^2}{4(2 + 2r - r^2)},
\]

\[
\frac{4 + 4r - 2r^2 + 8r^4 + 8r^5 + 3r^6 + 2\sqrt{3}r^3(1 + r)(1 + r^2)}{4(2 + 2r - r^2)},
\]

\[
\frac{r(1 + r)^2(2 + r^2)(1 + 2r + \sqrt{3}(1 + r))}{4(2 + 2r - r^2)} \left( \frac{r[9(1 + r)^2 + \sqrt{3}(5 + 11r + 5r^2)]}{6(2 + 2r - r^2)} \right),
\]

and the euclidean norm of the vector with the error constants \( C^2 = (C_{2,j})_{j=1}^9 \) satisfies for \( r \in [0.1, 2] \)

\[
0.333 \cdot 10^{-4} < \|C^2\| < 6.035.
\]

(c) Starting method of order 4: We are going to give the coefficients of a starting algorithm of type (9) and maximal order which has been obtained for the two stage RK-Gauss formula when it is applied with fixed stepsize. We have 15 parameters:

\[
\mu, \mu_0, \mu_1, \mu_2, \mu_3, \ b_{1,0}, b_{1,1}, b_{1,2}, b_{1,3}, b_{1,4} \quad \text{and} \quad b_{2,0}, b_{2,1}, b_{2,2}, b_{2,3}, b_{2,4}.
\]

The order equations to get \( \hat{\gamma}_0 \) and \( \hat{\gamma}_2 \) of order 4 are 10. If we impose the two simplifying equations:

\[
\mu_0 + \mu_1 + \mu_2 + \mu_3 = \mu, \quad \mu_1\mu_3 + \mu_2\mu_3 + \mu_3 = \mu^2/2,
\]

we can follow the proof of Theorem 4.3 in [7] and so we can obtain

- \( \mu_1 \) and \( \mu_2 \) as some functions of \( \mu, \mu_0 \) and \( \mu_3 \).
- \( b_{1,4} \) and \( b_{2,4} \) as some functions of \( \mu_0 \) and \( \mu_3 \) under the condition \( \mu_0 + \mu_3 \neq 0.\)
- Finally, \( b_{i,j} \), \( i = 1, 2, j = 0, 1, 2, 3 \), as some functions of \( \mu, \mu_0 \) and \( \mu_3 \).

So we obtain \( \hat{\gamma}_0 \) and \( \hat{\gamma}_2 \) of order 4 and there still remains three parameters: \( \mu, \mu_0 \) and \( \mu_3 \). To choose their values we consider the order equations for the trees with five vertices. Notice that as a consequence of the simplifying assumption \( C(2) \) being satisfied by the RK method, it is enough to study the equations corresponding to \( \tau_{5,1}, \tau_{5,3}, \tau_{5,6} \) and \( \tau_{5,8} \), and we have

- The equations for \( \hat{\gamma}_0 \) and \( \hat{\gamma}_2 \) corresponding to \( \tau_{5,8} \) are independent of the parameters \( \mu, \mu_0 \) and \( \mu_3 \) and they are not satisfied.
- The equations corresponding to the same tree among \( \tau_{5,1}, \tau_{5,3} \) or \( \tau_{5,6} \) for \( \hat{\gamma}_0 \) and \( \hat{\gamma}_2 \) are incompatible. So at most three order equations can be satisfied simultaneously.

We have studied all the possible cases for which exactly three of these equations are satisfied and we have chosen the parameters \( \mu, \mu_0, \mu_3 \) which minimize the euclidean norm \( \|C_5\| \), where \( C_5 = ((C_{5,1})^T, (C_{5,2})^T)^T = (C_{5,1}, \ldots, C_{5,9}, C_{5,1}, \ldots, C_{5,9}) \in \mathbb{R}^{18} \) is the vector of the error constants. It is obtained that the smallest value is

\[
\|C_5\| \simeq 2.96 \cdot 10^{-2}
\]
and it appears in case the order equations for $\hat{Y}_2^0$ corresponding to $\tau_{5.1}, \tau_{5.3}$ and $\tau_{5.6}$ (and its equivalent equations) are satisfied. Then, if we denote by $\hat{\mu}^T=(\mu, \mu_0, \mu_1, \mu_2, \mu_3)$, $\hat{b}_1^T=(b_{1,j})_{j=0}^4$ and $\hat{b}_2^T=(b_{2,j})_{j=0}^4$, we have

$$\hat{\mu}^T = \left(2 - \frac{\sqrt{3}}{6}, \frac{-81 + 41\sqrt{3}}{8}, \frac{-341 + 225\sqrt{3}}{16}, \frac{759 - 557\sqrt{3}}{48}, \frac{141 - 62\sqrt{3}}{8}\right),$$

$$\hat{b}_1^T = \left(0, \frac{-9 + 16\sqrt{3}}{36}, \frac{123 - 34\sqrt{3}}{276}, \frac{14 - 5\sqrt{3}}{11}, \frac{4(18 - 19\sqrt{3})}{2277}\right),$$

$$\hat{b}_2^T = \left(0, \frac{15 + 2\sqrt{3}}{36}, \frac{33 - 8\sqrt{3}}{92}, \frac{6 + \sqrt{3}}{11}, \frac{4(102 + 61\sqrt{3})}{2277}\right).$$

4. Starting algorithms for the 3-stage RK–Gauss

We will show now the initial values obtained for the Gauss method with three stages whose coefficients are given in [3, p. 204]. It can be checked that the tableaux to compute $\Phi(\tau)$ and $\Phi(\tau)$ in (10) satisfy the condition $C(3)$ and so, to study the order of the starting algorithms for this formula we only have to consider, among the trees with six or less vertices, those whose function $\Phi$ is: $e$, $c$, $c^2$, $c^3$, $Ac^3$, $c^5$, $Ac^3c$, $Ac^4$ and $A^2c^3$.

(a) Starting method of order 3: The coefficients of the unique starting method without additional cost of type (7) and maximal order, in this case order 3, are given by the following expressions [7, Theorem 2.7]:

$$b_{1,1}, b_{3,3} = \frac{5(10 + 24r + 33r^2 + 14r^3)}{180}, \quad b_{1,2}, b_{3,2} = \frac{5(4 - 3r - 12r^2 - 7r^3)}{45}, \quad b_{1,3}, b_{3,1} = \frac{5(10 + 6r + 15r^2 + 14r^3)}{180},$$

$$b_{2,1}, b_{2,3} = \frac{(2+r)(10(1 + r + r^2) + 3\sqrt{15}r)}{72}, \quad b_{2,2} = \frac{(2+r)(4 - 5r - 5r^2)}{18}.$$

None of the order equations corresponding to the rooted trees with 4 vertices is satisfied for $r \in [0.1, 2]$. For the euclidean norm of the vectors $\tilde{C}_i^\ell \in \mathbb{R}^4$, $i = 1, 2, 3$, whose components are the error constants corresponding to $\hat{Y}_i^0$, it follows that

$$3.47 \cdot 10^{-4} < \|C_1^\ell\| < 4.21 \cdot 10^{-2}, \quad 1.91 \cdot 10^{-3} < \|C_2^\ell\| < 0.607, \quad 4.13 \cdot 10^{-3} < \|C_3^\ell\| < 3.67.$$

(b) Starting methods of order 4: Firstly, a starting algorithm of the type (8) and maximal order has been obtained according to [7, Remark 3.4]. So, we get initial values of order 4 which also
satisfy the order equations corresponding to the tree \( \tau_{5,1} \) (and the equations equivalent to those). The coefficients are

\[
\begin{align*}
&b_{1,0}, b_{3,0} = \frac{r^2(1 + r)}{5} \left[ 1 + 6r + 6r^2 \mp \sqrt{15} \frac{5 + 31r + 29r^2}{20} \right], \\
&b_{1,1}, b_{3,3} = \frac{10 - 105r^2 - 256r^3 - 228r^4 - 72r^5}{36} \mp \frac{r^2(45 + 110r + 96r^2 + 29r^3)}{60}, \\
&b_{1,2}, b_{3,2} = \frac{2(10 + 12r^2 + 77r^3 + 96r^4 + 36r^5)}{45} \mp \frac{r^2(10 + 66r + 80r^2 + 29r^3)}{75}, \\
&b_{1,3}, b_{3,1} = \frac{10 - 15r^2 - 94r^3 - 156r^4 - 72r^5}{36} \mp \frac{r^2(r + 1)(5 + 35r + 29r^2)}{60}, \\
&b_{1,4}, b_{3,4} = \frac{r(1 + r)}{10} \left[ (1 + r)(1 + 2r)(5 + 6r) \mp \frac{10 + 55r + 71r^3 + 29r^5}{10} \right], \\
&b_{2,0} = \frac{r^2(1 + r)(1 + 3r)}{8}, \quad b_{2,1}, b_{2,3} = \frac{5(4 - 15r^2 - 25r^3 - 12r^4)}{72} \mp \frac{r^2(5 + 5r + r^2)}{24}, \\
&b_{2,2} = \frac{4 + 3r^2 + 11r^3 + 6r^4}{9}, \quad b_{2,4} = \frac{r(1 + r)^2(4 + 5r)}{8}. 
\end{align*}
\]  

(12)

The order equation corresponding to the tree \( \tau_{5,6} \) for \( \hat{Y}^0_1 \) is not satisfied for \( r \in [0.1, 2] \) and it is the same for \( \hat{Y}^0_2 \) and \( \hat{Y}^0_3 \).

When the stepsize is taken constant, these are the coefficients that appear.

The euclidean norms of the vectors \( C^i \in \mathbb{R}^9 \), whose components, (five of them zero), are the coefficients in the principal term of the error \( \hat{Y}_i - \hat{Y}^0_i \), satisfy for \( r \in [0.1, 2] \) that

\[
1.99 \cdot 10^{-6} < \|C^1\| < 3.99 \cdot 10^{-2}, \quad 4.36 \cdot 10^{-5} < \|C^2\| < 0.13, \quad 1.58 \cdot 10^{-4} < \|C^3\| < 1.63.
\]

We can improve the order of the initial approximation to the second stage if we use \( \hat{Y}^0_1 \) for the computation of \( \hat{Y}^0_2 \) according to the formula

\[
\hat{Y}^0_2 = y_0 + h[b_{2,0}F(y_0) + b_{2,1}F(Y_1) + b_{2,2}F(Y_2) + b_{2,3}F(Y_3) + b_{2,4}F(y_1) + b_{2,5}F(\hat{Y}^0_1)].
\]

Now, the order equations are (10) with \( i = 2 \), \( \hat{b}_{2} = (b_{2,j})_{j=0}^{5} \) and \( \hat{\Phi}(\tau) \) is computed from tableau (18) in [7]. Thus, an initial value \( \hat{Y}^0_2 \) of order 5 is obtained. For brevity, we only give its coefficients in case the stepsize is constant which are

\[
\hat{b}_{2} = \left( \frac{\sqrt{15}}{10}, -14 + \frac{\sqrt{15}}{24}, \frac{2(5 + \sqrt{15})}{15}, 10 - 3\sqrt{15}, -2\sqrt{15}, 4 + \sqrt{15} \right). 
\]

By proceeding in an analogous way, we have obtained an initial approximation \( \hat{Y}^0_3 \) to \( \hat{Y}_3 \) in the form

\[
\hat{Y}^0_3 = y_0 + h[b_{3,0}F(y_0) + b_{3,1}F(Y_1) + b_{3,2}F(Y_2) + b_{3,3}F(Y_3) + b_{3,4}F(y_1) + b_{3,5}F(\hat{Y}^0_1) + b_{3,6}F(\hat{Y}^0_2)], 
\]
which is of order 5 and satisfies 7 order equations corresponding to some trees with six vertices. If the step length is constant, those coefficients are given by

\[ b_3 = \left( \frac{580 + 143\sqrt{15}}{1047}, \frac{5(3319 + 642\sqrt{15})}{12564}, \frac{-2(22354 + 8437\sqrt{15})}{53397}, \frac{5(62429 + 12538\sqrt{15})}{213588}, \frac{4(580 + 143\sqrt{15})}{1047}, \frac{1533675 + 396901\sqrt{15}}{533970}, \frac{204330 + 53549\sqrt{15}}{266985} \right). \]

(c) Starting method of order 5: If we consider starting algorithms of type (9) for the 3-stage RK-Gauss formula, we have 24 parameters

\[ \mu, \mu_0, \mu_1, \mu_2, \mu_3, \mu_4 \quad \text{and} \quad b_{i,0}, b_{i,1}, b_{i,2}, b_{i,3}, b_{i,4}, b_{i,5}, \quad (i = 1, 2, 3). \]

To get the three initial values of order 5, we need to solve the equations corresponding to \( \tau_{1,1}, \tau_{2,1}, \tau_{3,1}, \tau_{4,1}, \tau_{5,1}, \text{and} \) \( \tau_{5,6} \) for each of the approximations, so 18 equations. Moreover, we impose as simplifying conditions

\[ \mu_0 + \mu_1 + \mu_2 + \mu_3 + \mu_4 = \mu, \quad \mu_1 c_1 + \mu_2 c_2 + \mu_3 c_3 + \mu_4 = \mu^2/2, \quad \mu_1 c_1^2 + \mu_2 c_2^2 + \mu_3 c_3^2 + \mu_4 = \mu^3/3, \]

and therefore we have 21 equations. From them and by applying Theorem 4.3 in [7], we can obtain \( \tilde{y}_i, i = 1, 2, 3, \) of order 5, and three parameters still remain: \( \mu, \mu_0 \) and \( \mu_4 \) satisfying \( \mu_0 \neq \mu_4 \). Their choice has been made as follows, taking into account that we only have to study the trees whose function \( \Phi \) is

\[ \Phi(\tau_{6,1}) = c^5, \quad \Phi(\tau_{6,6}) = Ac^3, \quad \Phi(\tau_{6,12}) = Ac^4, \quad \Phi(\tau_{6,17}) = A^2 c^3, \]

among the rooted trees with exactly six vertices. Then

- The order equations for \( \tau_{6,17} \) corresponding to the three initial values are not satisfied.
- The order equations obtained for \( \tau_{6,1} \) in each of the three approximations are two by two incompatible, and the same happens for \( \tau_{6,6} \) and \( \tau_{6,12} \). So, at most three of these equations will be satisfied.
- For all the cases in which exactly three of these equations are satisfied, we have computed the euclidean norm of the vector \( C_6 = ((C_6^1)^T, (C_6^0)^T, (C_6^3)^T)^T \in \mathbb{R}^{30} \) whose components are the coefficients that appear in the principal term of the error \( \hat{y} - \hat{y}_0 \). The lower value obtained is

\[ ||C_6|| \approx 2.81 \cdot 10^{-2}, \]

and it corresponds to the fact that the three order equations (for \( \tau_{6,1}, \tau_{6,6}, \text{and} \) \( \tau_{6,12} \)) corresponding to \( \hat{y}_3^0 \) are satisfied. In this case, by denoting by \( \hat{\mu}^T = (\mu, \mu_0, \mu_1, \mu_2, \mu_3, \mu_4) \) and \( \hat{b}_i^T = (b_{i,j})_{j=0}^5, i = 1, 2, 3, \) these are the parameters obtained:

\[ \hat{\mu}^T = \left( \frac{59 - 2\sqrt{15}}{30}, \frac{7(31143013601 - 6956187460\sqrt{15})}{6966202500}, \frac{-668327582554 + 156756279455\sqrt{15}}{8359443000}, \frac{275734492307 - 59280182800\sqrt{15}}{5224651875}, \frac{-493079014294 + 91083115325\sqrt{15}}{8359443000}, \frac{7(56555992001 - 11323383790\sqrt{15})}{6966202500} \right). \]
\[
\hat{b}_1 = \left(\frac{-634 + 163\sqrt{15}}{1555}, \frac{-60387 + 17308\sqrt{15}}{23052}, \frac{1427 - 127\sqrt{15}}{2345}, \frac{84705 - 12122\sqrt{15}}{93660}, \frac{4(2894 - 743\sqrt{15})}{3905}, \frac{2025(1204798955 - 264806436\sqrt{15})}{48799625993657}\right),
\]

\[
\hat{b}_2 = \left(\frac{-1115 + 36\sqrt{15}}{3110}, \frac{-9960 + 11927\sqrt{15}}{46104}, \frac{520 - 19\sqrt{15}}{2345}, \frac{5(1320 - 1411\sqrt{15})}{37464}, \frac{2(3205 - 156\sqrt{15})}{3905}, \frac{2025(847099280 + 145573211\sqrt{15})}{48799625993657}\right),
\]

\[
\hat{b}_3 = \left(\frac{-94 + 19\sqrt{15}}{1555}, \frac{45105 + 2162\sqrt{15}}{115260}, \frac{857 - 25\sqrt{15}}{2345}, \frac{93 - 1444\sqrt{15}}{18732}, \frac{4(554 + 119\sqrt{15})}{3905}, \frac{2025(5571995285 + 1429392124\sqrt{15})}{48799625993657}\right).
\]

5. Starting algorithms for the 4-stage RK–Gauss

The Butcher’s tableau for the RK–Gauss formula with four stages is given in [3, p. 205]. As it is well known, this formula satisfies the simplifying assumptions \(B(8)\) and \(C(4)\) and, as a consequence, the tableaux to compute \(\Phi(\tau)\) and \(\Phi(\tau)\) in (10) satisfy the condition \(C(4)\). Therefore, the rooted trees to consider with five or less vertices are only \(\tau_{1,1}, \tau_{2,1}, \tau_{3,1}, \tau_{4,1}\) and \(\tau_{5,1}\). With respect to the rooted trees of order 6, we must only study the order equations corresponding to

\[\tau_{6,1} = \text{\textbullet:\textbullet:\textbullet:::} \quad \text{and} \quad \tau_{6,12} = \text{\textbullet:::}\]

and among the rooted trees with seven vertices, it will be enough to consider the equations for

\[\tau_{7,1} = \text{\textbullet:\textbullet:\textbullet:::} \quad \tau_{7,12} = \text{\textbullet:::} \quad \tau_{7,29} = \text{\textbullet:::} \quad \text{and} \quad \tau_{7,40} = \text{\textbullet:::}\]

For this numerical method, we have constructed some starting algorithms of maximal order of the three considered types. By simplicity, we will only consider that the length of the step size is always the same.

(a) Starting method of order 4: The coefficients of the unique starting algorithm of type (7) and order 4 can be obtained by solving, for each initial approximation, the order equations corresponding
to the trees $\tau_{1,1}, \tau_{2,1}, \tau_{3,1}$ and $\tau_{4,1}$ (see [7, Theorem 2.7]). They are

$$
\begin{align*}
 b_{1,1}, b_{4,4} &= \frac{\alpha_1 \pm \sqrt{35}\beta_1}{1680}, & b_{2,2}, b_{3,3} &= \frac{\alpha_2 \pm \sqrt{35}\beta_2}{1680}, \\
 b_{1,4}, b_{4,1} &= \frac{\alpha_3 \pm \sqrt{35}\beta_3}{1680}, & b_{2,3}, b_{3,2} &= \frac{\alpha_4 \pm \sqrt{35}\beta_4}{1680}, \\
 b_{1,2}, b_{4,3} &= \frac{\alpha_5 \pm \sqrt{35}\beta_5}{11760}, & b_{1,3}, b_{4,2} &= \frac{\alpha_6 \pm \sqrt{35}\beta_6}{11760}, \\
 b_{2,1}, b_{3,4} &= \frac{\alpha_7 \pm \sqrt{35}\beta_7}{11760}, & b_{2,4}, b_{3,1} &= \frac{\alpha_8 \pm \sqrt{35}\beta_8}{11760},
\end{align*}
$$

(14)

where

$$
\begin{align*}
 \alpha_1, \alpha_2 &= 35(54 \pm 37\sqrt{30}), & \beta_1, \beta_2 &= 2(80 \mp 41\sqrt{30})\sqrt{15 \pm 2\sqrt{30}}, \\
 \alpha_3, \alpha_4 &= 35(6 \mp 19\sqrt{30}), & \beta_3, \beta_4 &= -8(22 \mp 7\sqrt{30})\sqrt{15 \pm 2\sqrt{30}}, \\
 \alpha_5, \alpha_6 &= 735(2 - 3\sqrt{30}) \mp 980(4 + \sqrt{30})\sqrt{105}, & \\
 \beta_5, \beta_6 &= -7(4 - 13\sqrt{30})\sqrt{15 + 2\sqrt{30}} \pm 3(1216 + 269\sqrt{30})\sqrt{15 - 2\sqrt{30}}, \\
 \alpha_7, \alpha_8 &= 735(2 + 3\sqrt{30}) \mp 980(4 - \sqrt{30})\sqrt{105}, & \\
 \beta_7, \beta_8 &= -7(4 + 13\sqrt{30})\sqrt{15 - 2\sqrt{30}} \pm 3(1216 - 269\sqrt{30})\sqrt{15 + 2\sqrt{30}}.
\end{align*}
$$

It can be checked that these initial approximations satisfy no order equation corresponding to the trees with five vertices.

(b) Starting method of order 5: A starting algorithm with one additional function evaluation per step of type (8) has been constructed for the 4-stage RK–Gauss formula according to [7, Remark 3.4]. In this way, we have obtained the unique starting algorithm of type (8) and order 5, which satisfies some order equations corresponding to the trees with six vertices. More specifically, for each initial value, 11 of these equations are satisfied, of a total of 20. The values of the parameters
\[b_{1,0}, b_{4,0} = \frac{2(x_1 \pm \sqrt{35}\beta_1)}{735}, \quad b_{2,0}, b_{3,0} = \frac{2(x_2 \pm \sqrt{35}\beta_2)}{735},\]
\[b_{1,1}, b_{4,4} = \frac{x_3 \pm \sqrt{35}\beta_3}{5040}, \quad b_{2,2}, b_{3,3} = \frac{x_4 \pm \sqrt{35}\beta_4}{5040},\]
\[b_{1,4}, b_{4,1} = \frac{x_5 \pm \sqrt{35}\beta_5}{5040}, \quad b_{2,3}, b_{3,2} = \frac{x_6 \pm \sqrt{35}\beta_6}{5040},\]
\[b_{1,5}, b_{4,5} = \frac{x_7 \pm \sqrt{35}\beta_7}{735}, \quad b_{2,5}, b_{3,5} = \frac{x_8 \pm \sqrt{35}\beta_8}{735},\]
\[b_{1,2}, b_{4,3} = \frac{x_9 \pm \sqrt{35}\beta_9}{35280}, \quad b_{1,3}, b_{4,2} = \frac{x_{10} \pm \sqrt{35}\beta_{10}}{35280},\]
\[b_{2,1}, b_{3,4} = \frac{x_{11} \pm \sqrt{35}\beta_{11}}{35280}, \quad b_{2,4}, b_{3,1} = \frac{x_{12} \pm \sqrt{35}\beta_{12}}{35280},\]

where

\[\alpha_1, \alpha_2 = -35(163 \pm 18\sqrt{30}), \quad \beta_1, \beta_2 = 13(18 \pm \sqrt{30})\sqrt{15 \pm 2\sqrt{30}},\]
\[\alpha_3, \alpha_4 = -105(1846 \pm 443\sqrt{30}), \quad \beta_3, \beta_4 = 2(2330 \pm 933\sqrt{30})\sqrt{15 \pm 2\sqrt{30}},\]
\[\alpha_5, \alpha_6 = 1785(50 \pm 13\sqrt{30}), \quad \beta_5, \beta_6 = -4(367 \pm 258\sqrt{30})\sqrt{15 \pm 2\sqrt{30}},\]
\[\alpha_7, \alpha_8 = 70(415 \pm 39\sqrt{30}), \quad \beta_7, \beta_8 = -(1203 \pm 47\sqrt{30})\sqrt{15 \pm 2\sqrt{30}},\]
\[\alpha_9, \alpha_{10} = -15435(2 - 3\sqrt{30}) \pm 4(11985 + 4643\sqrt{30})\sqrt{105},\]
\[\beta_9, \beta_{10} = 21(296 - 115\sqrt{30})\sqrt{15 + 2\sqrt{30}} \mp (53216 + 13731\sqrt{30})\sqrt{15 - 2\sqrt{30}},\]
\[\alpha_{11}, \alpha_{12} = -15435(2 + 3\sqrt{30}) \pm 4(11985 - 4643\sqrt{30})\sqrt{105},\]
\[\beta_{11}, \beta_{12} = 21(296 + 115\sqrt{30})\sqrt{15 - 2\sqrt{30}} \mp (53216 - 13731\sqrt{30})\sqrt{15 + 2\sqrt{30}}.\]

(c) Starting method of order 6: When we consider starting algorithms of type (9) for the 4-stage RK–Gauss formula, we have the following 35 parameters:

\[\mu, \mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5 \quad \text{and} \quad b_{i,0}, b_{i,1}, b_{i,2}, b_{i,3}, b_{i,4}, b_{i,5}, b_{i,6}, \quad (i = 1, 2, 3, 4).\]
We want to choose them so that the order of the algorithm be maximal and to get it, we impose as simplifying equations

\[(\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \begin{pmatrix} 1 & 0 & 0 & 0 \\ e & c & c^2 & c^3 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \left( \mu, \frac{\mu^2}{2}, \frac{\mu^3}{3}, \frac{\mu^4}{4} \right),\]

and then, we apply Theorem 4.3 in [7]. In this way we have a starting method of order at least 6 with three parameters yet: \(\mu, \mu_0\) and \(\mu_5\) with \(\mu_0 + \mu_5 \neq 0\). The order equation for \(\tau_{7,40}\) is not satisfied for any of the four initial values and so, the order is exactly 6. Moreover, any two of the four-order equations for \(\tau_{7,1}\) corresponding to \(\hat{\gamma}_{i,1}^0, i = 1,2,3,4\), have no solution, and it is the same for the equations corresponding to \(\tau_{7,12}\) and also to \(\tau_{7,29}\). In an analogous way that for 2- and 3-stage RK–Gauss formulae, we have chosen the three parameters by imposing that \(\hat{\gamma}_{i,0}^0\) satisfies the order equations for \(\tau_{7,1}, \tau_{7,12}\) and \(\tau_{7,29}\). So, we have computed the following approximate values:

\[
\begin{align*}
\mu &= 1.752439944983398153319998, \\
\mu_0 &= -19.22194960433033959787503, \\
\mu_1 &= 33.1500653769116544701566, \\
\mu_2 &= -29.12574408730442130954662, \\
\mu_3 &= 40.7267699899863664682286, \\
\mu_4 &= -76.849120594315183615586, \\
\mu_5 &= 53.072418864053051737915,
\end{align*}
\]

\[
\begin{align*}
b_{1,0} &= 0.02382621291926910250153012, \\
b_{1,1} &= 0.1382997287353034523184078, \\
b_{1,2} &= 0.3379687649300072756830599, \\
b_{1,3} &= 0.3538193696309141933736000, \\
b_{1,4} &= 0.198504318718995585988311, \\
b_{1,5} &= 0.198504318718995585988311, \\
b_{1,6} &= -0.00083368623062634689747318, \\
b_{2,0} &= 0.2489526234398835987450237, \\
b_{2,1} &= -0.2391602382876643062022328, \\
b_{2,2} &= 0.627418810035880083348153, \\
b_{2,3} &= 0.1834660500882094559823366, \\
b_{2,4} &= -0.782088624305953179992323, \\
b_{2,5} &= 1.281507527514747282171668, \\
b_{2,6} &= 0.0091332972247350859379398, \\
b_{3,0} &= 1.365707789674594024625174, \\
b_{3,1} &= -2.03129022020199038663586, \\
b_{3,2} &= 1.718220962190131293192696, \\
b_{3,3} &= 0.11130505127374134629540468, \\
b_{3,4} &= -4.351795076162033168226277, \\
b_{3,5} &= 4.743994985124451853327049, \\
b_{3,6} &= 0.11384983171174182185087176, \\
b_{4,0} &= 0.334231081180214833690191, \\
b_{4,1} &= -0.3871971811568727092255462, \\
b_{4,2} &= 0.742783782546430085669908, \\
b_{4,3} &= 0.2383305613398502226403704, \\
b_{4,4} &= -2.102124923830589282234340, \\
b_{4,5} &= 2.656339510458857892875305, \\
b_{4,6} &= 0.4482098252591456716201739.
\end{align*}
\]
6. Numerical experiments

In this section some numerical results showing the behavior of the starting algorithms for the RK–Gauss formulae with \( s = 3, 4 \) stages are presented. The Gauss methods have been implemented with constant step sizes and following some of the ideas described in [4, IV.8]. Thus, to reduce the influence of round-off errors, the smaller quantities \( Z_i = Y_i - y_0, \ i = 1, \ldots, s, \) have been considered for the step from \( t_0 \) to \( t_1 = t_0 + h. \) Then (2) becomes

\[
Z_i = h \sum_{j=1}^{s} a_{ij} F(y_0 + Z_j), \quad i = 1, \ldots, s, \tag{17}
\]

and, since the matrix \( A \) of the RK-coefficients is nonsingular, the numerical solution has been computed as follows:

\[
y_1 = y_0 + \sum_{i=1}^{s} d_i Z_i,
\]

where \((d_1, \ldots, d_s) = b^T A^{-1}.\)

The initial values to start the iterations that solve (17) have to be also computed according to the new quantities. Thus, the starting values given in (7) can be written

\[
\hat{y}_1 = y_0 + \sum_{j=1}^{s} d_{i,j} Z_j, \quad i = 1, \ldots, s,
\]

where \((d_{1,1}, \ldots, d_{i,s}) = (b_{1,1}, \ldots, b_{i,s}) A^{-1}.\) Moreover, we can put \( \hat{y}_1 = y_1 + \hat{Z}_1, \) \( i = 1, \ldots, s, \) and so the initial values considered have been the following:

\[
\hat{y}_1 = \sum_{j=1}^{s} (d_{i,j} - d_j) Z_j, \quad i = 1, \ldots, s.
\]

We can proceed in an analogous way to compute the new initial values corresponding to the other two types of starting algorithms studied. Thus, those that correspond to the predicted values given in (8) can be written as

\[
\hat{y}_1 = \sum_{j=1}^{s} (d_{i,j} - d_j) Z_j + h [b_{i,0} F(y_0) + b_{i,s+1} F(y_1)], \quad i = 1, \ldots, s.
\]

Finally, for the initial values given in (9) we have

\[
\hat{y}_1 = \sum_{j=1}^{s} (d_{i,j} - d_j) Z_j + h [b_{i,0} F(y_0) + b_{i,s+1} F(y_1) + b_{i,s+2} F(Y_\mu)], \quad i = 1, \ldots, s,
\]

where the arbitrary stage \( Y_\mu \) can be written as

\[
Y_\mu = y_1 + \sum_{j=1}^{s} (v_j - d_j) Z_j + h [\mu_0 F(y_0) + \mu_{s+1} F(y_1)],
\]
and \((v_1, \ldots, v_s) = (\mu_1, \ldots, \mu_s) A^{-1}\).

To solve the system (17), simple fixed-point iteration (also called functional iteration)

\[ Z_{i+1}^k = h \sum_{j=1}^{s} a_{ij} F(y_0 + Z_j^k), \quad i = 1, \ldots, s, \quad k = 0, 1, \ldots, \]

and the quasi-Newton method

\[ (I - hA \otimes J) AZ_k = -Z_k + h(A \otimes I) F(Z_k), \quad k = 0, 1, \ldots, \]

\[ Z_{k+1} = Z_k + AZ_k, \]

where

\[ J = \frac{\partial F}{\partial y}(y_0), \quad Z_k = (Z_1^k, \ldots, Z_s^k) \text{T}, \quad F(Z_k) = (F(y_0 + Z_1^k), \ldots, F(y_0 + Z_s^k)) \text{T}, \]

will be used. Each iteration requires, in both schemes, \(s\) evaluations of the derivative function and the simplified Newton iterations require the solution of an \(sD\)-dimensional linear system too. The matrix \((I - hA \otimes J)\) is the same for all iterations of each step and so its LU-decomposition is done only once. Savings in the linear algebra are possible by applying the technique due to Butcher and Bickart [4, IV.8]. However, this has not been applied here because the main purpose of the numerical experiments presented in this paper is to show the performance of the high-order starting algorithms when the same iterative scheme is used. Thus, for each iterative scheme, we will measure the computational cost by the total number of function evaluations, without attempting to compare them.

To stop the iterative scheme, the test,

\[ \left| y_{i,j}^{k+1} - y_{i,j}^k \right| \leq \tau(j), \quad j = 1, \ldots, D, \]

has been used, where \(y_{i,j}^k\) is the \(j\)th component of the numerical solution \(y_i^k\) obtained from \(Z_k\), and \(\tau(j) = 10 \cdot U \max\{1, |y_{0,j}|\}\), with \(U\) the rounding unit of the computer. It is required that the iterations stop when the components of \(y_i^k\) and \(y_{i,j}^{k+1}\) have the largest possible number of equal significant digits.

If this test of convergence has not been attained after 12 iterations or if some quotient

\[ \theta_k = \frac{\|AZ_k\|}{\|AZ_{k-1}\|}, \quad k = 1, 2, 3, \ldots, \]

is greater than 0.9, we consider that there is no convergence and the corresponding step size has not been included.

The initial values which have been obtained can be computed from the second step. In the first one we have taken \(y_i^0 = y_0\) as initial approximations to the internal stages \(Y_i, i = 1, \ldots, s\).

All the computations have been carried out in a DEC-Alpha 3000, 500/S in double precision.

We will present the results obtained with three Hamiltonian problems. They are representative of the general behavior of the starting algorithms studied.
Problem 1. Kepler’s problem given by [13, p. 6]

\[ q_i' = p_i, \quad p_i' = -\frac{q_i}{(q_i^2 + q_j^2)^{3/2}}, \quad i = 1, 2, \]

with initial conditions

\[ q_1(0) = 1 - e, \quad q_2(0) = 0, \quad p_1(0) = 0, \quad p_2(0) = \sqrt{\frac{1 + e}{1 - e}}, \]

where \( e \) is a parameter, \( 0 \leq e < 1 \). As is well known, its solution is \( 2\pi \)-periodic and its projection onto the configuration \((q_1, q_2)\)-plane is an ellipse with eccentricity \( e \). As \( e \) increases, numerical methods have more difficulties for its integration. With large \( e \) (say \( e = 0.9 \)), we have to use smaller step sizes to get acceptable global errors and not to have any convergence problems.

Problem 2. Planar restricted three-body problem [15, p. 232] given by the system

\[
\begin{aligned}
q_1' &= p_1 + q_2, \\
p_1' &= p_2 - \frac{(1 - \mu)(q_1 + \mu)}{r_1^3} - \frac{\mu(q_1 - 1 + \mu)}{r_2^3}, \\
q_2' &= p_2 - q_1, \\
p_2' &= -p_1 - \left( \frac{1 - \mu}{r_1^3} + \frac{\mu}{r_2^3} \right) q_2.
\end{aligned}
\]

According to the coordinate system considered, \( r_1^2 = (q_1 + \mu)^2 + q_2^2 \) and \( r_2^2 = (q_1 - 1 + \mu)^2 + q_2^2 \). By taking \( \mu = \frac{1}{8245} \) and the initial values

\[ q_1(0) = 1.2, \quad q_2(0) = 0, \quad q_1'(0) = 0, \quad q_2'(0) = -1.04935750983, \]

the solution \((q_1(t), q_2(t))\) is a closed periodic orbit with period \( T = 6.192169331396 \ldots \). In [15] it was shown the necessity of using variable step size for this problem, which requires very different step sizes along the orbit to obtain a given accuracy. The integration with a fixed step size is too expensive because the length of \( h \) must be approximately as the smallest step size required to obtain the same accuracy when the step size varies.

Problem 3. Störmer’s problem [3, p. 420]:

\[
\begin{aligned}
q_1' &= p_1 - \frac{q_2}{r^3}, \quad q_2' = p_2 + \frac{q_1}{r^3}, \quad q_3' = p_3, \\
p_1' &= -\frac{3q_1q_2}{r^5} p_1 + \frac{3q_1^2 - r^2}{r^5} p_2 + \frac{q_1(2r^2 - 3q_3^2)}{r^8}, \\
p_2' &= \frac{r^2 - 3q_3^2}{r^5} p_1 + \frac{3q_1q_3}{r^5} p_2 + \frac{q_2(2r^2 - 3q_3^2)}{r^8}, \\
p_3' &= -\frac{3q_2q_3}{r^5} p_1 + \frac{3q_1q_3}{r^5} p_2 + \frac{3q_3(q_1^2 + q_2^2)}{r^8},
\end{aligned}
\]

where \( r^2 = q_1^2 + q_2^2 + q_3^2 \). The following initial values have been taken:

\[ q_1(0) = R_0, \quad q_2(0) = 0, \quad q_3(0) = z_0, \quad p_1(0) = \sqrt{Q_0} \cos u, \quad p_2(0) = 2\gamma/R_0, \quad p_3(0) = \sqrt{Q_0} \sin u, \]
Table 1
Problem 1, $e = 0$, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G3O3</th>
<th>G3O4</th>
<th>G3O5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>64</td>
<td>8.05</td>
<td>1545</td>
<td>7.06</td>
</tr>
<tr>
<td>128</td>
<td>6.02</td>
<td>2313</td>
<td>5.03</td>
</tr>
<tr>
<td>256</td>
<td>5.01</td>
<td>3849</td>
<td>4.02</td>
</tr>
<tr>
<td>512</td>
<td>4.01</td>
<td>6153</td>
<td>3.01</td>
</tr>
<tr>
<td>1024</td>
<td>3.46</td>
<td>10626</td>
<td>2.46</td>
</tr>
</tbody>
</table>

Table 2
Problem 1, $e = 0.5$, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G3O3</th>
<th>G3O4</th>
<th>G3O5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>64</td>
<td>8.28</td>
<td>1590</td>
<td>7.33</td>
</tr>
<tr>
<td>128</td>
<td>6.41</td>
<td>2460</td>
<td>5.41</td>
</tr>
<tr>
<td>256</td>
<td>5.18</td>
<td>3978</td>
<td>4.19</td>
</tr>
<tr>
<td>512</td>
<td>4.46</td>
<td>6843</td>
<td>3.46</td>
</tr>
<tr>
<td>1024</td>
<td>3.56</td>
<td>10932</td>
<td>2.56</td>
</tr>
</tbody>
</table>

where $\gamma = -0.5$, $u = \pi/4$, $R_0 = 0.257453$, $z_0 = 0.314687$, $r_0 = \sqrt{R_0^2 + z_0^2}$ and $Q_0 = 1 - (2\gamma/R_0 + R_0/r_0^2)^2$.

Its solution $(q_1(t), q_2(t), q_3(t))$ represents the trajectory of an electric particle in the magnetic field created by an elementary magnet.

We are going to compare some of the starting algorithms obtained in this paper for the 3-stage RK–Gauss formula:

- **G3O3**: Starting algorithm of order 3 without additional cost given in (11) for $r = 1$.
- **G3O4**: Starting algorithm of order 4 with one additional function evaluation per step obtained in (12) for $r = 1$.
- **G3O5**: Starting algorithm of order 5 with two additional function evaluations per step obtained in (13).

As usual, we will measure the work by the number of evaluations (NFE) of the derivative function $F$. Together with this, the number of iterations per step (Iter./step) will be presented (notice that each of them can be computed from the other one). In the problems with periodic solution it will be enough to study what happens with only one period. Tables 1–5 have been obtained by using functional iteration. None of the starting procedures has affected the overall accuracy of the numerical schemes because the iterations have been solved to the level of round-off. It can be remarked that:

- As $h$ decreases G3O3 requires approximately one more iteration than G3O4 and the same happens with G3O4 and G3O5. Thus, formula (5) is well reflected in these tables, which, furthermore, show that the starting algorithms’ error terms are similar, at least for the studied problems.
Table 3
Problem 1, $e = 0.9$, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G303</th>
<th>G304</th>
<th>G305</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>256</td>
<td>5.39</td>
<td>4143</td>
<td>4.42</td>
</tr>
<tr>
<td>512</td>
<td>4.40</td>
<td>6756</td>
<td>3.41</td>
</tr>
<tr>
<td>1024</td>
<td>3.66</td>
<td>11 253</td>
<td>2.67</td>
</tr>
<tr>
<td>2048</td>
<td>3.05</td>
<td>18 729</td>
<td>2.06</td>
</tr>
<tr>
<td>4096</td>
<td>2.49</td>
<td>30 639</td>
<td>1.49</td>
</tr>
</tbody>
</table>

Table 4
Problem 2, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G303</th>
<th>G304</th>
<th>G305</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>512</td>
<td>6.15</td>
<td>9450</td>
<td>5.19</td>
</tr>
<tr>
<td>1024</td>
<td>4.52</td>
<td>13 881</td>
<td>3.53</td>
</tr>
<tr>
<td>2048</td>
<td>3.70</td>
<td>22 713</td>
<td>2.70</td>
</tr>
<tr>
<td>4096</td>
<td>3.32</td>
<td>40 752</td>
<td>2.32</td>
</tr>
</tbody>
</table>

Table 5
Problem 3, functional iteration, $t \in [0, 10]$

<table>
<thead>
<tr>
<th>Steps</th>
<th>G303</th>
<th>G304</th>
<th>G305</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>40</td>
<td>8.60</td>
<td>1032</td>
<td>7.82</td>
</tr>
<tr>
<td>80</td>
<td>6.21</td>
<td>1491</td>
<td>5.30</td>
</tr>
<tr>
<td>160</td>
<td>4.77</td>
<td>2289</td>
<td>3.83</td>
</tr>
<tr>
<td>320</td>
<td>3.99</td>
<td>3831</td>
<td>3.00</td>
</tr>
<tr>
<td>640</td>
<td>3.40</td>
<td>6531</td>
<td>2.38</td>
</tr>
</tbody>
</table>

In Table 3 for $h = 2\pi/4096$, that expected difference of one iteration per step does not appear between G304 and G305 because at least one iteration is always done and G304 only requires one iteration in some steps.

- Starting algorithm G304 requires less function evaluations than G303 and algorithm G305 requires less function evaluations than G304. So the starting algorithms of higher order are the most efficient compensating for its additional cost per step.

Let us notice that the total number of function evaluations for this 3-stage RK formula, depending on either G303 or G304 being used, can be written as

$$G303 : 3I_1N, \quad G304 : (3I_2 + 1)N,$$

where $I_1, I_2$ are, respectively, the iterations per step required by each of the starting algorithms, and $N$ is the total number of steps. Thus $(3I_2 + 1)N < 3I_1N$ is equivalent to $I_1 - I_2 > \frac{1}{3}$. Furthermore,
if \( I_1 - I_2 \) is close to 1, then \((3I_2 + 1)N - 3I_1N \approx 2N\). These same ideas can be used for an \( s \)-stage RK formula and so we can say that a starting algorithm is more efficient than another one which requires one additional evaluation per step, when the difference \( I_1 - I_2 \) between the respective iterations per step be greater than \( 1/s \). Furthermore, if that difference is close to 1, the total saving is of approximately \((s - 1)N\) function evaluations.

Some comments can be made. In Problem 1 with \( e = 0 \) the orbit is a circle and the same number of iterations are required along it. In Table 1 the most of averages that appear are close to integer numbers because the iterations in the first step, where the starting algorithms cannot act, have been taken into account. For \( h = 2\pi/1024 \) (G303 and G304) the computed iteration error after a given number of iterations is very close to the error tolerance and as a consequence, that number varies between two consecutive values. Another observation is that the number of iterations per step decreases as \( h \) does, as was expected by the asymptotic behavior of the iteration error with respect to \( h \).

Tables 6–7 show some of the results obtained when the iterative scheme is quasi-Newton. From them we can say:
- In general, starting algorithm G303 requires approximately one more iteration per step than G305. This was expected according to (6) if the error terms of these algorithms are similar, and as a consequence, this last algorithm turns out to be more efficient than the first one.
- The lower number of function evaluations always corresponds to one of the two starting algorithms of higher order: G304 or G305.

As can be observed, there are some step sizes for which the reduction in the number of iterations due to the use of a starting algorithm of one more order than another one does not compensate for

<table>
<thead>
<tr>
<th>Steps</th>
<th>G303</th>
<th>G304</th>
<th>G305</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>256</td>
<td>3.38</td>
<td>2595</td>
<td>3.02</td>
</tr>
<tr>
<td>512</td>
<td>2.96</td>
<td>4554</td>
<td>2.50</td>
</tr>
<tr>
<td>1024</td>
<td>2.49</td>
<td>7638</td>
<td>2.24</td>
</tr>
<tr>
<td>2048</td>
<td>2.24</td>
<td>13 764</td>
<td>1.85</td>
</tr>
<tr>
<td>4096</td>
<td>2.12</td>
<td>26 046</td>
<td>1.41</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Steps</th>
<th>G303</th>
<th>G304</th>
<th>G305</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
</tr>
<tr>
<td>512</td>
<td>3.67</td>
<td>5643</td>
<td>3.50</td>
</tr>
<tr>
<td>1024</td>
<td>3.19</td>
<td>9786</td>
<td>2.58</td>
</tr>
<tr>
<td>2048</td>
<td>2.50</td>
<td>15 387</td>
<td>2.24</td>
</tr>
<tr>
<td>4096</td>
<td>2.23</td>
<td>27 450</td>
<td>2.12</td>
</tr>
</tbody>
</table>
Table 8
Problem 1, $e = 0$, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G404</th>
<th></th>
<th>G405</th>
<th></th>
<th>G406</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>9.00</td>
<td>1152</td>
<td>8.00</td>
<td>1057</td>
<td>7.00</td>
<td>961</td>
</tr>
<tr>
<td>64</td>
<td>7.00</td>
<td>1792</td>
<td>6.00</td>
<td>1601</td>
<td>5.00</td>
<td>1409</td>
</tr>
<tr>
<td>128</td>
<td>5.00</td>
<td>2560</td>
<td>4.00</td>
<td>2177</td>
<td>3.00</td>
<td>1793</td>
</tr>
<tr>
<td>256</td>
<td>4.00</td>
<td>4096</td>
<td>3.00</td>
<td>3329</td>
<td>2.00</td>
<td>2565</td>
</tr>
<tr>
<td>512</td>
<td>3.00</td>
<td>6144</td>
<td>2.00</td>
<td>4613</td>
<td>1.00</td>
<td>3081</td>
</tr>
</tbody>
</table>

Table 9
Problem 1, $e = 0.5$, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G404</th>
<th></th>
<th>G405</th>
<th></th>
<th>G406</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td></td>
<td>Iter./step</td>
<td>NFE</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>9.38</td>
<td>1200</td>
<td>8.47</td>
<td>1117</td>
<td>8.19</td>
<td>1113</td>
</tr>
<tr>
<td>64</td>
<td>7.08</td>
<td>1812</td>
<td>6.14</td>
<td>1637</td>
<td>5.69</td>
<td>1585</td>
</tr>
<tr>
<td>128</td>
<td>5.61</td>
<td>2872</td>
<td>4.61</td>
<td>2489</td>
<td>3.94</td>
<td>2273</td>
</tr>
<tr>
<td>256</td>
<td>4.40</td>
<td>4508</td>
<td>3.40</td>
<td>3741</td>
<td>2.42</td>
<td>2993</td>
</tr>
<tr>
<td>512</td>
<td>3.55</td>
<td>7268</td>
<td>2.55</td>
<td>5733</td>
<td>1.59</td>
<td>4285</td>
</tr>
</tbody>
</table>

its additional cost per step. In quasi-Newton's method each iteration has two more orders than the previous one and so in this case, the theoretical saving is only 0.5 iterations per step (if the values of the respective error terms are similar). This is a too narrow advantage to be detected numerically unless higher orders are used.

As could be expected, the number of Newton-iterations per step is less or equal than the number of iterations required by functional iteration. However, it is not the purpose of this paper to compare both iterative schemes for which the extra cost of linear algebra of Newton's method should be considered.

We will show now the same type of results for the 4-stage RK-Gauss. The following starting algorithms have been considered:
- **G404**: Starting algorithm of order 4 without additional cost whose coefficients are given in (14).
- **G405**: Starting algorithm of order 5 with one additional function evaluation per step that was obtained in (15).
- **G406**: Starting algorithm of order 6 with two additional function evaluations per step given in (16).

The results of the Tables 8–11 have been obtained over each interval $[h, T + h]$, with $T$ the orbit's period. In this way, the iterations per step are not perturbed by the number of iterations in the first step.

In these tables the superior performance of the starting algorithms of high order can again be appreciated. We remark that the difference between the iterations per step required by G404 and
Table 10
Problem 1, $e = 0.9$, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G4O4</th>
<th>G4O5</th>
<th>G4O6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
<td>NFE</td>
</tr>
<tr>
<td>128</td>
<td>5.97</td>
<td>3056</td>
<td>5.02</td>
</tr>
<tr>
<td>256</td>
<td>4.57</td>
<td>4676</td>
<td>3.59</td>
</tr>
<tr>
<td>512</td>
<td>3.73</td>
<td>7636</td>
<td>2.74</td>
</tr>
<tr>
<td>1024</td>
<td>2.82</td>
<td>11568</td>
<td>2.15</td>
</tr>
</tbody>
</table>

Table 11
Problem 3, functional iteration, one period

<table>
<thead>
<tr>
<th>Steps</th>
<th>G4O4</th>
<th>G4O5</th>
<th>G4O6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iter./step</td>
<td>NFE</td>
<td>Iter./step</td>
<td>NFE</td>
</tr>
<tr>
<td>512</td>
<td>4.74</td>
<td>9708</td>
<td>3.77</td>
</tr>
<tr>
<td>1024</td>
<td>3.78</td>
<td>15492</td>
<td>2.79</td>
</tr>
<tr>
<td>2048</td>
<td>2.94</td>
<td>24064</td>
<td>2.30</td>
</tr>
<tr>
<td>4096</td>
<td>2.38</td>
<td>38992</td>
<td>1.69</td>
</tr>
</tbody>
</table>

G4O5 is greater than $\frac{1}{2}$ and the same happens with G4O5 and G4O6. In consequence, G4O5 is more efficient than G4O4 and G4O6 is more efficient than G4O5.

Moreover, as $h$ decreases to zero that difference is close to 1, depending on the size of the algorithms' error terms for the considered problem too. If this happens for a given $h$, the number of function evaluations required by G4O5 is approximately $3N(=s-1)N$ units less than the required by G4O4, with $N$ the total number of steps, and the same can be said for G4O6 and G4O5. For $s=3$ in a similar situation, that saving was of approximately $2N$ units and, in this sense, the advantages of the starting algorithms of high order are more remarkable with $s = 4$ stages than with $s = 3$.

For an $s$-stage RK formula, a starting algorithm is more efficient than another one that requires one more evaluation per step, if the difference between the number of iterations per step is greater than $1/s$. Furthermore, the order of the studied algorithms is always increased by one for any value of $s$ if we add one function evaluation per step. So it was expected that the number of iterations per step was also reduced in the same way for any value of $s$ (if the sizes of the starting algorithms' error terms are similar). Therefore, the saving on function evaluations grows as $s$ increases because each iteration requires $s$ evaluations per step.

Let us notice that for some representative step sizes, the number of the functions evaluations required by the starting algorithm of the highest order for the 4-stage RK–Gauss formula (G4O6) is lower than the required by the starting algorithm of the highest order for the 3-stage RK–Gauss formula (G3O5). So we obtain in these cases a lower global error, due to the use of an RK formula of higher order, and furthermore a lower computational cost.
Acknowledgements

This paper is based on a part of the author’s doctoral dissertation, written under the supervision of Professors M. Calvo and J.I. Montijano, to whom the author is extremely grateful. The author would also like to thank the anonymous referees for their useful suggestions and comments.

References