Available online at www.sciencedirect.com

ScienceDirect

Applied Numerical Mathematics 59 (2009) 707–722

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

Adaptive nested implicit Runge–Kutta formulas of Gauss type

G.Yu. Kulikov [*], S.K. Shindin

School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa

Available online 21 March 2008

Abstract

This paper deals with a special family of implicit Runge–Kutta formulas of orders 2, 4 and 6. These methods are of Gauss type; i.e., they are based on Gauss quadrature formulas of orders 2, 4 and 6, respectively. However, the methods under discussion have only explicit internal stages that lead to cheap practical implementation. Some of the stage values calculated in a step of the numerical integration are of sufficiently high accuracy that allows for dense output of the same order as the Runge–Kutta formula used. On the other hand, the methods developed are A-stable, stiffly accurate and symmetric. Moreover, they are conjugate to a symplectic method up to order 6 at least. All of these make the new methods attractive for solving nonstiff and stiff ordinary differential equations, including Hamiltonian and reversible problems. For adaptivity, different strategies of error estimation are discussed and examined numerically.

MSC: 65L05; 65L06

Keywords: Ordinary differential equations; Nested implicit Runge–Kutta formulas; Gauss-type methods; Almost symplectic integration; Local error estimation

1. Introduction

When dealing with initial value problems (IVP's) of the form

$$x'(t) = g\big(t, x(t)\big), \quad t \in [t_0, t_{\text{end}}], \qquad x(t_0) = x^0 \tag{1}$$

where $x(t) \in \mathbb{R}^n$ and $g : D \subset \mathbb{R}^{n+1} \to \mathbb{R}^n$ is a sufficiently smooth function, one can apply a one-step numerical scheme as follows:

$$x_{ki} = x_k + \tau_k \sum_{j=1}^{l} a_{ij} g(t_k + c_j \tau_k, x_{kj}), \quad i = 1, 2, \dots, l, \tag{2a}$$

$$x_{k+1} = x_k + \tau_k \sum_{j=1}^{l} b_j g(t_k + c_j \tau_k, x_{kj}), \quad k = 0, 1, \dots, K-1 \tag{2b}$$

[*] Corresponding author.
E-mail addresses: gkulikov@cam.wits.ac.za (G.Yu. Kulikov), sshindin@cam.wits.ac.za (S.K. Shindin).

doi:10.1016/j.apnum.2008.03.019

where $x_0 = x^0$, τ_k is a step size, the values x_{ki} are referred to as stage values of l-stage Runge–Kutta (RK) formula (2). RK method (2) is completely defined by the real coefficients a_{ij}, b_j and c_j, $i, j = 1, 2, \dots, l$, which can be represented conveniently in the form of a partitioned Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

where A is a square matrix of size l, b and c are l-dimensional vectors.

Application of Runge–Kutta schemes is quite advantageous for many reasons. First, these methods can be A-stable and keep high order convergence rate, that is not possible for multistep formulas because of Dahlquist's second barrier [17]. Second, method (2) makes no difficulty for a variable stepsize implementation (see, for example, [9,20,21]). Third, some RK formulas are of great importance when solving Hamiltonian or reversible problems [19,32]. Thus, they are a powerful tool in the area of numerical ordinary differential equations (ODE's).

Unfortunately, RK methods (2) possess a limitation in the sense of long execution time when applied to large-scale IVP's (1). Notice that Implicit Runge–Kutta (IRK) schemes are most suitable for stiff problems. However, IRK methods of high orders are quite time-consuming because of the need to solve, in general, nonlinear systems (2a) of dimension ln in each step of numerical integration (see [9,20,21]).

Significant progress was made in the realm of effective implementation of IRK methods by Bickart [4] and Butcher [8] in the late 70's. Their idea was to transform the matrix A of a Runge–Kutta formula to a diagonal form, which implies successive solutions of l linear systems of dimension n (arising from application of a Newton iteration) rather than one solution of a linear system of dimension ln. On the other hand, that approach has its own drawbacks. For instance, the linear transformation used means that we do not calculate the stage values of method (2) directly, but its linear combinations only. Thus, some extra work is needed to compute the stage values for formula (2a).

At the same time the idea of developing special classes of sufficiently cheap IRK formulas became dominant in numerical analysis of stiff ODE's. The attention was paid to Singly Diagonally Implicit Runge–Kutta (SDIRK) methods (see, for example, [1,2,16,25,29,26] or [18,21] for a review) and to Singly Implicit Runge–Kutta (SIRK) methods (see, for example, [5,6,30], or [18,21] for a review). Both classes of RK formulas admit an efficient implementation directly or through the linear transformation mentioned above. Unfortunately, all these methods also have some limitations.

It is well known that the stage order of SDIRK methods is limited by 1. This could make difficulties for integration of very stiff ODE's, which arise, for instance, from the method of lines applied to some partial differential equations and lead to the order reduction phenomenon for lower stage order RK formulas (see [18,21] for more detail). SIRK methods do not suffer from this limitation. However, their abscissae c_j are determined via zeros of Laguerre polynomials and, hence, can be greater than 1. The latter makes difficulties for integration of nonsmooth and discontinuous problems.

It is necessary to mention here that Butcher and Chen [11] and Butcher and Cash [10] have discovered some SDIRK and SIRK methods with relaxed limitations. The first paper explains how to raise the stage order of SDIRK methods up to 2. The second paper presents a combination of SDIRK and SIRK methods to obtain RK formulas that do not suffer from the above-mentioned difficulties. Nevertheless, the matrix A of any SDIRK or SIRK method must have a single multiple real eigenvalue and that can be quite restrictive. For example, Nørsett and Wolfbrandt [31] prove that the maximum order of such l-stage RK methods is $l + 1$. Thus, other ideas should be tried to overcome the limitations arising from a single multiple real eigenvalue.

We point out that SDIRK and SIRK methods considered above explore the coefficient matrices A of special structures. In the first case it is a lower triangular matrix where the diagonal entries have the same value. So, it is clear that a Newton-type iteration leads to successive solution of linear systems with the same coefficient matrix for all stage values x_{ki} in (2a). The second methods do not use triangular coefficient matrices, but matrices with a multiple real eigenvalue only. This means that they can be transformed to a form admitting the solution of a linear system with the same coefficient matrix for each stage value. Nevertheless, the sole implicit equation in both Runge–Kutta schemes is (2a). Formula (2b) is explicit.

As far as we know, Cash [12] was the first who offered another approach for obtaining cheap RK formulas named Mono-Implicit Runge–Kutta (MIRK) methods. He left the stage values (2a) to satisfy explicit formulas and puts all implicitness into the last equation (2b). Cash and Singhal gave samples of A- and L-stable MIRK methods up to order 4 and considered carefully their implementation in [13,14] and [15]. A numerical comparison with other efficient numerical schemes was also presented there. Later, van Bokhoven [33] introduced the same RK formulas and termed them Implicit Endpoint Quadrature (IEQ) formulas. He constructed A-stable examples of such methods up to order 6. We stress that the dimension of linear systems arising in implementation of a Newton-type iteration is the same as the dimension of the source problem (1). Other properties of these methods were studied in a number of subsequent publications (see, for instance, [7] and [28]).

Kulikov and Shindin [23] explore the same idea to construct cheap RK methods of classical order 4 and of stage order 2 or 3. Moreover, their methods are A-stable, stiffly accurate and symmetric. We show below that they are also conjugate to a symplectic method up to order 6 at least. So, the methods of Kulikov and Shindin are potentially useful in practice.

The negative feature of all high order MIRK (or IEQ) formulas is that the Jacobi matrices and, hence, linear systems arising in such formulas are of a quite complicated structure. More precisely, they are matrix-value polynomials of degree less than or equal to the number of stages inserted in the final formula (2b) plus one (or the number of nested levels in this paper). Full evaluation of such a polynomial can be extremely time-consuming, especially for large-scale problems. That is why a simplified Newton iteration is crucial for an effective implementation.

Cash and Singhal [15] propose to construct MIRK methods with Jacobian that can be either factorized exactly as $(I - \nu\tau_k J(t_k, x_k))^{r+1}$, where I is the identity matrix, ν is a scalar, r is some fixed positive integer and $J(t_k, x_k)$ denotes the Jacobian of the right-hand side of ODE (1) evaluated at the point (t_k, x_k), or approximated as a power of a suitable matrix. The first condition is quite restrictive for high order MIRK methods. That is why they presented L-stable embedded MIRK formulas of orders 3 and 4 with suitable approximations of the Jacobian. A limitation of the methods listed in [15] is the fact that the authors did not address the issue of how their approximations influence the rate of convergence of the modified Newton iteration considered in that paper (a theoretical tool to do that appears in [22]). Most importantly, they did not recognize that their approximations can ruin the stability of the designed methods. To see that, let us consider the following order 3 MIRK method presented in [15]:

$$x_{k1}^2 = \frac{1}{4}x_k + \frac{3}{4}x_{k+1} - \frac{\tau_k}{4}g(t_{k+1}, x_{k+1}), \tag{3a}$$

$$x_{k+1} = x_k + \frac{\tau_k}{6}\bigl(g(t_k, x_k) + 4g\bigl(t_{k1}^2, x_{k1}^2\bigr) + g(t_{k+1}, x_{k+1})\bigr) \tag{3b}$$

where $t_{k1}^2 = t_k + 0.5\tau_k$. Method (3) is L-stable. Indeed, if we apply this scheme to the Dahlquist test equation $x' = \lambda x$ where λ is a fixed complex number we will obtain the following stability function:

$$R(z) = \frac{1 + 1/3z}{1 - 2/3z + 1/24z^2} \tag{4}$$

where $z = \tau_k\lambda$.

To solve nonlinear equation (3b) for x_{k+1}, Cash and Singhal [15] recommend to apply the modified Newton iteration with Jacobian $(I - \nu\tau_k J(t_k, x_k))^2$ where $\nu = 0.38534019921340$. Earlier, Cash [14] discussed using one Newton-type iteration step with the trivial predictor, i.e. starting with the initial guess x_k, to implement his MIRK methods. If we consider such an implementation for MIRK method (3) and the above-mentioned iteration we will obtain the numerical solution with the stability function

$$\tilde{R}(z) = \frac{1 + (1 - 2\nu)z + (\nu^2 - 1/6)z^2}{1 - 2\nu z + \nu^2 z^2}, \tag{5}$$

which is no longer L-stable. Notice that if we apply the same modified Newton iteration but with exact Jacobian $I - 2/3\tau_k J(t_k, x_k) + 1/6\tau_k^2 J^2(t_k, x_k)$ the stability function will be given by formula (4) again. It is quite clear since the Newton iteration with exact Jacobian solves precisely any linear problem for one iteration step. So, it inherits all linear stability properties of the underlying RK formula. However, if we change the Jacobian this might no longer be the case.

Thus, we see from (4) and (5) and the above analysis that the approximation of the exact Jacobian of a MIRK (or IEQ) method must be chosen carefully in terms of both accuracy and stability. Kulikov and Shindin present such an approximation for their RK methods in [23]. Its influence on the convergence rate of the modified Newton iteration is studied theoretically and numerically in [22]. It is also important to mention here that the number r appearing in the power of the approximate Jacobian above implies the number of additional successive solutions of linear systems to perform one step of the Newton iteration. Thus, we are interested to keep this number to a minimum. For example, all order 4 MIRK formulas in [15] use approximations where $r = 2$. The methods of order 4 presented in [23] implement the Jacobian approximation when $r = 1$. In other words, they required less work but have a broader set of useful properties.

Unfortunately, the paper of van Bokhoven [33] contains some mistakes, as explained in Sections 3 and 5. Moreover, he did not propose any satisfactory implementation for his IEQ formulas. Cash and Singhal criticized van Bokhoven's approach in [15]. Therefore, the methods presented in [33] are unlikely to be efficient in practice. In addition, all computational schemes designed by the above-cited authors are intended only to possess specific stability properties and do not cover numerical methods suitable for Hamiltonian and reversible problems.

In this paper, we present cheap RK formulas which have almost the same properties as Gauss methods (or even superior to the latter formulas in some sense). More precisely, we extend the approach by Kulikov and Shindin [23] to Nested Implicit Runge–Kutta (NIKR) schemes. All of them possess the property of explicit internal stages in the sense that they are easily reduced to a single nonlinear equation with respect to the numerical solution x_{k+1} at the new grid point t_{k+1}. The most important consequence is that the dimension of the nonlinear equation to be solved is the same as the dimension of the source problem (1). It means that they admit an efficient implementation in practice. Here, we explain how to construct a NIRK method of order 6 with use of the Gauss quadrature formula of order 6. We also present necessary and sufficient conditions for a lower order RK formula to be conjugate to a symplectic method up to order 6. For instance, this theoretical result shows that the RK methods discovered in [23] are conjugate to a symplectic method up to order 6 at least. In addition, our Gauss-type NIRK methods of orders 2, 4 and 6 are A-stable, stiffly accurate, symmetric and embedded. The latter property is important for cheap local error estimation. We also have to mention that the methods discussed in this paper allow naturally for dense output of the same order as the NIRK method applied. Thus, these methods can treat effectively many practical IVP's of the form (1). In addition, we design and examine different stepsize selection policies grounded in local error control. We try both Richardson extrapolation and Gauss-type embedded NIRK formulas on one numerical example with the purpose of finding an effective step-changing strategy.

2. Nested implicit Runge–Kutta methods

One step of size τ_k of an l-level NIRK method applied to ODE (1) is presented by the formulas

$$x_{kj}^2 = a_{j1}^2 x_k + a_{j2}^2 x_{k+1} + \tau_k\big(d_{j1}^2 g(t_k, x_k) + d_{j2}^2 g(t_{k+1}, x_{k+1})\big), \quad j = 1, 2, \tag{6a}$$

$$\begin{aligned} x_{kj}^i &= a_{j1}^i x_k + a_{j2}^i x_{k+1} + \tau_k\big(d_{j1}^i g(t_k, x_k) + d_{j2}^i g(t_{k+1}, x_{k+1})\big) \\ &\quad + \tau_k \sum_{m=1}^{i-1} d_{j,m+2}^i g\big(t_{km}^{i-1}, x_{km}^{i-1}\big), \quad i = 3, 4, \ldots, l,\ j = 1, 2, \ldots, i, \end{aligned} \tag{6b}$$

$$x_{k+1} = x_k + \tau_k \sum_{i=1}^{l} b_i g\big(t_{ki}^l, x_{ki}^l\big) \tag{6c}$$

where $x_0 = x^0$, $t_{kj}^i = t_k + \tau_k c_j^i$. The definition of NIRK methods implies that $l \geqslant 2$ and 1-level NIRK formulas do exist. In this case, formulas (6a) and (6b) are absent and such methods are defined by the single formula (6c). The conventional Trapezoidal Rule is an example of a 1-level NIRK method.

To motivate methods (6) we present the following idea: It is clear that any RK method (2) is grounded in some quadrature formula. More precisely, the nodes and weights of this formula determine the coefficients of the step update formula (2b). The remaining formulas (2a) are used to approximate values of the exact solution at a number of fixed points in the interval $[t_k, t_{k+1}]$ for the underlying quadrature formula. Certainly, the accuracy of the approximation influences the order of the RK method, which does not exceed the order of the underlying quadrature formula. By the way, all NIRK methods discussed in this paper are based on Gauss quadrature formulas. That is why we call them Gauss-type NIRK methods. So, the principal idea of our methods is to approximate the exact solution at the fixed Gauss nodes by a Hermite interpolating polynomial. If we take into account the numerical solution and its first derivative evaluated at the grid points t_k and t_{k+1} we will be able to fit the Hermite polynomial of degree 3 at most. It is sufficient to approximate correctly the exact solution up to order 4. Therefore we can construct Gauss-type NIRK methods up to order 4 by using the information from the endpoints t_k and t_{k+1} only (see [23]). However, it is not sufficient for an RK method of order higher than 4. For example, if we take the Gauss quadrature formula of order 6 we will need a Hermite interpolating polynomial of degree 5 to approximate the exact solution at the nodes of this formula with sufficiently high accuracy. Thus, we use first the Hermite interpolating polynomial of degree 3 to approximate the exact solution at two extra point (the stage values of the second level). We remark that the numerical solutions x_k and x_{k+1} computed at the grid points t_k and t_{k+1} are considered to be stage values of the first level. Then, we fit the Hermite polynomial of degree 5 to all available information in order to calculate approximations of the exact solution for the step update formula (6c) (the stage values of the third level) and so on. A regular way of implementing the above-mentioned scheme is discussed in the next section.

We stress that we use only the more accurate stage values x_{kj}^{i-1} (and do not use lower level stage values x_{kj}^{i-m}, $m = 2, 3, \dots, i-2$, which are also available) to calculate stage values of the next level in the NIRK methods. This allows the accuracy of approximation to be raise to the necessary order. Notice that the interpolating polynomial used to calculate stage values of the last level can be also used for dense output of the same order as the step update formula (6c). Our motivation implies that the matrix A of any high order NIRK method (6) has a block diagonal part, as shown below.

We deal here with methods (6) satisfying the condition

$$a_{j1}^i + a_{j2}^i = 1, \quad i = 2, 3, \dots, l, \; j = 1, 2, \dots, i. \tag{7a}$$

It also follows from the above explanation that the stage values x_{kj}^i approximate the exact solution of problem (1) at the points t_{kj}^i where

$$c_j^i = a_{j2}^i + \sum_{m=1}^{i+1} d_{jm}^i. \tag{7b}$$

First of all we want to show that any method (6) of order $s \geqslant 1$ and satisfying conditions (7) can be represented in the form of an implicit RK formula (2). Notice that the required first order is used to define the coefficients c_j^i by formula (7b). To reach our goal, let us introduce the notation

$$D^i = \begin{pmatrix} d_{13}^i & \cdots & d_{1,i+1}^i \\ \vdots & \ddots & \vdots \\ d_{i3}^i & \cdots & d_{i,i+1}^i \end{pmatrix}, \quad d_1^i = \begin{pmatrix} d_{11}^i \\ \vdots \\ d_{i1}^i \end{pmatrix}, \quad d_2^i = \begin{pmatrix} d_{12}^i \\ \vdots \\ d_{i2}^i \end{pmatrix},$$
$$a^i = \begin{pmatrix} a_{12}^i \\ \vdots \\ a_{i2}^i \end{pmatrix}, \quad c^i = \begin{pmatrix} c_1^i \\ \vdots \\ c_i^i \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_l \end{pmatrix}.$$

Then method (6) is converted to an RK formula with the Butcher tableau

$$\begin{array}{c|ccccccc}
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
c^2 & d_1^2 & 0 & 0 & \cdots & 0 & a^2 b^T & d_2^2 \\
c^3 & d_1^3 & D^3 & 0 & \cdots & 0 & a^3 b^T & d_2^3 \\
c^4 & d_1^4 & 0 & D^4 & \cdots & 0 & a^4 b^T & d_2^4 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
c^l & d_1^l & 0 & 0 & \cdots & D^l & a^l b^T & d_2^l \\
1 & 0 & 0 & 0 & \cdots & 0 & b^T & 0 \\
\hline
 & 0 & 0 & 0 & \cdots & 0 & b^T & 0
\end{array} \tag{8}$$

We conclude with the following:

(A) Methods (6) are a special case of IRK formulas (2).
(B) They can be rewritten in a form where all stages (6a), (6b) have been inserted into the last formula (6c).

Conclusions (A) and (B) are the reason to term methods (6) as Nested Implicit Runge–Kutta formulas. We remark that these methods are similar in some sense to the nested scheme for evaluating polynomials.

We emphasize that, in contrast to SDIRK and SIRK methods, the only implicit formula in any NIRK method is (6c). Thus, the dimension of a nonlinear problem arising in implementation of method (6) does not depend on the number of internal stages and always coincides with the dimension of source problem (1). This makes NIRK methods potentially attractive for solving stiff ODE's. The Butcher tableau (8) shows that the stability function of method (6) is not limited by the rational approximation to the exponential function with a single multiple real pole and, hence, we have more freedom to construct efficient methods possessing some extra properties such as high order, stability, symmetry and so on.

Our NIRK methods are a particular case of MIRK (or IEQ) formulas introduced earlier. However, Cash and Singhal [12–15] did not consider Gauss-type MIRK formulas at all, and van Bokhoven [33] failed to determine correctly IEQ methods corresponding to Gauss quadrature formulas of order 4 or higher and in an optimal way. So, we intend to present NIRK methods based on the mentioned quadrature formulas of orders 2, 4 and 6. We will also show that the methods derived in the next section are suitable for solving stiff, Hamiltonian and reversible problems, including differential-algebraic systems.

3. Gauss-type NIRK methods of orders 2, 4 and 6

First of all we notice that the Gauss-type NIRK method of order 2 is the conventional Implicit Mid-Point Rule. To see that, we write the Implicit Mid-Point Rule in the form of a NIRK method, as follows:

$$x_{k1}^2 = \frac{1}{2}x_k + \frac{1}{2}x_{k+1}, \tag{9a}$$

$$x_{k+1} = x_k + \tau_k g\bigl(t_{k1}^2, x_{k1}^2\bigr) \tag{9b}$$

where $t_{k1}^2 = t_k + 0.5\tau_k$. It is clear that method (9) is a particular case of NIRK method (6). Conditions (7) also hold. Evidently, method (9) possesses all the above-mentioned properties. So, we now continue with NIRK methods based on the Gauss quadrature formula of order 4.

We do not repeat the analysis done in [23] and merely present the final result in the form of a one-step method as follows:

$$x_{k1}^2 = a_{11}^2 x_k + a_{12}^2 x_{k+1} + \tau_k\bigl(d_{11}^2 g(t_k, x_k) + d_{12}^2 g(t_{k+1}, x_{k+1})\bigr), \tag{10a}$$

$$x_{k2}^2 = a_{21}^2 x_k + a_{22}^2 x_{k+1} + \tau_k\bigl(d_{21}^2 g(t_k, x_k) + d_{22}^2 g(t_{k+1}, x_{k+1})\bigr), \tag{10b}$$

$$x_{k+1} = x_k + \tau_k\bigl(b_1 g\bigl(t_k + c_1^2\tau_k, x_{k1}^2\bigr) + b_2 g\bigl(t_k + c_2^2\tau_k, x_{k2}^2\bigr)\bigr) \tag{10c}$$

where the coefficients b_i, c_i^2 are determined uniquely as the nodes and weights of the Gauss quadrature formula; i.e., they are: $b_1 = b_2 = 1/2$, $c_1^2 = (3-\sqrt{3})/6$, $c_2^2 = (3+\sqrt{3})/6$. The remaining coefficients a_{ij}^2, d_{ij}^2, $i, j = 1, 2$, are found to ensure order 4 for NIRK method (10). It has been shown in [23] that these coefficients must be:

$$a_{11}^2 = \theta, \quad a_{12}^2 = 1-\theta, \quad a_{21}^2 = 1-\theta, \quad a_{22}^2 = \theta, \tag{11a}$$

$$d_{11}^2 = \frac{6\theta - 2 - \sqrt{3}}{12}, \qquad d_{12}^2 = \frac{6\theta - 4 - \sqrt{3}}{12}, \tag{11b}$$

$$d_{21}^2 = \frac{4 + \sqrt{3} - 6\theta}{12}, \qquad d_{22}^2 = \frac{2 + \sqrt{3} - 6\theta}{12} \tag{11c}$$

where θ is a free real parameter.

It is not difficult to represent all methods (10) with coefficients (11) as an RK scheme with Butcher tableau of the form (8), as follows:

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
c_1^2 & \dfrac{6(c_1^2+\theta)-5}{12} & \dfrac{1-\theta}{2} & \dfrac{1-\theta}{2} & \dfrac{6(c_1^2+\theta)-7}{12} \\
1-c_1^2 & \dfrac{7-6(c_1^2+\theta)}{12} & \dfrac{\theta}{2} & \dfrac{\theta}{2} & \dfrac{5-6(c_1^2+\theta)}{12} \\
1 & 0 & \dfrac{1}{2} & \dfrac{1}{2} & 0 \\
\hline
 & 0 & \dfrac{1}{2} & \dfrac{1}{2} & 0
\end{array}
$$

where c_1^2 has been defined above.

Then, it is easy to check that NIRK methods (10), (11) are of stage order 2, A-stable, stiffly accurate and symmetric. Moreover, a special choice of θ (i.e., when $\theta = 1/2 + 2\sqrt{3}/9$) gives us the method of stage order 3. In other words, NIRK methods (10), (11) satisfy simplifying assumptions $\mathcal{B}(4)$ and $\mathcal{C}(2)$ for any θ. The choice of $\theta = 1/2 + 2\sqrt{3}/9$ results in even $\mathcal{C}(3)$. We refer the reader to [9,18] or [21] for more information concerning simplifying assumptions.

We remark that the sole IEQ formula corresponding to the Gauss quadrature of order 4 and discovered by van Bokhoven [33] is Method IV. The author claims that his Method IV is of order 3 and satisfies simplifying assumptions $\mathcal{B}(3)$ and $\mathcal{C}(1)$. On the other hand, the same method is obtained by substitution of $\theta = (2+\sqrt{3})/6$ into formulas (10) and (11). Therefore we conclude that Method IV of [33] is of order 4 and satisfies simplifying assumptions $\mathcal{B}(4)$ and $\mathcal{C}(2)$.

We recall that a Taylor expansion of the defect of method (10) was used directly to determine coefficients (11) in [23]. However, it becomes much more complicated when implemented in NIRK methods of higher order. Thus, we are constrained to application of the conventional theory of order conditions for RK methods (2). This theory is presented in detail in [9] or [20].

To construct Gauss-type NIRK methods of order 6 we start with an RK scheme having a Butcher tableau of the form

$$
\begin{array}{c|ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & a_{21} & 0 & 0 & a_{24} & a_{25} & a_{26} & a_{27} \\
c_3 & a_{31} & 0 & 0 & a_{34} & a_{35} & a_{36} & a_{37} \\
c_4 & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\
c_5 & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} \\
c_6 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} \\
1 & 0 & 0 & 0 & b_4 & b_5 & b_6 & 0 \\
\hline
 & 0 & 0 & 0 & b_4 & b_5 & b_6 & 0
\end{array}
\tag{12}
$$

We want the coefficients in tableau (12) to satisfy the following conditions:

I. The coefficients c_2 and c_3 are zeros of the second-degree Legendre polynomial

$$L_2(t) = \frac{d^2}{dt^2}\bigl(t^2(1-t)^2\bigr).$$

II. The coefficients c_4, c_5 and c_6 are zeros of the third-degree Legendre polynomial

$$L_3(t) = \frac{d^3}{dt^3}\bigl(t^3(1-t)^3\bigr).$$

III. The coefficients b_4, b_5 and b_6 are weights of the Gauss quadrature formula of order 6.

It is easy to see that the last two conditions guarantee $\mathcal{B}(6)$. Condition I is used to embed method (10), (11) into the new one.

Notice that the form of the Butcher tableau (12) follows from the form of general tableau (8). It also results in the conclusion that

$$a_{ij} = a_{i2}^2 b_j, \quad i = 2, 3, \ j = 4, 5, 6, \tag{13a}$$

$$a_{ij} = a_{i2}^3 b_j, \quad i = 4, 5, 6, \ j = 4, 5, 6. \tag{13b}$$

The remaining free coefficients in rows 2 and 3 of (12) are determined from simplifying assumption $\mathcal{C}(3)$ and the remaining free coefficients in rows 4, 5 and 6 are found from simplifying assumption $\mathcal{C}(5)$. This means that we have to solve the following linear systems:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1/2 & 1 \\ 0 & 1/3 & 1 \end{pmatrix} \begin{pmatrix} a_{i1} \\ a_{i2}^2 \\ a_{i7} \end{pmatrix} = \begin{pmatrix} c_i \\ c_i^2/2 \\ c_i^3/3 \end{pmatrix}, \quad i = 2, 3; \tag{14}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & c_2 & c_3 & 1/2 & 1 \\ 0 & c_2^2 & c_3^2 & 1/3 & 1 \\ 0 & c_2^3 & c_3^3 & 1/4 & 1 \\ 0 & c_2^4 & c_3^4 & 1/5 & 1 \end{pmatrix} \begin{pmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i2}^3 \\ a_{i7} \end{pmatrix} = \begin{pmatrix} c_i \\ c_i^2/2 \\ c_i^3/3 \\ c_i^4/4 \\ c_i^5/5 \end{pmatrix}, \quad i = 4, 5, 6. \tag{15}$$

We recall that the coefficients c_i are fixed by conditions I and II. These conditions say also that the coefficient matrices of linear systems (14) and (15) are nonsingular and, hence, the nontrivial coefficients a_{ij} are determined uniquely:

$$a_{21} = \frac{c_3}{6}, \quad a_{24} = a_{26} = \frac{35 - 40c_3}{108}, \quad a_{25} = \frac{56 - 64c_3}{108}, \quad a_{27} = \frac{c_3 - 1}{6},$$
$$a_{31} = \frac{c_2}{6}, \quad a_{34} = a_{36} = \frac{35 - 40c_2}{108}, \quad a_{35} = \frac{56 - 64c_2}{108}, \quad a_{37} = \frac{c_2 - 1}{6},$$
$$a_{41} = -a_{67} = \frac{20c_6 - 3}{200}, \quad a_{42} = \frac{36c_6 + 18c_3 - 27}{200}, \quad a_{43} = \frac{36c_6 - 18c_3 - 9}{200},$$
$$a_{44} = a_{46} = \frac{39c_4 - 7}{90}, \quad a_{45} = \frac{780c_4 - 140}{1125}, \quad a_{47} = -a_{61} = \frac{3 - 20c_4}{200}, \quad a_{55} = \frac{2}{9},$$
$$a_{51} = -a_{57} = \frac{1}{32}, \quad a_{52} = -a_{53} = \frac{\sqrt{27}}{32}, \quad a_{54} = a_{56} = \frac{5}{36}, \quad a_{65} = \frac{780c_6 - 140}{1125},$$
$$a_{62} = \frac{36c_4 + 18c_3 - 27}{200}, \quad a_{63} = \frac{36c_4 - 18c_3 - 9}{200}, \quad a_{64} = a_{66} = \frac{39c_6 - 7}{90}$$

where

$$c_2 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \quad c_3 = \frac{1}{2} + \frac{\sqrt{3}}{6}, \quad c_4 = \frac{1}{2} - \frac{\sqrt{15}}{10}, \quad c_5 = \frac{1}{2} \quad \text{and} \quad c_6 = \frac{1}{2} + \frac{\sqrt{15}}{10}.$$

Similar to the well-know simplifying assumption $\mathcal{C}(3)$, which is defined for all stage values, we introduce an analogous simplifying assumption $\mathcal{C}(3,5)$ that is defined stage-wise. Our simplifying assumption $\mathcal{C}(3,5)$ means merely that conditions (14) and (15) hold for the stage values of the second and third levels, respectively. In other words, $\mathcal{C}(3)$ is satisfied for the second level stage values and $\mathcal{C}(5)$ holds for the third level stage values. The new simplifying assumption shows how we raise the accuracy of stage values in the NIRK method under discussion.

Thus, we have built the unique NIRK method with tableau (12) that satisfies conditions I–III and simplifying assumption $\mathcal{C}(3,5)$. We prove now that the constructed NIRK method is of classical order 6.

Theorem 1. *The Gauss-type NIRK method with Butcher tableau* (12) *that satisfies conditions* I–III *and simplifying assumption* $\mathcal{C}(3,5)$ *is of classical order* 6 *and stage order* 3.

Proof. The statement of Theorem 1 concerning the stage order follows from the simplifying assumption $\mathcal{C}(3,5)$ at once. Let us prove further that the method is of classical order 6.

First of all we introduce the set $\mathcal{T}$ of rooted trees. The order of a tree (denoted by $|t|$) is the number of vertices in the tree t. The set of rooted trees of order up to 6 is denoted by $\mathcal{T}_6$. We refer the reader to [9] or [20] for the detailed theory of rooted trees. To complete the proof, we have to check the order conditions of RK methods for all rooted trees in $\mathcal{T}_6$ (see Theorem 315A in [9, p. 145] or Theorem 2.13 in [20, p. 153]).

For the sake of simplicity, we introduce two sets of indexes:

$$I_3 = \{2, 3\}, \qquad I_5 = \{1, 4, 5, 6, 7\}.$$

Then, it is clear from simplifying assumptions $\mathcal{C}(3)$ when $i \in I_3$ and $\mathcal{C}(5)$ when $i \in I_5$ that all rooted trees of orders less than or equal to 5 are reduced to the quadrature formula order conditions (corresponding to bush trees), which evidently hold because of $\mathcal{B}(6)$. Here, we have used also that $b_i = 0$ when $i \in I_3$. Thus, it is left to check the trees of order 6 only.

It is not difficult to see that any rooted tree of order 6 is reduced by $\mathcal{C}(3)$ when $i \in I_3$, $\mathcal{C}(5)$ when $i \in I_5$ and the fact that $b_i = 0$ when $i \in I_3$ to the following two trees: $t^* = [\bullet, \bullet, \bullet, \bullet, \bullet]$ and $t^{**} = [[[\bullet, \bullet, \bullet]]]$. It is clear that the first tree t^* corresponds to the quadrature condition $\sum b_i c_i^5 = 1/6$, which is satisfied because of $\mathcal{B}(6)$. Thus, it is left to check the order condition for the tree t^{**}.

Again, with use of the facts mentioned above, we obtain the following order condition for the tree t^{**}:

$$\frac{1}{120} = \sum_{i,j,k} b_i a_{ij} a_{jk} c_k^3 = \frac{1}{5 \cdot 4} \sum_i b_i c_i^5 + \sum_i b_i \sum_{j \in I_3} a_{ij} \sum_k \left(a_{jk} c_k^3 - \frac{c_j^4}{4} \right). \tag{16}$$

Now we prove that the last summand in (16) is equal to zero and, hence, the last order condition holds. This guarantees that the Gauss-type NIRK method under discussion is of order 6.

For the proof, we remark that, because of simplifying assumption $\mathcal{C}(3)$, the stage values for $i \in I_3$ can be interpreted as approximations to a sufficiently smooth function f at the nodes c_i, $i \in I_3$, derived by means of the Hermite interpolating polynomial h fitted to the conditions:

$$h(0) = f(0), \quad h(1) = f(1), \quad h'(0) = f'(0), \quad h'(1) = f'(1).$$

For the special choice of the function f; i.e., when $f = t^4/4$, the last sum on the right-hand side of formula (16) represents precisely the error of the above-mentioned interpolation. We refer the reader, for instance, to [3] for the interpolation error of Hermite interpolating polynomials. Hence, this interpretation yields

$$\sum a_{jk} c_k^3 - \frac{c_j^4}{4} = h(c_j) - \frac{c_j^4}{4} = -\frac{c_j^2(1 - c_j)^2}{4}, \quad j \in I_3.$$

Next, it is well known that zeros of the Legendre polynomial of degree s satisfy the condition $c_i = 1 - c_{s-i+1}$, $i = 1, 2, \dots, s$. Therefore we transform the last summand on the right-hand side of formula (16) to the form

$$-\frac{c_2^2 c_3^2}{4} \sum_i b_i \sum_{j \in I_3} a_{ij}. \tag{17}$$

The term (17) can be easily evaluated. Indeed, let us consider the polynomial $p(t) = t^3(1 - t)^3$. Since $\mathcal{B}(6)$ holds for the Gauss quadrature formula of order 6 and deg $p(t) = 6$ then

$$\sum b_i p'(c_i) = p(1) = 0, \tag{18}$$

because this quadrature formula is exact for any polynomial of degree $\leqslant 6$. On the other hand, the following formula is also valid: $\sum a_{ij} p''(c_j) = p'(c_i)$. It is a consequence of simplifying assumption $\mathcal{C}(5)$ when $i \in I_5$. Finally, the symmetry of the polynomial $p(t)$ (i.e., $p(t) = p(1 - t)$) results in

$$p'(c_i) = p''(c_2) \sum_{j \in I_3} a_{ij}, \quad i \in I_5. \tag{19}$$

The substitution of (19) into formula (18) leads to

$$p''(c_2) \sum_i b_i \sum_{j \in I_3} a_{ij} = 0. \tag{20}$$

It is obvious that $p''(c_2) \neq 0$. Otherwise, if c_2 and c_3 were roots of the polynomial $p''(t)$ it would be of the form

$$p''(t) = Rt(1-t)L_2(t)$$

where $R \neq 0$ is a constant. However, the latter is impossible because

$$\int_0^1 p''(t)\,dt = 0,$$

$$R\int_0^1 t(1-t)L_2(t)\,dt = -2R\int_0^1 t(1-t)L_1^2(t)\,dt$$

and the last integral is nonzero provided that $R \neq 0$. This follows from the fact that $t(1-t)L_1^2(t)$ is a nonnegative function in the interval [0, 1] and is not zero identically.

Thus, we have proved that the last summand in (16) equals zero and, hence, the Gauss-type NIRK method under consideration is of classical order 6. The theorem is proved. □

We complete our analysis of Gauss-type NIRK methods with the following remarks:

Remark 1. It is not difficult to check that the order 6 Gauss-type NIRK method is symmetric. For that, it is sufficient to rearrange the nodes c_i in growing order as well as the other coefficients of tableau (12). Then, the proof is straightforward.

This fact allows Theorem 1 to be proved in a trivial way. Indeed, it is easy to prove that the order of the Gauss-type NIRK method with Butcher tableau (12) satisfying conditions I–III and simplifying assumption $\mathcal{C}(3,5)$ is not less than 5. On the other hand, the order of any symmetric RK formula is an even integer. Therefore the order of the constructed NIRK method is 6 because $\mathcal{B}(7)$ does not evidently hold.

Remark 2. Gauss-type NIRK methods of orders 4 and 6 discussed above are A-stable and their stability functions are the Padé approximations $R_{22}(z)$ and $R_{33}(z)$, respectively.

Remark 3. It follows from the construction of Gauss-type NIRK methods of orders 4 and 6 that a part of the coefficients in their Butcher tableaux coincide. Therefore they form a pair of embedded RK formulas that can be used for cheap local error estimation.

Remark 4. It is also quite clear from the construction of Gauss-type NIRK methods of orders 4 and 6 that stage values of the last levels and the numerical solution computed in each step can be used for dense output of the same order as the method applied (via polynomial interpolation).

Finally, we stress that all NIRK methods are stiffly accurate. This follows from the Butcher tableau (8). Thus, the methods presented here can be advantageous to integrate numerically very stiff ODE's as well as differential-algebraic systems (see, for example, [21]).

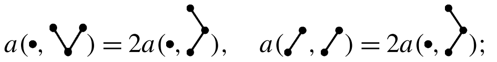
4. Hamiltonian problems

We have seen above that all Gauss-type NIRK methods presented in Section 3 are symmetric and, hence, can be used to integrate reversible problems. Below, we show that they are also suitable for integration of Hamiltonian systems in the sense that they are conjugate to a symplectic method up to order 6 at least. For that, we prove one lemma that extends the result of Leone [27].

We start with some additional notation: Let $B(a,x)$ be a B-series. We introduce also coefficients $a(t_1,t_2) = a(t_1)a(t_2) - a(t_1 \circ t_2) - a(t_2 \circ t_1)$ and $\delta(t) = \gamma^{-1}(t) - a(t)$ where $\circ$ denotes the Butcher product of trees. It is defined by the formula $[t_{1,1},\dots,t_{1,k}] \circ t_2 = [t_{1,1},\dots,t_{1,k},t_2]$. Then, we have

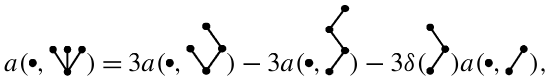
Lemma 2. *Consider a one-step method $\Phi_\tau(x) = B(a, x)$ of order $s \geqslant 2$. Then, it is conjugate to a symplectic method*

(i) *up to order* 4 *if and only if*

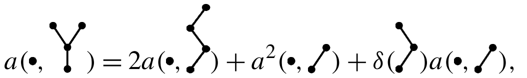


(21)

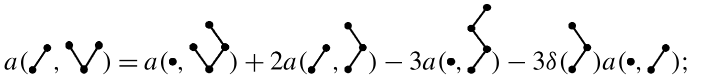
(ii) *up to order* 5 *if and only if conditions* (21) *hold and, additionally,*



(22a)

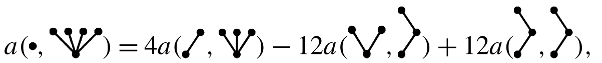


(22b)

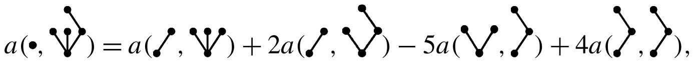


(22c)

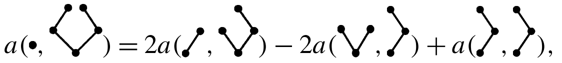
(iii) *up to order* 6 *if and only if conditions* (21), (22) *hold and, additionally,*



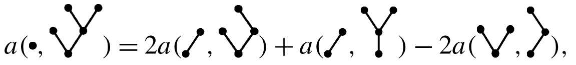
(23a)



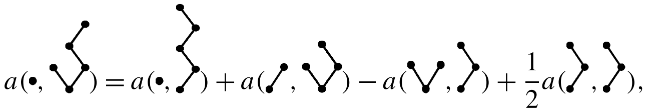
(23b)



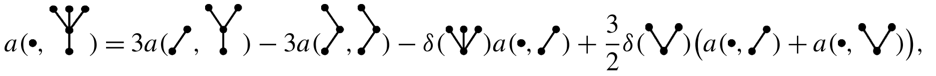
(23c)



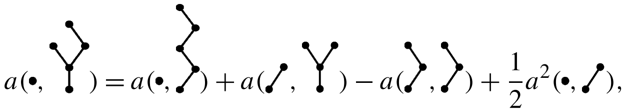
(23d)



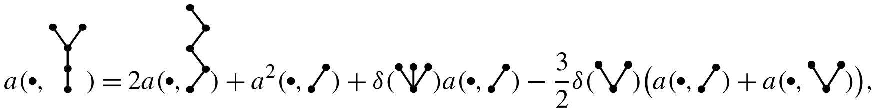
(23e)



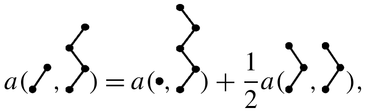
(23f)



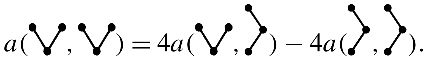
(23g)



(23h)



(23i)



(23j)

Proof. Statement (i) is proved by Leone (see [27]). Statement (ii) extends his result to a one-step method $\Phi_\tau(x)$ of order $s < 4$.

To prove statement (iii), one can exploit directly the approach presented in [27]. Namely, let a series $B(a, x)$ be conjugate to a symplectic method $B(b, x)$ up to order 6. This means that there exists a change of variable $B(c, x)$ which satisfies the formula $ac(t) \equiv cb(t)$ for all trees with the property $|t| \leqslant 6$. By expressing $b(t)$ for $|t| \leqslant 6$ in terms of the coefficients $a(t)$ and $c(t)$ and, then, substituting these expressions into the formula $b(t_1, t_2) = 0$ for all trees such that $|t_1| + |t_2| = 6$, we arrive at a system of 16 equations. Further analysis shows that 6 of them are satisfied by an appropriate choice of the coefficients $c(t)$ when $|t| = 5$. Finally, conditions (21) and (22) transform the remaining 10 equations to the form of formulas (23). □

At the end of Section 4, we stress that the Gauss-type NIRK method of order 2 is symplectic (see, for example, [19,32]). The direct substitution of coefficients of the Gauss-type NIRK methods of order 4 into formulas (22) and (23) proves that they are conjugate to a symplectic method up to order 6 at least. Notice that condition (21) holds because of the order of the methods. Therefore our NIRK methods of order 4 are effective to integrate Hamiltonian problems as well. This conclusion is confirmed numerically in [24]. Unfortunately, Lemma 2 is useless for RK methods of order 6 or higher. All conditions (21), (22) and (23) are obviously satisfied. This follows from the order conditions. However, the lemma gives us no information on the behaviour of the error of such methods.

5. Error estimation

In modern computational practice, it is accepted to use embedded RK formulas for the local error estimation and stepsize selection (see, for example, [20,21]). The reason is that such methods give the cheapest way of local error control. Thus, it is important to try that idea on the NIRK methods developed above. We recall that the embedded NIRK methods arise naturally from their construction (see Section 3).

First of all we refer again to paper [33] for an embedded family of IEQ formulas of orders from 1 to 6 (see formula (35) in the cited paper). Here, we represent those methods in the usual RK form

$$\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 11/72 & 1/18 & 1/48 & 1/18 & -5/144 \\
1/2 & -61/360 & 38/45 & 1/15 & -22/45 & 89/360 \\
3/4 & 9/80 & 3/10 & 9/80 & 3/10 & -3/40 \\
1 & 7/90 & 16/45 & 2/15 & 16/45 & 7/90 \\
\hline
 & 7/90 & 16/45 & 2/15 & 16/45 & 7/90 \\
\hline
(a) & 0 & 0 & 0 & 0 & 1 \\
(b) & 1/2 & 0 & 0 & 0 & 1/2 \\
(c) & 5/18 & 0 & 0 & 8/9 & -1/6 \\
(d) & 1/18 & 4/9 & 0 & 4/9 & 1/18
\end{array} \tag{24}$$

Van Bokhoven [33] writes that "... (24) represents a true variable order method with error estimation. As to the stability of these lower order methods, it is easily verified that all methods are strictly A-stable, except for the first and third order methods which are strongly A-stable. As pointed out by Burrage [5], it is of advantage to have the numerical estimating and error estimating method both A-stable."

We agree with Burrage. More precisely, the stability function of an embedded formula for error estimation must be bounded with a reasonable constant for efficient implementation in practice. This ensures that the error estimation formula will not limit step sizes unnecessarily. However, embedded IEQ formulas (24) are not A-stable. They do not satisfy even the weaker property mentioned above, except for formula (b) in scheme (24). Our analysis shows the following: The sixth order formula itself (denoted as Method VIII in [33]) is symmetric, A-stable and has the stability function

$$R_{33}(z) = \frac{1 + 1/2z + 1/10z^2 + 1/120z^3}{1 - 1/2z + 1/10z^2 - 1/120z^3},$$

but is not symplectic. The remaining embedded formulas (a)–(d) are both nonsymmetric and not A-stable. In addition, they are not symplectic and not conjugate to a symplectic method as well (in the sense of Lemma 2). The stability functions of these methods are easily calculated to be:

$$R_{(a)}(z) = \frac{1 + 1/2z + 3/5z^2 + 11/120z^3 + 1/120z^4}{1 - 1/2z + 1/10z^2 - 1/120z^3},$$

$$R_{(\mathrm{b})}(z) = \frac{1 + 1/2z + 1/10z^2 + 11/120z^3}{1 - 1/2z + 1/10z^2 - 1/120z^3},$$
$$R_{(\mathrm{c})}(z) = \frac{1 + 1/2z + 1/10z^2 + 1/120z^3 - 1/144z^4 - 1/720z^5}{1 - 1/2z + 1/10z^2 - 1/120z^3},$$
$$R_{(\mathrm{d})}(z) = \frac{1 + 1/2z + 1/10z^2 + 1/120z^3 - 1/720z^5}{1 - 1/2z + 1/10z^2 - 1/120z^3}.$$

Our conclusion is obviously confirmed by these formulas. Thus, IEQ formula (24) constructed in [33] cannot be recommended for practical integration of stiff problems. Below, we investigate embedded Gauss-type NIRK methods.

Our intention is to construct pairs of embedded NIRK methods with similar properties. We start with Gauss-type NIRK method (10), (11). Thus, we are looking for embedded RK formulas of the following form:

$$\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
c_1^2 & \frac{6(c_1^2+\theta)-5}{12} & \frac{1-\theta}{2} & \frac{1-\theta}{2} & \frac{6(c_1^2+\theta)-7}{12} \\
1-c_1^2 & \frac{7-6(c_1^2+\theta)}{12} & \frac{\theta}{2} & \frac{\theta}{2} & \frac{5-6(c_1^2+\theta)}{12} \\
1 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\
\hline
 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\
\hline
 & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 & \tilde{b}_4
\end{array}$$

Hence, the local error estimator is taken to be

$$le_{k+1} = \tau_k\big(e_1 g(t_k, x_k) - e_2 g(t_{k1}^2, x_{k1}^2) - e_3 g(t_{k2}^2, x_{k2}^2) + e_4 g(t_{k+1}, x_{k+1})\big) \tag{25}$$

where $e_i = \tilde{b}_i - b_i$, $i = 1, 2, 3, 4$, and b_i are the coefficients of method (10), (11).

Unfortunately, any embedded RK formula of order 2 and giving the error estimate (25) is not A-stable. In particular, if the quadrature Trapezoidal Rule is used as the embedded formula for local error evaluation in NIRK method (10), (11); i.e. we choose the coefficients of formula (25) to be $e_1 = e_4 = 1/2$, $e_2 = e_3 = -1/2$, then the stability function of the Embedded Trapezoidal Rule will be

$$R_{\mathrm{ETR}}(z) = \frac{1 + 1/2z + 1/12z^2 + 1/12z^3}{1 - 1/2z + 1/12z^2}.$$

It is obvious that $R_{\mathrm{ETR}}(z)$ is not bounded in the complex plane.

So a direct implementation of the error estimation (25) can be inefficient for integrating some stiff ODE's in practice. In order to make the local error estimate limited for any step size we follow Shampine's idea (see, for instance, [21, p. 123]) and take the solution of the following linear system as the local error estimate:

$$Q_1\big(\tau_k J(t_k, x_k)\big)\widetilde{le}_{k+1} = le_{k+1} \tag{26}$$

where $Q_1(z) = (1 - z/4)^3$ and $J(t_k, x_k)$ is the Jacobi matrix, as defined in Section 1. We emphasize that the error evaluation (26) is not expensive in practice because it means three solutions of linear systems with the coefficient matrix $I - \tau_k J(t_k, x_k)/4$. The latter matrix is computed and decomposed to advance a step of method (10), (11) (see [23] or [22]). It is easy to check that the Gauss-type NIRK method of order 4 and with the error estimation (26) is suitable for integration of stiff problems. Indeed, the stability function of the extrapolated numerical solution (with local error estimate (26)) is

$$\widetilde{R}_{\mathrm{ETR}}(z) = \frac{1 - 1/4z - 5/48z^2 + 19/192z^3 + 1/128z^4 - 1/768z^5}{1 - 5/4z + 31/48z^2 - 11/64z^3 + 3/128z^4 - 1/768z^5};$$

i.e., $\widetilde{R}_{\mathrm{ETR}}(z) = R_{\mathrm{NIRK4}}(z) + Q_1^{-1}(z)(R_{\mathrm{ETR}}(z) - R_{\mathrm{NIRK4}}(z))$ where $R_{\mathrm{NIRK4}}(z)$ is the stability function of method (10), (11). This stability function is bounded.

Table 1
Errors evaluated at the end point 6 for the adaptive Gauss-type NIRK method of order 4 with different error estimation techniques

Error	Error estimation technique				
tolerance	EMEE	ESEE	MEMEE	MESEE	REEE
10^{-1}	1.291×10^{-1}	3.865×10^{-1}	3.315×10^{-1}	$1.049 \times 10^{+0}$	8.691×10^{-2}
5×10^{-2}	5.595×10^{-2}	1.822×10^{-1}	4.692×10^{-1}	9.702×10^{-1}	5.260×10^{-2}
10^{-2}	1.291×10^{-2}	4.410×10^{-2}	2.863×10^{-1}	2.551×10^{-1}	2.655×10^{-2}
5×10^{-3}	1.005×10^{-2}	2.473×10^{-2}	1.338×10^{-1}	1.742×10^{-1}	7.801×10^{-3}
10^{-3}	2.848×10^{-3}	8.317×10^{-3}	8.806×10^{-3}	3.629×10^{-2}	2.959×10^{-3}
5×10^{-4}	1.785×10^{-3}	4.418×10^{-3}	2.675×10^{-3}	1.446×10^{-2}	1.202×10^{-3}
10^{-4}	4.747×10^{-4}	2.044×10^{-3}	7.649×10^{-4}	2.627×10^{-3}	6.468×10^{-4}
5×10^{-5}	2.062×10^{-4}	1.318×10^{-3}	2.578×10^{-4}	1.177×10^{-3}	1.873×10^{-4}
10^{-5}	2.835×10^{-5}	1.116×10^{-4}	5.655×10^{-5}	1.227×10^{-4}	6.389×10^{-5}

We stress that estimation (26) is suitable for stiff ODE's. However, other useful properties of methods (10), (11) seem not to be preserved. For a better error estimate, we want to try the idea of methods with embedded stage values. We recall that the principal feature of NIRK formulas is that the calculation of all stage values is explicit and, hence, very cheap. When $\theta = 1/2 + 2\sqrt{3}/9$ the Gauss-type method (10), (11) is of stage order 3. It is of stage order 2 for other θ's. Therefore if we fix, say, $\hat{\theta} \neq \theta$ then the error estimate is computed by

$$\widehat{le}_{k+1} = x_{k2}^2 - \hat{x}_{k2}^2 \tag{27}$$

where the additional stage values $\hat{x}_{k2}^2$ are calculated for $\hat{\theta}$. Notice that both stage values x_{k2}^2 and $\hat{x}_{k2}^2$ are computed by the same numerical solution obtained by the method (10), (11) of stage order 3. We also use the error of the second stage value only because of the identity $\|x_{k1}^2 - \hat{x}_{k1}^2\| \equiv \|x_{k2}^2 - \hat{x}_{k2}^2\|$. Then, estimate (27) is used in a stepsize selection algorithm in the usual way.

It is easy to see that estimate (27) can be represented in the form

$$\widehat{le}_{k+1} = \frac{\nu\tau_k}{2}\bigl(g(t_k, x_k) - g\bigl(t_{k1}^2, x_{k1}^2\bigr) - g\bigl(t_{k2}^2, x_{k2}^2\bigr) + g(t_{k+1}, x_{k+1})\bigr) \tag{28}$$

where $\nu = \hat{\theta} - \theta$. Again, it is not suitable for stiff ODE's for any ν. However, if we apply Shampine's idea to estimate (28) then the improved estimate

$$Q_2\bigl(\tau_k J(t_k, x_k)\bigr)\overline{le}_{k+1} = \frac{\nu\tau_k}{2}\bigl(g(t_k, x_k) - g\bigl(t_{k1}^2, x_{k1}^2\bigr) - g\bigl(t_{k2}^2, x_{k2}^2\bigr) + g(t_{k+1}, x_{k+1})\bigr) \tag{29}$$

where $Q_2(z) = 1 - z/4$ will be bounded for any step size when $0 < |\nu| \leqslant 1/4$. We take $\nu = 1/4$ in the numerical experiment below. We stress that estimation (29) is cheap because of the reasons given above.

In the next section, we are going to compare all these error estimation techniques on a large-scale numerical example. For the sake of completeness, we include also Richardson extrapolation to evaluate the local error in the Gauss-type NIRK method of order 4 when $\theta = 1/2 + 2\sqrt{3}/9$. In this technique, we continue with the extrapolated numerical solution. The usual stepsize selection is implemented and local errors are evaluated in sup-norm. More numerical experiments showing performance of the error estimations under discussion (in terms of accuracy and CPU time) are presented for nonstiff and stiff problems in [24].

Finally, we mention that similar error estimation techniques are possible in the pair of embedded Gauss-type NIRK methods of orders 4 and 6. They are our plan for future research.

6. Numerical example

For testing, we apply the Gauss-type NIRK method of order 4 to the two-dimensional Brusselator with diffusion and periodic boundary conditions (see [21, p. 151–152] for full detail). We take 50 grid points in each dimension. It results in a system of ODE's of dimension 5000, which is mildly stiff. It is also a good example of large practical problems because of its size.

We solve this problem on the interval [0, 6] by our adaptive method with all error estimations discussed in Section 5. The following abbreviation is used: Embedded Methods Error Estimation (EMEE) is calculated by (25), Modified

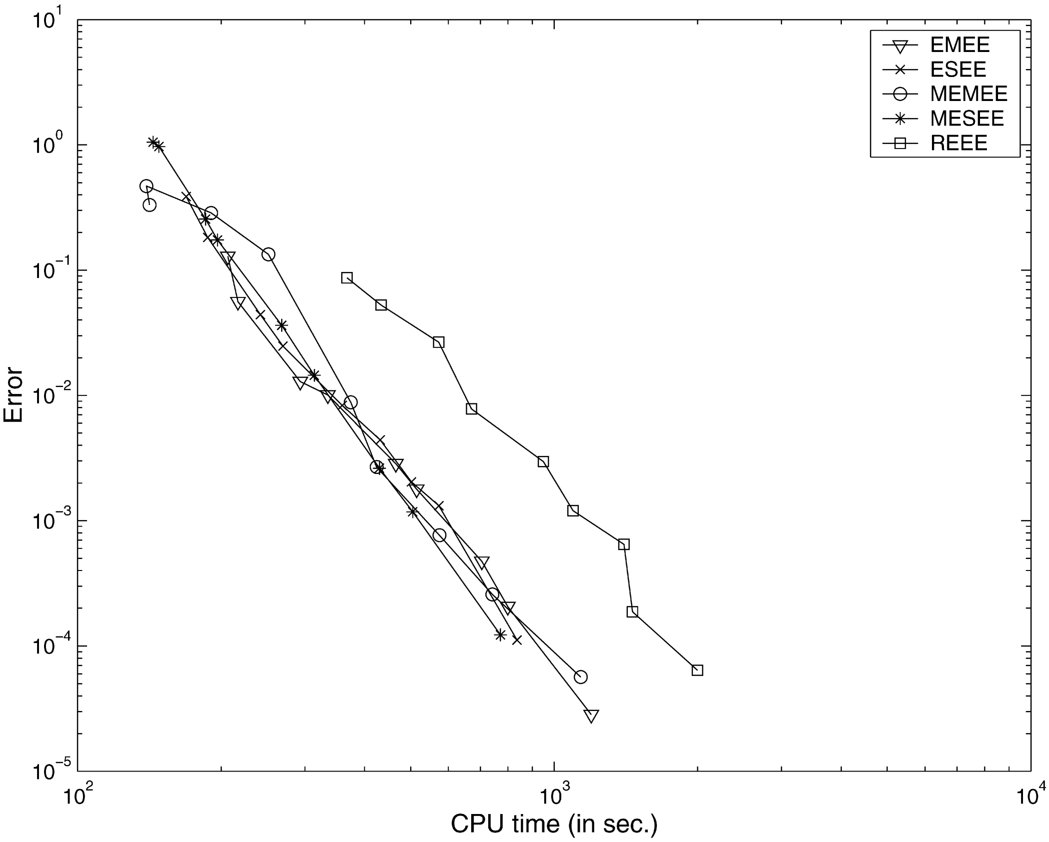


Fig. 1. Accuracy versus CPU time for the adaptive Gauss-type NIRK method of order 4 with different error estimation techniques.

Table 2
CPU time (in sec.) for the adaptive Gauss-type NIRK method of order 4 with different error estimation techniques

Error	Error estimation technique				
tolerance	EMEE	ESEE	MEMEE	MESEE	REEE
10^{-1}	$2.067 \times 10^{+2}$	$1.690 \times 10^{+2}$	$1.416 \times 10^{+2}$	$1.440 \times 10^{+2}$	$3.676 \times 10^{+2}$
5×10^{-2}	$2.171 \times 10^{+2}$	$1.875 \times 10^{+2}$	$1.395 \times 10^{+2}$	$1.482 \times 10^{+2}$	$4.335 \times 10^{+2}$
10^{-2}	$2.932 \times 10^{+2}$	$2.420 \times 10^{+2}$	$1.907 \times 10^{+2}$	$1.856 \times 10^{+2}$	$5.726 \times 10^{+2}$
5×10^{-3}	$3.351 \times 10^{+2}$	$2.699 \times 10^{+2}$	$2.514 \times 10^{+2}$	$1.967 \times 10^{+2}$	$6.710 \times 10^{+2}$
10^{-3}	$4.653 \times 10^{+2}$	$3.592 \times 10^{+2}$	$3.745 \times 10^{+2}$	$2.682 \times 10^{+2}$	$9.497 \times 10^{+2}$
5×10^{-4}	$5.148 \times 10^{+2}$	$4.315 \times 10^{+2}$	$4.240 \times 10^{+2}$	$3.139 \times 10^{+2}$	$1.096 \times 10^{+3}$
10^{-4}	$7.051 \times 10^{+2}$	$5.021 \times 10^{+2}$	$5.747 \times 10^{+2}$	$4.302 \times 10^{+2}$	$1.402 \times 10^{+3}$
5×10^{-5}	$7.990 \times 10^{+2}$	$5.729 \times 10^{+2}$	$7.423 \times 10^{+2}$	$5.049 \times 10^{+2}$	$1.460 \times 10^{+3}$
10^{-5}	$1.196 \times 10^{+3}$	$8.360 \times 10^{+2}$	$1.137 \times 10^{+3}$	$7.717 \times 10^{+2}$	$1.999 \times 10^{+3}$

Embedded Methods Error Estimation (MEMEE) is computed by (26), Embedded Stages Error Estimation (ESEE) is calculated by (28), Modified Embedded Stages Error Estimation (MESEE) is computed by (29) and Richardson Extrapolation Error Estimation (REEE) is obtained by the Richardson technique. Here, we apply the modified Newton iteration with nontrivial predictor and refer the reader to [23] and [22] for full particulars of the implementation of the order 4 Gauss-type NIRK method. Paper [23] presents also a comparison of our NIRK method (10), (11) with the Gauss method of order 4.

The code is written in MATLAB 6.1 and run on a personal computer with processor Intel Pentium IV, 3.0 GHz under operating system MICROSOFT WINDOWS XP. An accuracy versus CPU time plot is displayed in Fig. 1. We have used the same NIRK method with local error control based on REEE for error tolerance $= 10^{-10}$, to calculate the reference solution at the end point of the integration interval [0, 6]. Quantitative information on accuracy obtained and CPU time expended is shown in Tables 1 and 2.

From the results presented, we see that the MESEE approach is at least competitive for large-scale ODE's. So the idea of Embedded Stages Error Estimation looks promising for practical implementation of NIRK methods. Certainly, more numerical experience is required to gain confidence in it. An automatic global error control facility for NIRK methods is planned.

Acknowledgements

The first author thanks the National Research Foundation of South Africa for partial support of his work under grant No. FA2004033000016. The second author thanks the University of the Witwatersrand, Johannesburg for a Post-Doctoral Fellowship. The authors are also very grateful to anonymous referees for their valuable remarks and comments on the paper.

References

[1] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff ODEs, SIAM J. Numer. Anal. 14 (1977) 1006–1024.

[2] R. Alt, Methodes A-stables pour l'integration de systemes differentiels mal conditionnes, PhD thesis, Universite Paris, 1971.

[3] I.S. Berezin, N.P. Zhidkov, Computing Methods, Fizmatgiz, Moscow, 1962 (in Russian). Translation in Pergamon, Oxford, 1965.

[4] T.A. Bickart, An efficient solution process for implicit Runge–Kutta methods, SIAM J. Numer. Anal. 14 (1977) 1022–1027.

[5] K. Burrage, A special family of Runge–Kutta methods for solving stiff differential equations, BIT 18 (1978) 22–41.

[6] K. Burrage, J.C. Butcher, F.H. Chipman, An implementation of singly implicit Runge–Kutta methods, BIT 20 (1980) 326–340.

[7] K. Burrage, F.H. Chipman, P.H. Muir, Order results for mono-implicit Runge–Kutta methods, SIAM J. Numer. Anal. 31 (1994) 876–891.

[8] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, BIT 16 (1976) 237–240.

[9] J.C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley and Sons, Chichester, 2003.

[10] J.C. Butcher, J.R. Cash, Towards efficient Runge–Kutta methods for stiff systems, SIAM J. Numer. Anal. 27 (1990) 753–761.

[11] J.C. Butcher, D. Chen, A new type of singly implicit Runge–Kutta methods, Appl. Numer. Math. 34 (2000) 179–188.

[12] J.R. Cash, A class of implicit Runge–Kutta methods for the numerical solution of stiff ordinary differential equations, J. Assoc. Comput. Mach. 22 (1975) 504–511.

[13] J.R. Cash, On a class of implicit Runge–Kutta procedures, J. Inst. Math. Appl. 19 (1977) 455–470.

[14] J.R. Cash, On a note of the computational aspects of a class of implicit Runge–Kutta procedures, J. Inst. Math. Appl. 20 (1977) 425–441.

[15] J.R. Cash, A. Singhal, Mono-implicit Runge–Kutta formulae for the numerical integration of stiff differential systems, IMA J. Numer. Anal. 2 (1982) 211–227.

[16] M. Crouzeix, Sur l'approximation ds equations differentielles operationnelles lineaires par de methodes de Runge–Kutta, PhD thesis, Universite Paris, 1975.

[17] G. Dahlquist, A special stability problem for linear multistep methods, BIT 3 (1963) 27–43.

[18] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations, North-Holland, Amsterdam, 1984.

[19] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations, Springer-Verlag, Berlin, 2002.

[20] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer-Verlag, Berlin, 1993.

[21] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer-Verlag, Berlin, 1996.

[22] G.Yu. Kulikov, A.I. Merkulov, S.K. Shindin, Asymptotic error estimate for general Newton-type methods and its application to differential equations, Russian J. Numer. Anal. Math. Model. 22 (2007) 567–590.

[23] G.Yu. Kulikov, S.K. Shindin, On a family of cheap symmetric one-step methods of order four, in: V.N. Alexandrov, et al. (Eds.), Computational Science – ICCS 2006, Proceedings, Part I, 6th International Conference, Reading, UK, May 28–31, 2006, Lecture Notes in Computer Science, vol. 3991, Springer-Verlag, Berlin, 2006, pp. 781–785.

[24] G.Yu. Kulikov, S.K. Shindin, Numerical tests with Gauss-type nested implicit Runge–Kutta formulas, in: Y. Shi, et al. (Eds.), Computational Science — ICCS 2007, Proceedings, Part I, 7th International Conference, Beijing, China, May 27–30, 2007, Lecture Notes in Computer Science, vol. 4487, Springer-Verlag, Berlin, 2007, pp. 136–143.

[25] M. Kurdi, Stable high order methods for time discretization of stiff differential equations, PhD thesis, University of California, 1974.

[26] A. Kværnø, Singly diagonally implicit Runge–Kutta methods with an explicit first stage, BIT 44 (2004) 489–502.

[27] P. Leone, Symplecticity and symmetry of general integration methods, Thèse, Section de Mathématiques, Université de Gèneve, 2000.

[28] P.H. Muir, W.E. Enright, Relations amogn some classes of implicit Runge–Kutta methods and their stability functions, BIT 27 (1987) 403–423.

[29] S.P. Nørsett, Semi-explicit Runge–Kutta methods, Report No. 6/74 (1974), Dept. of Math., University of Trondheim, Norway.

[30] S.P. Nørsett, Runge–Kutta methods with multiple real eigenvalue only, BIT 16 (1976) 388–393.

[31] S.P. Nørsett, A. Wolfbrandt, Attainable order of rational approximations to the exponential function with only real poles, BIT 17 (1977) 200–208.

[32] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamilton Problems, Chapman & Hall, Berlin, 1994.

[33] W.M.G. van Bokhoven, Efficient higher order implicit one-step methods for integration of stiff differential equations, BIT 20 (1980) 34–43.