Economical Sixth Order Runge–Kutta Method for Systems of Ordinary Differential Equations

Alexey S. Eremin(✉), Nikolai A. Kovrizhnykh, and Igor V. Olemskoy

Saint-Petersburg State University, Saint-Petersburg, Russia
{a.eremin,i.olemskoj}@spbu.ru, sagoyewatha@mail.ru

Abstract. Structural partitioning of systems of ordinary differential equations is made on base of right-hand side dependencies on the unknown variables. It is used to construct fully explicit Runge–Kutta methods with several computational schemes applied to different parts of the system. The constructed structural methods require fewer right-hand side evaluations (stages) per step for some parts of the system than classic explicit Runge–Kutta methods of the same order. The full structural form of the system is presented, which after permutation of variables can be applied to any system of ordinary differential equation. For such structure a multischeme method is formulated and conditions of the sixth order are written down. We present simplifying conditions and reduce the system to a solvable smaller system. A particular computational scheme, that requires seven stages for a group without special structure and only six stages for other equations, is presented. Its sixth order is confirmed by a numerical convergence test.

Keywords: explicit Runge–Kutta · Partitioned methods · Structural partitioning · Order conditions · Multischeme methods

1 Introduction

Nowadays Runge–Kutta type methods (RKs) are one of the most widely used class of numerical methods for solving ordinary differential equations (ODEs). They have been intensively studied for more than a century and each new challenge in numerical treatment of ODEs like stiffness or geometrical properties preservation gave new inspiration to researchers for constructing and modifying Runge–Kutta methods. There is a lot of literature on the topic. One can find basics in classical textbooks on general numerical analysis, like [12,22], as well as read special monographs on ODE solution as [3,8].

New methods are proposed every year. Obviously they are not new kinds of "classical" explicit methods, but usually are designed to have one or few special properties or are intended for some peculiar classes of ODE systems.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11619, pp. 89–102, 2019.
https://doi.org/10.1007/978-3-030-24289-3_8

It is well known that partitioning of variables in ODE systems helps to simplify or reduce the system analytically. However even without analytical simplification it might be useful to separate the system into two or more parts and treat them numerically in different ways, i.e. to apply different (though interconnected) numerical schemes. Often this partitioning can be done in some natural way, say different parts of the system correspond to the unknowns of different physical nature. In other cases the partitioning is made on base of the system structure, stiffness or characteristic speeds of the processes.

Pioneer papers dealing with partitioning proposed methods for stiff problems of ODEs which were split up into "stiff" and "non-stiff" subsystems. Those methods were named implicit-explicit, since the "non-stiff" part was solve by fast explicit scheme and only the "stiff" part was solved by an implicit one (e.g. [9, 23]). In many recent paper such methods are called multischeme. Multischeme methods are studied to obtain simplectic integrators [13,14,29], to treat partial differential equations with parts of different physical nature [10,11], to solve fast and slow processes in large systems with different step sizes [27,28].

Here we consider the partitioning based on the right-hand side functions dependence on the unknowns. The structure of such dependencies can be used to construct fully explicit multischeme Runge–Kutta methods with fewer total right-hand side evaluations than *classic* explicit RKs require per step (the constraints known as *Butcher barriers* [3,8]).

We have presented several types of structural partitioning in [5]. Using the same denotations as there we present a *full structurally partitioned* system. It contains the *general group* of unknown functions — all those that don't have structural properties — and two *structural groups*.

$$\begin{cases} y_0'(t) = f_0(t, y_0, y_{11}, ..., y_{1,n_1}, y_{21}, ..., y_{2n_2}), \\ y_{1r}'(t) = f_{1r}(t, y_0, y_{11}, ..., y_{1,r-1}, y_{21}, ..., y_{2n_2}), \quad r = 1, ..., n_1, \\ y_{2r}'(t) = f_{2r}(t, y_0, y_{11}, ..., y_{1,n_1}, y_{21}, ..., y_{2,r-1}), \quad r = 1, ..., n_2. \end{cases} \tag{1}$$

We distinguish between every unknown function in the first $(y_{11}, ..., y_{1,n_1})$ and the second $(y_{21}, ..., y_{2,n_2})$ structural groups since the numerical method is implemented differently to each of them. At the same time all the unknowns and equations in the general group are treated in the same way and the denotation y_0 can be formally considered as a vector of arbitrary length.

Any system of ODEs can be rewritten to have the structure (1). The general algorithm of finding the permutation of unknowns to obtain the structure (1) can be found in [17]. It is designed to maximize the size of structural groups, since the advantage over classic RKs is provided for the right-hand sides within them. In the worst case when for every possible r the derivative y_r' depends on y_r, no structural group can be formed and all the equations are included into the vector y_0. In this case the method studied in the present paper reduces to a classic Runge–Kutta method. However many problems of celestial mechanics, high-energy physics or some schemes of partial differential equations spatial discretization allow reformulation with a suitable structure (1) (see e.g. [24–26]).

Remark 1. In some systems all equations are included into the first or second structural group. In this case the advantage over the classic RKs is more considerable. The fifth order method can be constructed with four stages for every equation [15,18] (along with six for classic RKs). The sixth order method for the system with two structural groups requires six stages [1,6,20,21], while classic RKs require seven. The similar advantage can be obtained for continuous Runge–Kutta methods as well [5,7].

In the present paper we construct an explicit partitioned Runge–Kutta type method of order six for the system (1). It uses seven stages for the equations of general group (the mentioned Butcher's barrier) and only six stages for every equation of the first and second structural groups.

In the next section we present the general form of structurally partitioned system and the explicit Runge–Kutta type method for it.

The third section deals with sixth order conditions of the considered method and describes the simplifying assumptions on the methods parameters.

The algorithm of solving the system and one possible solution are given in Sect. 4 and the convergence test for a simple test problem in the last section.

2 Structural Numerical Method

Considering the initial point t_0 we write down the computational scheme for the Runge–Kutta type one-step method for obtaining the approximate solution in the point $t_0 + h$, where h is the step size.

$$\begin{aligned} y_0(t_0+h) &\approx y_0(t_0) + h\sum_{i=1}^{s_0} b_{0i}K_{0i}, \\ y_{1r}(t_0+h) &\approx y_{1r}(t_0) + h\sum_{i=1}^{s_1} b_{1i}K_{1ri}, \quad r = 1,...,n_1, \\ y_{2r}(t_0+h) &\approx y_{2r}(t_0) + h\sum_{i=1}^{s_2} b_{2i}K_{2ri}, \quad r = 1,...,n_2. \end{aligned} \tag{2}$$

The difference from the formal expansion of classic Runge–Kutta methods to the systems of ODEs is that the right-hand side functions are evaluated one by one in strict order K_{01}, K_{111}, K_{121}, ..., K_{1n_11}, K_{211}, ..., K_{2n_21}, K_{02}, K_{112}, ..., K_{1n_12}, K_{212}, etc. according to

$$\begin{aligned} K_{0i} = f_0\Big(&t_0 + c_{0i}h, y_0(t_0) + h\textstyle\sum_{j=1}^{i-1} a_{00ij}K_{0j}, \\ &y_{11}(t_0) + h\textstyle\sum_{j=1}^{i} a_{01ij}K_{11j},\ ...,\ y_{1,n_1}(t_0) + h\sum_{j=1}^{i} a_{01ij}K_{1,n_1,j}, \\ &y_{21}(t_0) + h\textstyle\sum_{j=1}^{i-1} a_{02ij}K_{21j},\ ...,\ y_{2n_2}(t_0) + h\sum_{j=1}^{i-1} a_{02ij}K_{2n_2j}\Big), \end{aligned}$$

$$
\begin{aligned}
K_{1ri} = f_{1r}\Big(&t_0 + c_{1i}h, y_0(t_0) + h\textstyle\sum_{j=1}^{i} a_{10ij}K_{0j}, && r = 1, ..., n_1,\\
&y_{11}(t_0) + h\textstyle\sum_{j=1}^{i} a_{11ij}K_{11j},\ ...,\ y_{1,r-1}(t_0) + h\textstyle\sum_{j=1}^{i} a_{11ij}K_{1,r-1,j},\\
&y_{21}(t_0) + h\textstyle\sum_{j=1}^{i-1} a_{12ij}K_{21j},\ ...,\ y_{2n_2}(t_0) + h\textstyle\sum_{j=1}^{i-1} a_{12ij}K_{2n_2j}\Big),\\
K_{2ri} = f_{2r}\Big(&t_0 + c_{2i}h, y_0(t_0) + h\textstyle\sum_{j=1}^{i} a_{20ij}K_{0j}, && r = 1, ..., n_2,\\
&y_{11}(t_0) + h\textstyle\sum_{j=1}^{i} a_{21ij}K_{11j},\ ...,\ y_{1n_1}(t_0) + h\textstyle\sum_{j=1}^{i} a_{21ij}K_{1n_1j},\\
&y_{21}(t_0) + h\textstyle\sum_{j=1}^{i} a_{22ij}K_{21j},\ ...,\ y_{2,r-1}(t_0) + h\textstyle\sum_{j=1}^{i} a_{22ij}K_{2,r-1,j}\Big).
\end{aligned}
\tag{3}
$$

In each formula i varies in the corresponding range from 1 to s_0, s_1 or s_2. Notice that in general the algorithms demands that either $s_0 = s_1 = s_2$, or $s_0 = s_1 = s_2 + 1$, or $s_1 = s_2 = s_0 - 1$. The last case is the best for efficiency.

After the paper [2] it's being conventional to collect coefficients of Runge–Kutta type methods into so-called Butcher tables. Here we use an extended Butcher table for structural methods of the form

c_0	A_{00}	A_{01}	A_{02}	b_0
c_1	A_{10}	A_{11}	A_{12}	b_1
c_2	A_{20}	A_{21}	A_{22}	b_2

Here $b_v = (b_{v1}, ..., b_{vs_v})^T$ and $c_v = (c_{v1}, ..., c_{vs_v})^T$ are vectors and $A_{uv} = \{a_{uvij}\}$ are $[s_u \times s_v]$-matrices for $u, v = 0, 1, 2$.

Each group (general or structural) is referred to three blocks of parameters A_{uv}. The table form of the method corresponds to its structural properties and their algorithmic use. The matrices A_{00}, A_{01}, A_{02} and A_{12} are strictly lower-triangular (as is the case with classic explicit Runge–Kutta methods), but A_{10}, A_{20}, A_{11}, A_{21} and A_{22} are lower-triangular, i.e. can have non-zero diagonal elements.

Remark 2. As mentioned above, if a system (1) has no structural properties and no structural groups can be formed ($n_1 = n_2 = 0$) then the method reduces to classic Runge–Kutta method using only c_{0i}, b_{0i} and a_{00ij} coefficients. As a result of this we cannot construct a structural method of order p with s_0 smaller than a minimal number of stages required for a classic explicit Runge–Kutta method, i.e. s_0 respects usual Butcher barriers [8]. A structural methods is more efficient when $s_1 = s_2 = s_0 - 1$, and the efficiency grows as the size of the structural groups grows over the size of the general group.

In the following section we present the conditions of the sixth order, which form quite large system of algebraic equations. We apply additional relations between the method's coefficients, named simplifying conditions, and reduce the original large system to a much smaller one.

3 Conditions for the Sixth Order

The method (2), (3) is a one-step explicit method, so the convergence theorems can be applied, i.e. its convergence order equals its local order for smooth enough problems (e.g. [8]). In order to construct a method of order six we should find the coefficients such that the Taylor expansions in respect of h of the exact and numerical solutions are the same up to the h^6 terms, for each component. This means that for any problem smooth enough and some vector norm $\|\cdot\|$ there exist such positive numbers C_0, C_{11}, ..., C_{1n_1}, C_{21}, ..., C_{2n_2} that for any h small enough

$$\begin{aligned}
&\|y_0(t_0+h) - y_0(t_0) - h\sum_{i=1}^{s_0} b_{0i}K_{0i}\| < C_0 h^7,\\
&|y_{1r}(t_0+h) - y_{1r}(t_0) - h\sum_{i=1}^{s_1} b_{1i}K_{1ri}| < C_{1r}h^7, \quad r = 1,...,n_1,\\
&|y_{2r}(t_0+h) - y_{2r}(t_0) - h\sum_{i=1}^{s_2} b_{2i}K_{2ri}| < C_{2r}h^7, \quad r = 1,...,n_2.
\end{aligned} \tag{4}$$

Remark 3. Instead of considering the order componentwise in (4) we could use a vector norm over all unknown functions $y_0(t)$, $y_{11}(t)$, ..., $y_{1n_1}(t)$, $y_{21}(t)$, ..., $y_{2n_2}(t)$ simultaneously, but this doesn't change the order conditions, which follow.

It is well known [8] that a classic sixth order Runge–Kutta method requires at least seven stages and for such method the order conditions form a system of 37 nonlinear algebraic equations with 28 unknown parameters b_{0i} and a_{00ij} (in terms of Remark 2). The parameters c_{0i} are linearly dependent on a_{00ij}, so we don't count them in this consideration.

In [1,20] sixth order methods for systems without general group with only six stages for each unknown were constructed. In case of $n_1 = n_2 = 1$ the number of order conditions is 74 and number of parameters is 48. However if $n_1 > 1$ and $n_2 > 1$ the parameters a_{11ij} and a_{22ij} are used as well and in this case we obtain a system of 292 order conditions for 90 parameters [21].

In case of full system (1) we construct a sixth order method with $(s_0, s_1, s_2) = (7, 6, 6)$ stages. Besides of so-called *basic simplifying conditions* which connect c-parameters to a-parameters

$$\begin{aligned}
&\sum_{j=1}^{i-1} a_{00ij} = \sum_{j=1}^{i-1} a_{01ij} = \sum_{j=1}^{i-1} a_{02ij} = c_{0i}, \quad i = 1,...,7,\\
&\sum_{j=1}^{i} a_{10ij} = \sum_{j=1}^{i} a_{11ij} = \sum_{j=1}^{i-1} a_{12ij} = c_{1i}, \quad i = 1,...,6,\\
&\sum_{j=1}^{i} a_{20ij} = \sum_{j=1}^{i} a_{21ij} = \sum_{j=1}^{i} a_{22ij} = c_{2i}, \quad i = 1,...,6,
\end{aligned} \tag{5}$$

we have a system of 1224 order conditions obtained with the extension of labeled-trees theory for classic Runge–Kutta methods [3,4]. In the following system the indices u, v, w, α and β can be 0, 1 or 2 in any combination. The summation is made over all possible (and meaningful) values of the summation indices.

$$
\begin{aligned}
&\sum_i b_{ui} c_{ui}^q = \frac{1}{q+1}, \quad q = 0, 1, ..., 5, \\
&\sum_i b_{ui} c_{ui}^q \sum_j a_{uvij} c_{vj} = \frac{1}{2 \cdot (q+3)}, \quad q = 0, 1, 2, 3, \\
&\sum_i b_{ui} c_{ui}^q \sum_j a_{uvij} c_{vj}^2 = \frac{1}{3 \cdot (q+4)}, \quad q = 0, 1, 2, \\
&\sum_i b_{ui} c_{ui}^q \sum_j a_{uvij} \sum_k a_{vwjk} c_{wk} = \frac{1}{6 \cdot (q+4)}, \quad q = 0, 1, 2, \\
&\sum_i b_{ui} c_{ui}^q \left(\sum_j a_{uvij} c_{vj} \right) \left(\sum_k a_{uwik} c_{wk} \right) = \frac{1}{4 \cdot (q+5)}, \quad q = 0, 1, \\
&\sum_i b_{ui} c_{ui}^q \sum_j a_{uvij} c_{vj}^3 = \frac{1}{4 \cdot (q+5)}, \quad q = 0, 1, \\
&\sum_i b_{ui} \sum_j a_{uvij} c_{vj}^q \sum_k a_{vwjk} c_{wk}^d = \frac{1}{10(q+d)(1+d)}, \\
&\qquad (q, d) \in \{(0,2), (0,3), (1,1), (1,2), (2,1)\}, \\
&\sum_i b_{ui} \sum_j a_{uvij} c_{vj}^q \sum_k a_{vwjk} c_{wk}^d \sum_l a_{w\alpha kl} c_{\alpha l}^\rho = \frac{1}{60(s+2d+2\rho)}, \\
&\qquad (q, d, \rho) \in \{(0,0,1), (1,0,1), (0,1,1)\}, \\
&\sum_i b_{ui} \left(\sum_j a_{uvij} c_{vj} \right) \left(\sum_j a_{uwik} c_{wk}^2 \right) = \frac{1}{36}, \qquad (6) \\
&\sum_i b_{ui} \sum_j a_{uvij} c_{vj}^4 = \frac{1}{30}, \\
&\sum_i b_{ui} \left(\sum_j a_{uvij} \sum_k a_{vwjk} c_{wk} \right) \left(\sum_l a_{u\alpha il} c_{\alpha l} \right) = \frac{1}{72}, \\
&\sum_i b_{ui} c_{ui} \sum_j a_{uvij} c_{vj} \sum_k a_{vwjk} c_{r,\xi} = \frac{1}{48}, \\
&\sum_i b_{ui} c_{ui} \sum_j a_{uvij} \sum_k a_{vwjk} c_{r,\xi}^2 = \frac{1}{72}, \\
&\sum_i b_{ui} \sum_j a_{uvij} \left(\sum_k a_{vwjk} c_{wk} \right) \left(\sum_l a_{v\alpha jl} c_{\alpha l} \right) = \frac{1}{120},
\end{aligned}
$$

$$\sum_i b_{ui}c_{ui}\sum_j a_{uvij}\sum_k a_{vwjk}\sum_l a_{w\alpha kl}c_{\alpha l} = \frac{1}{144},$$

$$\sum_i b_{ui}\sum_j a_{uvij}\sum_k a_{vwjk}\sum_l a_{w\alpha kl}c_{\alpha l}^2 = \frac{1}{360},$$

$$\sum_i b_{ui}c_{ui}\sum_j a_{uvij}\sum_k a_{vwjk}\sum_l a_{w\alpha kl}\sum_m a_{\alpha\beta lm}c_{\beta m} = \frac{1}{720}.$$

The total number of parameters to be defined (including c-parameters) for (7,6,6) stages is 221.

Every classic sixth order Runge–Kutta method constructed by now has its b_2 parameter equal to zero. We make the same assumption for all three groups and additionally we assume that for all three groups the first right-hand side computation is done in the initial point of the step, i.e.

$$b_{02} = b_{12} = b_{22} = c_{01} = c_{11} = c_{21} = 0. \tag{7}$$

Next step to simplify the system is to introduce special assumptions known in the literature as *simplifying conditions* [3]. For different high-order Runge–Kutta methods different simplifying conditions can be used, each combination resulting into a separate family of solutions. We tried to apply various simplifying conditions and ended up with using different simplifying conditions of classic Runge–Kutta methods for different groups of parameters in system (6). These simplifying conditions are gathered blockwise and presented in the Table 1.

Table 1. Simplifying conditions for the system (6)

$q = 0, 1, 2;\ d = 0, 1;\ i = 3, ..., 7;\ k = 2, ..., 6;\ \kappa = 1, ...5;\ l = 1, ..., 6;\ m = 3, ..., 6$		
$\sum_{j=1}^{i-1} a_{00ij}c_{0j}^q = \frac{c_{0i}^{q+1}}{q+1}$	$\sum_{j=1}^{i-1} a_{01ij}c_{1j}^d = \frac{c_{0i}^{d+1}}{d+1}$	$\sum_{j=1}^{i-1} a_{02ij}c_{2j}^d = \frac{c_{0i}^{d+1}}{d+1}$
$\sum_{j=3}^{7} b_{0j}c_{0j}^{d+1}a_{00j2} = 0$	$\sum_{j=3}^{7} b_{0j}c_{0j}^2 a_{01j2} = 0$	$\sum_{j=3}^{7} b_{0j}c_{0j}^2 a_{02j2} = 0$
$\sum_{j=l+1}^{7} b_{0j}a_{00jl} = b_{0l}(1 - c_{0l})$	$\sum_{j=l+1}^{7} b_{0j}c_{0j}^d a_{01jl} = \frac{b_{1l}(1-c_{1l}^{d+1})}{d+1}$	$\sum_{j=l+1}^{7} b_{0j}c_{0j}^d a_{02jl} = \frac{b_{2l}(1-c_{2l}^{d+1})}{d+1}$
$\sum_{j=1}^{k} a_{10kj}c_{0j}^q = \frac{c_{1k}^{q+1}}{q+1}$	$\sum_{j=1}^{k} a_{11kj}c_{1j}^d = \frac{c_{1k}^{d+1}}{d+1}$	$\sum_{j=1}^{m-1} a_{12mj}c_{2j}^d = \frac{c_{1m}^{d+1}}{d+1}$
$\sum_{j=2}^{6} b_{1j}c_{1j}^{d+1}a_{10j2} = 0$	$\sum_{j=2}^{6} b_{1j}c_{1j}^2 a_{11j2} = 0$	
$\sum_{j=l}^{6} b_{1j}a_{10jl} = b_{0l}(1 - c_{0l})$	$\sum_{j=l}^{6} b_{1j}c_{1j}^d a_{11jl} = \frac{b_{1l}(1-c_{1l}^{d+1})}{d+1}$	$\sum_{j=\kappa+1}^{6} b_{1j}a_{12j\kappa} = b_{2\kappa}(1 - c_{2\kappa})$
$\sum_{j=1}^{k} a_{20kj}c_{0j}^q = \frac{c_{2k}^{q+1}}{q+1}$	$\sum_{j=1}^{k} a_{21kj}c_{1j}^d = \frac{c_{2k}^{d+1}}{d+1}$	$\sum_{j=1}^{k} a_{22kj}c_{2j}^d = \frac{c_{2k}^{d+1}}{d+1}$
$\sum_{j=2}^{6} b_{2j}c_{2j}^{d+1}a_{20j2} = 0$	$\sum_{j=2}^{6} b_{2j}c_{2j}^2 a_{21j2} = 0$	
$\sum_{j=l}^{6} b_{2j}a_{20jl} = b_{0l}(1 - c_{0l})$	$\sum_{j=l}^{6} b_{2j}c_{2j}^d a_{21jl} = \frac{b_{1l}(1-c_{1l}^{d+1})}{d+1}$	$\sum_{j=l}^{6} b_{2j}c_{2j}^d a_{22jl} = \frac{b_{2l}(1-c_{2l}^{d+1})}{d+1}$

The reducing of the original system with use of simplifying conditions can be found in details in [16,21]. Here we present only the reduced system consisting of 47 equations (obtained from 1224 equations of (6)):

$$
\begin{aligned}
&\sum_{i=1}^{s_u} b_{ui} c_{ui}^{q} = \frac{1}{q+1}, \quad u = 0,1,2, \quad q = 0,1,...,5,\\
&\sum_{i=5}^{7} b_{0i} c_{0i} \sum_{j=4}^{i-1} a_{00ij} \sum_{k=3}^{j-1} a_{0wjk} c_{wk}^{2} = \frac{1}{72}, \quad w = 0,1,\\
&\sum_{i=4}^{7} b_{0i} c_{0i} \sum_{j=3}^{i-1} a_{00ij} \sum_{k=2}^{j-1} a_{0wjk} c_{wk}^{2} = \frac{1}{72}, \quad w = 1,2,\\
&\sum_{i=4}^{7} b_{0i} c_{0i} \sum_{j=3}^{i-1} a_{00ij} c_{0j}^{3} = \frac{1}{24},\\
&\sum_{i=4}^{7} b_{0i} c_{0i}^{2} \sum_{j=3}^{i-1} a_{01ij} c_{1j}^{2} = \frac{1}{18}, \qquad \sum_{i=3}^{7} b_{0i} c_{0i}^{2} \sum_{j=2}^{i-1} a_{02ij} c_{2j}^{2} = \frac{1}{18},\\
&\sum_{i=4}^{6} b_{1i} c_{1i} \sum_{j=3}^{i} a_{10ij} \sum_{k=3}^{j-1} a_{00jk} c_{0k}^{2} = \frac{1}{72},\\
&\sum_{i=3}^{6} b_{ui} c_{ui} \sum_{j=3}^{i} a_{u0ij} c_{0j}^{3} = \frac{1}{24}, \quad u = 1,2,\\
&\sum_{i=3}^{6} b_{1i} c_{1i} \sum_{j=3}^{i} a_{10ij} \sum_{k=2}^{j-1} a_{0wjk} c_{wk}^{2} = \frac{1}{72}, \quad w = 1,2,\\
&\sum_{i=4}^{6} b_{1i} c_{1i} \sum_{j=4}^{i} a_{10ij} \sum_{k=3}^{j-1} a_{01jk} c_{1k}^{2} = \frac{1}{72},\\
&\sum_{i=3}^{6} b_{1i} c_{1i}^{2} \sum_{j=3}^{i} a_{10ij} c_{0j}^{2} = \frac{1}{18},\\
&\sum_{i=4}^{6} b_{1i} c_{1i} \sum_{j=3}^{i-1} a_{12ij} \sum_{k=3}^{j} a_{20jk} c_{0k}^{2} = \frac{1}{72},\\
&\sum_{i=4}^{6} b_{2i} c_{2i} \sum_{j=4}^{i} a_{20ij} \sum_{k=3}^{j-1} a_{0wjk} c_{wk}^{2} = \frac{1}{72}, \quad w = 0,1,\\
&\sum_{i=3}^{6} b_{2i} c_{2i} \sum_{j=3}^{i} a_{20ij} \sum_{k=2}^{j-1} a_{0wjk} c_{wk}^{2} = \frac{1}{72}, \quad w = 1,2,\\
&\sum_{i=3}^{6} b_{1i} c_{1i}^{2} \sum_{j=2}^{i} a_{11ij} c_{1j}^{2} = \frac{1}{18}, \qquad \sum_{i=3}^{6} b_{1i} c_{1i}^{2} \sum_{j=2}^{i-1} a_{12ij} c_{2j}^{2} = \frac{1}{18},
\end{aligned}
\tag{8}
$$

$$\sum_{i=3}^{6} b_{1i}c_{1i}\sum_{j=2}^{i-1} a_{12ij}c_{2j}^{q} = \frac{1}{(1+q)(3+q)}, \quad q = 1,2,3;$$

$$\sum_{i=3}^{6} b_{1i}c_{1i}\sum_{j=2}^{i-1} a_{12ij}\sum_{k=2}^{j} a_{2wjk}c_{wk}^{2} = \frac{1}{72}, \quad w = 1,2,$$

$$\sum_{i=4}^{6} b_{1i}c_{1i}\sum_{j=3}^{i-1} a_{12ij}\sum_{k=3}^{j} a_{21jk}c_{1k}^{2} = \frac{1}{72},$$

$$\sum_{i=3}^{6} b_{2i}c_{2i}^{2}\sum_{j=2}^{i} a_{2vij}c_{vj}^{2} = \frac{1}{18}, \quad v = 1,2.$$

Together with 198 simplifying conditions from Table 1 the total number of equations is 245, i.e. five times fewer than the original system (6). The number of unknown parameters (considering conditions (5)) is 202 now.

The reduced system (8) keeps the peculiarities of the original system: visible block structure and nonlinear relations between parameters within and between the blocks.

4 Particular Computational Scheme

Here we make certain special assumptions to find a particular solution of the reduced system:

$$\begin{gathered} c_{02} = a_{0021} = a_{0121} = a_{0221} = \frac{2}{15}, \quad c_{12} = c_{22} = c_{03} = c_{13} = \frac{1}{5}, \\ c_{23} = c_{0,4} = \frac{1}{3}, \quad c_{05} = \frac{2}{3}, \quad c_{06} = \frac{4}{5}, \quad c_{07} = c_{16} = c_{26} = 1. \end{gathered} \tag{9}$$

We leave $c_{24} = \alpha$ to be a free parameter. The remaining node parameters c_{14}, c_{15} and c_{25} can be found from the reduced system:

$$c_{14} = \frac{5\alpha - 2}{5(2\alpha - 1)}, \quad c_{15} = c_{25} = \frac{4\alpha - 3}{5\alpha - 4}.$$

Now the b-parameters (weights) are determined from the first equations of the reduced (or original) system, i.e. from the pure quadrature order conditions:

$$b_{01} = \frac{7}{96}, \quad b_{03} = \frac{125}{672}, \quad b_{04} = \frac{27}{112}, \quad b_{05} = \frac{27}{112}, \quad b_{06} = \frac{125}{672}, \quad b_{07} = \frac{7}{96},$$

$$b_{11} = \frac{15\alpha^2 - 18\alpha + 5}{12(5\alpha - 2)(4\alpha - 3)}, \quad b_{13} = \frac{125(5\alpha^2 - 5\alpha + 1)}{48(3\alpha - 1)(15\alpha - 11)},$$

$$b_{14} = \frac{125(2\alpha - 1)^5}{12(3\alpha - 1)(15\alpha^2 - 20\alpha + 7)(5\alpha - 2)(5\alpha - 3)},$$

$$b_{15} = \frac{(5\alpha - 4)^5}{12(15\alpha^2 - 20\alpha + 7)(15\alpha - 11)(\alpha - 1)(4\alpha - 3)},$$

$$b_{16} = \frac{15\alpha^2 - 27\alpha + 11}{48(5\alpha - 3)(\alpha - 1)}, \quad b_{21} = \frac{25\alpha^2 - 20\alpha + 1}{60\alpha(4\alpha - 3)},$$

$$b_{24} = \frac{1}{60(3\alpha - 1)(5\alpha^2 - 8\alpha + 3)\alpha(\alpha - 1)},$$

$$b_{25} = \frac{(5\alpha - 4)^5}{60(5\alpha^2 - 8\alpha + 3)(7\alpha - 5)(\alpha - 1)(4\alpha - 3)},$$

$$b_{23} = \frac{81(5\alpha^2 - 5\alpha + 1)}{40(3\alpha - 1)(7\alpha - 5)}, \quad b_{26} = \frac{5\alpha^2 - 15\alpha + 9}{120(\alpha - 1)^2}.$$

It is natural to demand that all c-parameters lie in $[0, 1]$ and that all weights b are positive, i.e. lie in $[0, 1]$ as well. This is provided when α satisfies

$$\alpha \in \left\{ \left(\frac{2 - \sqrt{3}}{5}, \frac{1}{5} \right) \cup \left(\frac{1}{5}, \frac{5 - \sqrt{5}}{10} \right) \right\}.$$

We choose $\alpha = \frac{1}{10}$ and solve the rest of the system as a sequence of linear subsystems. We omit the details and present only the computed values of the parameters.

$$A_{00} = \begin{pmatrix} 0 & & & & & & \\ \frac{2}{15} & & & & & & \\ \frac{1}{20} & \frac{3}{20} & & & & & \\ \frac{11}{108} & \frac{-5}{36} & \frac{10}{27} & & & & \\ \frac{23}{54} & \frac{-5}{18} & \frac{-35}{54} & \frac{7}{6} & & & \\ \frac{-83}{125} & \frac{3}{5} & \frac{9}{5} & \frac{-189}{125} & \frac{72}{125} & & \\ \frac{23}{28} & \frac{-15}{28} & \frac{-80}{49} & \frac{108}{49} & \frac{-18}{49} & \frac{25}{49} & \end{pmatrix}, \quad A_{01} = \begin{pmatrix} 0 & & & & & & \\ \frac{2}{15} & & & & & & \\ \frac{1}{10} & \frac{1}{10} & & & & & \\ \frac{1}{18} & \frac{-5}{54} & \frac{10}{27} & & & & \\ \frac{20}{81} & \frac{-5}{27} & \frac{-5}{27} & \frac{64}{81} & & & \\ \frac{-206}{975} & \frac{2}{5} & \frac{62}{95} & \frac{-2944}{7725} & \frac{43218}{127205} & & \\ \frac{145}{546} & \frac{-5}{14} & \frac{80}{931} & \frac{7936}{15141} & \frac{12250}{25441} & 0 & \end{pmatrix},$$

$$A_{02} = \begin{pmatrix} 0 & & & & & \\ \frac{2}{15} & & & & & \\ \frac{1}{10} & \frac{1}{10} & & & & \\ \frac{1}{18} & \frac{5}{18} & 0 & & & \\ \frac{-4}{9} & \frac{-25}{9} & \frac{5}{3} & \frac{20}{9} & & \\ \frac{314}{325} & \frac{22}{5} & \frac{-1494}{1075} & \frac{-88}{25} & \frac{4802}{13975} & \\ \frac{-141}{182} & \frac{-45}{14} & \frac{2988}{2107} & \frac{1360}{441} & \frac{2450}{5031} & 0 \end{pmatrix},$$

$$C_0 = \begin{pmatrix} 0 \\ \frac{2}{15} \\ \frac{1}{5} \\ \frac{1}{3} \\ \frac{2}{3} \\ \frac{4}{5} \\ 1 \end{pmatrix}, \quad A_{10} = \begin{pmatrix} 0 & & & & & \\ \frac{1}{20} & \frac{3}{20} & & & & \\ \frac{1}{20} & \frac{3}{20} & & & & \\ \frac{255}{2048} & -\frac{495}{2048} & \frac{945}{2048} & \frac{63}{2048} & & \\ \frac{30953}{600250} & \frac{429}{3430} & \frac{2717}{168070} & \frac{199329}{600250} & \frac{457938}{2100875} & \\ \frac{95}{676} & -\frac{165}{676} & \frac{395}{1183} & \frac{393}{1183} & -\frac{45}{1183} & \frac{1125}{2366} \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 0 \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{3}{8} \\ \frac{26}{35} \\ 1 \end{pmatrix}, \quad A_{11} = \begin{pmatrix} 0 & & & & \\ \frac{1}{10} & \frac{1}{10} & & & \\ \frac{1}{10} & 0 & \frac{1}{10} & & \\ \frac{1}{24} & 0 & \frac{5}{16} & \frac{1}{48} & \\ \frac{18817}{154350} & 0 & \frac{871}{36015} & \frac{252928}{540225} & \frac{9}{70} \\ -\frac{3581}{39546} & 0 & \frac{107935}{134862} & -\frac{370688}{1096641} & \frac{2701125}{4299529} & 0 \end{pmatrix}, \tag{10}$$

$$C_2 = \begin{pmatrix} 0 \\ \frac{1}{5} \\ \frac{1}{3} \\ \frac{1}{10} \\ \frac{26}{35} \\ 1 \end{pmatrix}, \quad A_{12} = \begin{pmatrix} 0 & & & & & \\ \frac{1}{5} & & & & & \\ \frac{1}{10} & \frac{1}{10} & & & & \\ \frac{45}{1024} & \frac{615}{2048} & \frac{63}{2048} & & & \\ -\frac{167401}{60025} & -\frac{244751}{24010} & \frac{3393039}{840350} & \frac{814112}{84035} & & \\ \frac{60745}{4394} & \frac{16415}{338} & -\frac{848271}{50869} & -\frac{54400}{1183} & \frac{120050}{94471} & \end{pmatrix},$$

$$A_{20} = \begin{pmatrix} 0 & & & & & \\ \frac{1}{20} & \frac{3}{20} & & & & \\ \frac{11}{108} & -\frac{5}{36} & \frac{10}{27} & & & \\ \frac{19}{800} & \frac{33}{160} & -\frac{5}{32} & \frac{21}{800} & & \\ \frac{6877}{120050} & \frac{429}{3430} & -\frac{143}{33614} & \frac{1677}{4802} & \frac{90558}{420175} & \\ \frac{20}{151} & -\frac{165}{604} & \frac{865}{2114} & \frac{615}{2114} & -\frac{81}{2114} & \frac{2025}{4228} \end{pmatrix}, \quad B_0 = \begin{pmatrix} \frac{7}{96} \\ 0 \\ \frac{125}{672} \\ \frac{27}{112} \\ \frac{27}{112} \\ \frac{125}{672} \\ \frac{7}{96} \end{pmatrix},$$

$$A_{21} = \begin{pmatrix} 0 & & & & & \\ \frac{1}{10} & \frac{1}{10} & & & & \\ \frac{1}{18} & -\frac{5}{54} & \frac{10}{27} & & & \\ \frac{163}{1800} & \frac{11}{80} & -\frac{7}{48} & \frac{4}{225} & & \\ \frac{2773}{22050} & \frac{143}{1715} & -\frac{1378}{19551} & \frac{3792256}{7949025} & \frac{3483}{27398} & \\ -\frac{1876}{17667} & -\frac{55}{302} & \frac{61790}{60249} & -\frac{362624}{979839} & \frac{4862025}{7683182} & 0 \end{pmatrix}, \quad B_1 = \begin{pmatrix} \frac{67}{936} \\ 0 \\ \frac{1375}{6384} \\ \frac{8192}{32445} \\ \frac{2100875}{5495256} \\ \frac{169}{2160} \end{pmatrix},$$

$$A_{22} = \begin{pmatrix} 0 & & & & & \\ \frac{1}{10} & \frac{1}{10} & & & & \\ \frac{1}{18} & \frac{5}{18} & 0 & & & \\ \frac{2}{15} & \frac{1}{12} & 0 & -\frac{7}{60} & & \\ -\frac{547}{14406} & -\frac{5603}{7203} & \frac{69498}{84035} & \frac{152048}{252105} & \frac{9}{70} & \\ \frac{2011}{5889} & \frac{2195}{906} & -\frac{42768}{45451} & -\frac{4640}{3171} & \frac{108045}{168818} & 0 \end{pmatrix}, \quad B_2 = \begin{pmatrix} \frac{5}{104} \\ 0 \\ \frac{891}{2408} \\ \frac{200}{1701} \\ \frac{420175}{1086696} \\ \frac{151}{1944} \end{pmatrix}.$$

5 Numerical Convergence Test

Since we don't have an error estimator embedded into the presented scheme, we make only the convergence test to confirm the declared sixth order of the method.

The system of four equations

$$\begin{aligned} y_1' &= 2xy_1y_3 && = f_1(x, y_1, y_3), \\ y_2' &= 2xy_3 && = f_2(x, y_3), \\ y_3' &= -2x(y_2 - 1) && = f_3(x, y_2), \\ y_4' &= 10xy_3y_1^5 && = f_4(x, y_1, y_3), \end{aligned} \tag{11}$$

with the initial conditions $y_1(0) = y_2(0) = y_3(0) = y_4(0) = 1$ has the exact solution

$$y_1 = \exp(\sin x^2), \quad y_2 = \sin x^2 + 1, \quad y_3 = \cos x^2, \quad y_2 = \exp(5 \sin x^2).$$

Here we have 1 unknown in the general group (y_1), one in the first structural group (y_2) and two in the second structural group (y_3, y_4).

We solve it at the interval $[0, 5]$ with constant step size h and compute the maximal norm of the exact error in every step *Err*. We change the step size as $h = \frac{1}{3} \cdot \left(\frac{3}{2}\right)^k$ for $k = 1, ..., 11$. The results are presented at Fig. 1a and in the Table 2 as Test 1. The average slope of the segmented line for h small enough shows the rate of convergence in double logarithmic scale. The test confirms the sixth order of the method.

In order to check the case of several equations in the first structural group, we make the change of variables $u_1 = y_1$, $u_2 = y_3$, $u_3 = y_4$, $u_4 = y_2$. In this case two equation are in the first group and one is in the second group. The results of the same test for this system are presented at Fig. 1b and in the Table 2 as Test 2.

Table 2. Results of the numerical tests of convergence

k	Test 1		Test 2	
	Err	Order	*Err*	Order
1	14.6832		5.00611	
2	1.11106	6.15	1.95241	2.24
3	$2.58338 \cdot 10^{-1}$	3.52	$2.41613 \cdot 10^{-1}$	5.04
4	$8.90772 \cdot 10^{-3}$	8.18	$6.31942 \cdot 10^{-3}$	8.85
5	$3.24414 \cdot 10^{-4}$	8.17	$3.66468 \cdot 10^{-4}$	7.02
6	$3.35677 \cdot 10^{-5}$	5.59	$4.19328 \cdot 10^{-5}$	5.35
7	$2.64534 \cdot 10^{-6}$	6.27	$3.36406 \cdot 10^{-6}$	6.22
8	$2.46517 \cdot 10^{-7}$	5.85	$3.02240 \cdot 10^{-7}$	5.94
9	$2.11060 \cdot 10^{-8}$	6.05	$2.70710 \cdot 10^{-8}$	5.94
10	$1.79768 \cdot 10^{-9}$	6.07	$2.28522 \cdot 10^{-9}$	6.09
11	$2.09015 \cdot 10^{-10}$	5.31	$2.29279 \cdot 10^{-10}$	5.67

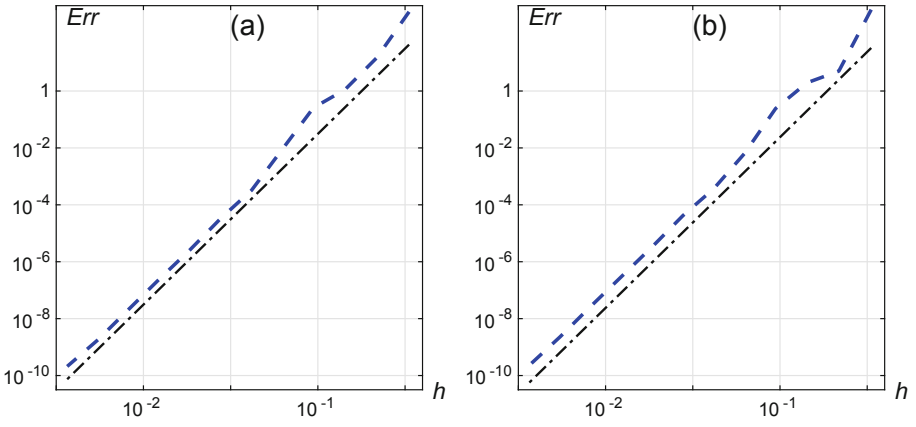

Fig. 1. The global error to step size ratio in the double logarithmic scale. The reference lines have slope 6.

6 Conclusion

The presented sixth order method for many systems transformable to the structure (1) provides better per step performance than formal application of classic Runge–Kutta methods to such systems. However, practical use of RKs nowadays require an automatic time step size control which is usually based on local error estimation. For RKs it is most often made with use of embedded scheme of lower order, and it can also be done within the structural partitioning approach [18,19]. This requires solution of even more complicated system of order conditions.

References

1. Bubnov, V.P., Eremin, A.S., Kovrizhnykh, N.A., Olemskoy, I.V.: Comparative study of the advantages of structural numerical integration methods for ordinary differential equations. SPIIRAS Proc. **4**(53), 51–72 (2017)
2. Butcher, J.C.: On Runge-Kutta processes of high order. J. Australian Math. Soc. **4**(2), 179–194 (1964)
3. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 2nd edn. Wiley, Chichester (2008)
4. Butcher, J.C., Chan, T.M.H.: The tree and forest spaces with applications to initial-value problem methods. BIT Numer. Math. **50**, 713–728 (2010)
5. Eremin, A.S., Kovrizhnykh, N.A.: Continuous extensions for structural Runge–Kutta methods. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10405, pp. 363–378. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62395-5_25
6. Eremin, A.S., Kovrizhnykh, N.A., Olemskoy, I.V.: An explicit one-step multischeme sixth order method for systems of special structure. Appl. Math. Comput. **347**, 853–864 (2019)
7. Eremin, A.S., Olemskoy, I.V.: Functional continuous Runge-Kutta methods for special systems. In: Proceedings of the International Conference on Numerical Analysis and Applied Mathematics 2015 (ICNAAM-2015)/AIP Conference Proceedings, vol. 1738, p. 100003 (2016)
8. Hairer, E., Nørsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems. Springer Series in Computational Mathematics, 3rd edn. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78862-1
9. Hofer, E.: A partially implicit method for large stiff systems of ODEs with only few equations introducing small time-constants. SIAM J. Numer. Anal. **13**(5), 645–663 (1976)
10. Huang, D., Persson, P.O., Zahr, M.: High-order, linearly stable, partitioned solvers for general multiphysics problems based on implicit-explicit Runge-Kutta schemes. Comput. Methods Appl. Mech. Eng. **346**, 674–706 (2019)
11. Ketcheson, D.I., MacDonald, C., Ruuth, S.J.: Spatially partitioned embedded Runge-Kutta methods. SIAM J. Numer. Anal. **51**(5), 2887–2910 (2013)
12. Kincaid, D., Cheney, W.: Numerical Analysis: Mathematics of Scientific Computing, Pure and Applied Undergraduate Texts, vol. 2, 3rd edn. AMS, Providence (2002)
13. McLachlan, R., Ryland, B., Sun, Y.: High order multisymplectic Runge-Kutta methods. SIAM J. Sci. Comput. **36**(5), A2199–A2226 (2014)

14. Monovasilis, T., Kalogiratou, Z., Simos, T.E.: A family of trigonometrically fitted partitioned Runge-Kutta symplectic methods. Appl. Math. Comput. **209**(1), 91–96 (2009)
15. Olemskoy, I.V.: Fifth-order four-stage method for numerical integration of special systems. Comput. Math. Math. Phys. **42**(8), 1135–1145 (2002)
16. Olemskoy, I.V.: Structural approach to the design of explicit one-stage methods. Comput. Math. Math. Phys. **43**(7), 918–931 (2003)
17. Olemskoy, I.V.: Modifikatsiya algoritma vydeleniya strukturnykh osobennostei [modification of structural properties detection algorithm]. Vestn. St-Petersburg Uni. Appl. Math. Comp. Sci. Contr. Proc. **2**, 55–64 (2006). [In Russian]
18. Olemskoy, I.V., Eremin, A.S.: An embedded method for integrating systems of structurally separated ordinary differential equations. Comput Math. Math. Phys. **50**(3), 414–427 (2010)
19. Olemskoy, I.V., Eremin, A.S.: An embedded fourth order method for solving structurally partitioned systems of ordinary differential equations. Appl. Math. Sci. **9**(97–100), 4843–4852 (2015)
20. Olemskoy, I.V., Eremin, A.S., Ivanov, A.P.: Sixth order method with six stages for integrating special systems of ordinary differential equations. In: 2015 International Conference "Stability and Control Processes" in Memory of V.I. Zubov (SCP), pp. 110–113 (2015)
21. Olemskoy, I.V., Kovrizhnykh, N.A.: A family of sixth-order methods with six stages. Vestn. St-Petersburg Uni. Appl. Math. Comp. Sci. Contr. Proc. **14**(3), 215–229 (2018)
22. Ralston, A., Rabinowitz, P.: A First Course in Numerical Analysis. Dover Books on Mathematics, 2nd edn. Dover Publications, New York (2001)
23. Rentrop, P.: Partitioned Runge-Kutta methods with stiffness detection and stepsize control. Numer. Math. **47**, 545–564 (1985)
24. Respondek, J.S.: Controllability of dynamical systems with constraints. Syst. Control Lett. **54**(4), 293–314 (2005)
25. Respondek, J.S.: Numerical simulation in the partial differential equation controllability analysis with physically meaningful constraints. Math. Comput. Simul. **81**(1), 120–132 (2010)
26. Respondek, J.S.: Incremental numerical recipes for the high efficient inversion of the confluent Vandermonde matrices. Comput. Math. Appl. **71**(2), 489–502 (2016)
27. Sandu, A., Günther, M.: A generalized-structure approach to additive Runge-Kutta methods. SIAM J. Numer. Anal. **53**(1), 17–42 (2015)
28. Sandu, A., Günther, M.: Multirate generalized additive Runge-Kutta methods. Numer. Math. **133**(3), 497–524 (2016)
29. Wang, D., Xiao, A., Li, X.: Parametric symplectic partitioned Runge-Kutta methods with energy-preserving properties for Hamiltonian systems. Comput. Phys. Commun. **184**(2), 303–310 (2013)