



Residual-based optimization of fifth-order Runge-Kutta methods

Séka Hippolyte^{1,*}, N'gohissé Konan Firmin², and Taha Manin Mathurin¹

¹Institut National Polytechnique Félix Houphouët-Boigny, Côte d'Ivoire.

²Université Peleforo GON COULIBALY, Côte d'Ivoire.

Abstract

In this work, we study a parametric family of fifth-order Runge–Kutta methods depending on a free parameter and their optimization using an approach based on minimizing the sum of squares of the residuals associated with higher-order conditions. Unlike the classical approach restricted to conditions of order 6, we successively consider conditions of orders 6, 7, and 8 in order to assess their influence on the quality of the optimization. The results suggest that the use of order-8 conditions in the construction of the error function may lead, in several cases, to improved satisfaction of higher-order conditions compared with optimizations based on orders 6 or 7. This improvement appears to result from a more comprehensive reduction of higher-order approximation defects, allowing a more balanced distribution of errors. This study highlights the interest of incorporating higher-order conditions into optimization procedures for Runge–Kutta methods and provides a residual-based strategy for selecting free parameters in parametric families of methods. Finally, the optimized methods obtained are compared with the classical Butcher, Fehlberg, and Dormand–Prince methods.

Keywords. Runge–Kutta methods, order conditions, residual minimization, numerical methods for ODEs, Butcher theory.

2010 Mathematics Subject Classification. 65L05, 65L06, 65D30.

1. INTRODUCTION

Explicit Runge–Kutta methods are fundamental tools for the numerical solution of ordinary differential equations and constitute one of the most widely used time integration techniques in scientific computing. Owing to their simplicity, efficiency, and high-order accuracy, they have been successfully applied in numerous fields, including the numerical solution of nonlinear partial differential equations. Recent applications include the numerical approximation of the Korteweg–de Vries–Burgers equation, the nonlinear Schrödinger equation, the extended Fisher–Kolmogorov equation and the modified Burgers equation, where Runge–Kutta time integration is combined with differential quadrature methods based on B-spline approximations [2–5, 15].

The construction of high-order Runge–Kutta schemes relies on satisfying an increasing number of algebraic order conditions derived from Butcher's theory [7]. However, the number of these conditions grows rapidly with the order, making the explicit construction of high-order methods particularly difficult. For example, the numbers of order conditions for orders 5, 6, 7 and 8 are respectively 17, 37, 85 and 200 [1, 7, 10, 14].

In this work, we consider a family of explicit fifth-order Runge–Kutta methods depending on a free parameter [7, 11, 14]. The main objective is to determine the value of this parameter that yields the best overall numerical performance according to a suitable optimization criterion.

Several approaches have been proposed for selecting free parameters in Runge–Kutta methods, including the minimization of principal error coefficients, improvement of stability properties, and optimization for specific applications.

Received: *; Accepted: *.

* Corresponding author. Email: hippolyte.seka@inphb.ci.

In this work, we adopt an alternative strategy based on minimizing the residuals associated with unsatisfied higher-order conditions. This leads naturally to introducing a global objective function defined as the sum of squares of these residuals [1, 6, 8, 9, 12, 13].

More precisely, for a parameter vector θ , we define

$$S_q^2(\theta) = \sum_k \left(\Phi_k^{(q)}(A(\theta), b(\theta), c(\theta)) - \alpha_k^{(q)} \right)^2, \quad (1.1)$$

where $\Phi_k^{(q)}$ denotes the polynomial expressions associated with the order conditions of order $q > p$, and $\alpha_k^{(q)}$ are the corresponding exact constants.

Most optimization procedures available in the literature are restricted to the immediately higher order ($q = p + 1$). Although this strategy can reduce the dominant truncation error, it exploits only a small portion of the available higher-order algebraic information.

To the best of our knowledge, residual-based optimization of free parameters in Runge–Kutta methods has not been systematically investigated. The present work builds upon Butcher’s algebraic framework [8] while introducing a different optimization philosophy. In contrast to the Dormand–Prince approach [9], which minimizes principal error coefficients, the proposed method directly minimizes the residuals of higher-order conditions. Similarly, although Mahnken [12] employs residuals for global error estimation and correction, they do not formulate them as an optimization criterion for determining free parameters. The originality of the present work lies in transforming the complete set of higher-order residuals into a global optimization problem.

The proposed methodology successively considers the residual functions associated with orders $q = 6, 7$, and 8 in order to investigate how the choice of the optimization order influences the optimal parameter and the resulting numerical method. Although the proposed optimization strategy can, in principle, be extended to higher values of q , the number of algebraic order conditions increases very rapidly with the order. Consequently, the associated optimization problems become considerably more expensive from both symbolic and computational viewpoints. The cases $q = 6, 7$, and 8 therefore provide a meaningful compromise between improved accuracy and computational feasibility.

The numerical experiments indicate that minimizing $S_8^2(\theta)$ generally produces methods satisfying higher-order algebraic conditions more accurately than optimizations based on $S_6^2(\theta)$ or $S_7^2(\theta)$. This suggests that incorporating additional higher-order conditions leads to a more balanced distribution of approximation errors and consequently improves the overall quality of the resulting Runge–Kutta method.

2. A NEW FAMILY OF THE METHODS OF ORDER 5

Let $y' = f(t, y)$, $t \in \mathbb{R}$, $y \in \mathbb{R}^n$ and the initial condition $y(t_0) = y_0$. We choose a small step h and find the vector $y_1 = y(t_1) = y(t_0 + h)$ according to the 6-stage formulas:

$$\begin{aligned} k_1 &= hf(t_0, y_0), \\ k_2 &= hf(t_0 + c_2h, y_0 + a_{21}k_1), \\ k_3 &= hf(t_0 + c_3h, y_0 + a_{31}k_1 + a_{32}k_2), \\ k_4 &= hf(t_0 + c_4h, y_0 + a_{41}k_1 + a_{42}k_2 + a_{43}k_3), \\ k_5 &= hf(t_0 + c_5h, y_0 + a_{51}k_1 + a_{52}k_2 + a_{53}k_3 + a_{54}k_4), \\ k_6 &= hf(t_0 + c_6h, y_0 + a_{61}k_1 + a_{62}k_2 + a_{63}k_3 + a_{64}k_4 + a_{65}k_5), \\ y_1 &= y_0 + b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4 + b_5k_5 + b_6k_6. \end{aligned} \quad (2.1)$$

All the coefficients of the method are generally recorded in the form of a Butcher table.



TABLE 1. Butcher tableau of RK5 method.

0						
c_2	$a_{2,1}$					
c_3	$a_{3,1}$	$a_{3,2}$				
c_4	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$			
c_5	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}$		
c_6	$a_{6,1}$	$a_{6,2}$	$a_{6,3}$	$a_{6,4}$	$a_{6,5}$	
	b_1	b_2	b_3	b_4	b_5	b_6
						b_7

Equalities must be respected

$$\begin{aligned}
 c_2 &= a_{21}, \\
 c_3 &= a_{31} + a_{32}, \\
 c_4 &= a_{41} + a_{42} + a_{43}, \\
 c_5 &= a_{51} + a_{52} + a_{53} + a_{54}, \\
 c_6 &= a_{61} + a_{62} + a_{63} + a_{64} + a_{65}, \\
 1 &= b_1 + b_2 + b_3 + b_4 + b_5 + b_6.
 \end{aligned} \tag{2.2}$$

The coefficients of the method are not arbitrary; they must satisfy a system of polynomial equations (order conditions). For methods of order 5, this involves the system of 17 equations $E_1, \dots, E_{17}$, whose explicit expressions are given in Butcher [7]. The Butcher fifth-order method is given by the coefficients

$$b_1 = \frac{7}{90}; b_2 = 0; b_3 = \frac{16}{45}; b_4 = \frac{2}{15}; b_5 = \frac{16}{45}; b_6 = \frac{7}{90} \tag{2.3}$$

$$c_2 = \frac{1}{4}; c_3 = \frac{1}{4}; c_4 = \frac{1}{2}; c_5 = \frac{3}{4}; c_6 = 1; \tag{2.4}$$

The remaining coefficients a_{ij} are obtained by solving the classical fifth-order Butcher order conditions $E_1, \dots, E_{17}$ using the coefficients 2.3–2.4:

$$a_{2,1} = \frac{1}{4}; a_{3,1} = \frac{1}{8}; a_{3,2} = \frac{1}{8}; \tag{2.5}$$

$$a_{4,1} = 0; a_{4,2} = -\frac{1}{2} + \frac{4a_{5,3}}{3}; a_{4,3} = -\frac{4a_{5,3}}{3} + 1; \tag{2.6}$$

$$a_{5,1} = \frac{3}{16}; a_{5,2} = -a_{5,3}; a_{5,4} = \frac{9}{16}; \tag{2.7}$$

$$a_{6,1} = \frac{-3}{7}; a_{6,2} = \frac{2}{7} + \frac{16a_{5,3}}{7}; a_{6,3} = \frac{12}{7} - \frac{16a_{5,3}}{7}; a_{6,4} = \frac{-12}{7}; a_{6,5} = \frac{8}{7} \tag{2.8}$$

Remark 2.1. For $a_{5,3} = \frac{3}{8}$, we obtain the Butcher method of order 5.

Our objective is to determine the value of the coefficient $a_{5,3}$ that optimizes the considered family of methods. More precisely, we seek the parameter value that minimizes the residuals associated with higher-order conditions.

The underlying idea is to select the fifth-order method that best satisfies the algebraic conditions of higher orders $q \geq 6$. In this work, we consider the cases $q = 6, 7$, and 8.



3. COMPLETE CONSTRUCTION OF THE ERROR FUNCTION

Let an explicit Runge–Kutta method with s stages, defined by its Butcher tableau (A, b, c) , be of order $p = 5$. For higher orders $q > 5$, the corresponding order conditions are generally not satisfied and give rise to residuals. For a fixed order q , the order conditions can be written as

$$\Phi_k^{(q)}(A, b, c) = \alpha_k^{(q)}, \quad k = 1, \dots, N_q, \quad (3.1)$$

where $\Phi_k^{(q)}$ are polynomial functions of the coefficients of the method, and $\alpha_k^{(q)}$ are known rational constants. We define the residuals by

$$E_k^{(q)} = \Phi_k^{(q)}(A, b, c) - \alpha_k^{(q)}. \quad (3.2)$$

A global error functional is then introduced as

$$S_q^2 = \sum_{k=1}^{N_q} \left(E_k^{(q)} \right)^2. \quad (3.3)$$

The residuals $E_k = \Phi_k - \alpha_k$ measure the extent to which the higher-order Butcher order conditions are violated. For an explicit Runge–Kutta method of order p , the local truncation error admits the expansion

$$LTE = C_{p+1} h^{p+1} + O(h^{p+2}), \quad (3.4)$$

where the coefficient C_{p+1} depends on the unsatisfied higher-order order conditions. Consequently, reducing the residuals E_k decreases the magnitude of these leading error coefficients. Although minimizing the residual function $S_q^2 = \sum_k E_k^2$ does not directly minimize the local truncation error, it reduces the algebraic defects responsible for the leading error terms and is therefore expected to improve the local accuracy of the resulting Runge–Kutta method. This theoretical observation is confirmed by the numerical experiments presented in Section 5, where methods associated with smaller residual functions generally produce smaller maximum absolute errors.

In the case $q = 6$, the order conditions can be written explicitly, leading to the following expression of the error functional:

$$\begin{aligned} S_6^2 = & (E_{18})^2 + (E_{19})^2 + (E_{20})^2 + (E_{21})^2 + (E_{22})^2 + (E_{23})^2 + (E_{24})^2 + (E_{25})^2 \\ & + (E_{26})^2 + (E_{27})^2 + (E_{28})^2 + (E_{29})^2 + (E_{30})^2 + (E_{31})^2 + (E_{32})^2 + (E_{33})^2 \\ & + (E_{34})^2 + (E_{35})^2 + (E_{36})^2 + (E_{37})^2. \end{aligned} \quad (3.5)$$

Using symbolic computations performed with Maple 18 on an HP EliteBook 840 G6 laptop equipped with an Intel® Core™ i7-8665U processor running at 1.90 GHz, 16 GB of RAM, and a 64-bit operating system, we obtain the following expression as a function of the free parameter $a_{5,3}$.

$$S_6^2(a_{5,3}) = \frac{383}{66355200} - \frac{1}{5529600} a_{5,3} + \frac{1}{184320} a_{5,3}^2. \quad (3.6)$$

Proceeding similarly for order $q = 7$, using the conditions E_{18} to E_{85} , we obtain

$$\begin{aligned} S_7^2(a_{5,3}) = & \frac{588443}{26011238400} - \frac{823}{144506880} a_{5,3} + \frac{752401}{19508428800} a_{5,3}^2 \\ & - \frac{1357}{1219276800} a_{5,3}^3 + \frac{3481}{3657830400} a_{5,3}^4. \end{aligned} \quad (3.7)$$

Finally, using the conditions up to order $q = 8$ (from E_{18} to E_{200}), we obtain

$$\begin{aligned} S_8^2(a_{5,3}) = & \frac{229934269}{3329438515200} - \frac{6145367}{277453209600} a_{5,3} + \frac{67802087}{624269721600} a_{5,3}^2 \\ & - \frac{162433}{39016857600} a_{5,3}^3 + \frac{20411}{8360755200} a_{5,3}^4. \end{aligned} \quad (3.8)$$

4. MINIMIZATION OF ERROR FUNCTIONS

The optimization is carried out using the order conditions of 6, 7 and 8.



4.1. Optimization using the conditions of order 6. We consider the function

$$S_6^2(a_{5,3}) = \frac{383}{66355200} - \frac{1}{5529600}a_{5,3} + \frac{1}{184320}a_{5,3}^2. \quad (4.1)$$

This is a quadratic polynomial, which can be written in the form

$$S_6^2(x) = Ax^2 + Bx + C, \quad (4.2)$$

where

$$A = \frac{1}{184320}, \quad B = -\frac{1}{5529600}, \quad C = \frac{383}{66355200}. \quad (4.3)$$

Since $A > 0$, the function S_6^2 is strictly convex and admits a unique global minimum. The optimal value is obtained by solving $(S_6^2)'(x) = 0$, which gives

$$a_{5,3}^{\text{opt}} = -\frac{B}{2A} = \frac{1}{60}. \quad (4.4)$$

The second derivative is constant and given by

$$(S_6^2)''(x) = \frac{1}{92160} > 0, \quad (4.5)$$

which confirms that $a_{5,3}^{\text{opt}}$ is the unique global minimizer.

4.2. Optimization using the conditions of order 7. We consider the function

$$\begin{aligned} S_7^2(a_{5,3}) = & \frac{588443}{26011238400} - \frac{823}{144506880}a_{5,3} + \frac{752401}{19508428800}a_{5,3}^2 \\ & - \frac{1357}{1219276800}a_{5,3}^3 + \frac{3481}{3657830400}a_{5,3}^4. \end{aligned} \quad (4.6)$$

This is a quartic polynomial, which can be written in the form

$$S_7^2(x) = Ax^4 + Bx^3 + Cx^2 + Dx + E, \quad (4.7)$$

where

$$\begin{aligned} A = & \frac{3481}{3657830400}, \quad B = -\frac{1357}{1219276800}, \\ C = & \frac{752401}{19508428800}, \quad D = -\frac{823}{144506880}, \\ E = & \frac{588443}{26011238400}. \end{aligned} \quad (4.8)$$

The first derivative is given by

$$(S_7^2)'(x) = 4Ax^3 + 3Bx^2 + 2Cx + D. \quad (4.9)$$

Solving the equation $(S_7^2)'(x) = 0$ using symbolic computation (Maple 18), we obtain the optimal value

$$\begin{aligned} a_{5,3}^{\text{opt}} = & -\frac{1}{472}\Delta^{1/3} + \frac{742879}{236\Delta^{1/3}} + \frac{69}{236}, \\ \Delta = & 230203602 + 82\sqrt{495652894875359}, \\ a_{5,3}^{\text{opt}} \approx & 0.0740509534. \end{aligned} \quad (4.10)$$

The second derivative is

$$(S_7^2)''(x) = 12Ax^2 + 6Bx + 2C, \quad (4.11)$$

and satisfies

$$(S_7^2)''(a_{5,3}^{\text{opt}}) \approx 7.6704 \times 10^{-5} > 0. \quad (4.12)$$

Therefore, the function S_7^2 admits a local minimum at $a_{5,3}^{\text{opt}}$.



4.3. Optimization using the conditions of order 8. We consider the function

$$S_8^2(a_{5,3}) = \frac{229934269}{3329438515200} - \frac{6145367}{277453209600}a_{5,3} + \frac{67802087}{624269721600}a_{5,3}^2 - \frac{162433}{39016857600}a_{5,3}^3 + \frac{20411}{8360755200}a_{5,3}^4. \quad (4.13)$$

This is a quartic polynomial, which can be written in the form:

$$S_8^2(x) = \frac{229934269}{3329438515200} - \frac{6145367}{277453209600}x + \frac{67802087}{624269721600}x^2 - \frac{162433}{39016857600}x^3 + \frac{20411}{8360755200}x^4. \quad (4.14)$$

The derivative gives

$$(S_8^2)'(x) = -\frac{6145367}{277453209600} + \frac{67802087}{312134860800}x - \frac{162433}{13005619200}x^2 + \frac{20411}{2090188800}x^3 \quad (4.15)$$

Solving the equation $(S_8^2)'(x) = 0$ using symbolic computation (Maple 18), we obtain the optimal value

$$\begin{aligned} x^{\text{opt}} &= -\frac{1}{23330099576} \Delta^{1/3} + \frac{4724949234449}{28 \Delta^{1/3}} + \frac{487299}{1143016} \\ \Delta &= 44827093759042410368047359567732 \\ &\quad + 5832524894\sqrt{1852787123198187770522334778848228391976954326} \\ x^{\text{opt}} &\approx 0.1025223395 \end{aligned} \quad (4.16)$$

The second derivative is

$$(S_8^2)''(x) = \frac{67802087}{312134860800} - \frac{162433}{6502809600}x + \frac{20411}{696729600}x^2. \quad (4.17)$$

and satisfies

$$(S_8^2)''(x^{\text{opt}}) \approx 0.0002149675133 > 0. \quad (4.18)$$

Therefore, the function S_8^2 admits a local minimum at $a_{5,3}^{\text{opt}} = x^{\text{opt}}$.

4.4. Comparison of the results. For each value of q , the optimal value of the parameter $a_{5,3}$ is obtained for the proposed method:

Order q	$a_{5,3}^{\text{opt}}$
6	≈ 0.01666666667
7	≈ 0.0740509534
8	≈ 0.1025223395

(4.19)

We also compare the values of the residual functional S_q^2 for classical methods and the proposed optimized methods:

q	Butcher	Fehlberg	Dormand–Prince	New method
6	0.6467×10^{-5}	556.27×10^{-5}	0.2105×10^{-5}	0.5770×10^{-5}
7	2.5871×10^{-5}	1191.12×10^{-5}	6.1406×10^{-5}	2.2412×10^{-5}
8	7.5849×10^{-5}	1618.50×10^{-5}	14.6122×10^{-5}	6.7930×10^{-5}

(4.20)

These results show that the Dormand–Prince method provides a better satisfaction of order 6 conditions, while the optimized methods yield smaller residuals for orders 7 and 8. This indicates an improved fulfillment of higher-order algebraic conditions.

A smaller value of S_q^2 reflects a reduced violation of the corresponding order conditions and suggests a more accurate approximation of higher-order terms, although this does not necessarily guarantee improved global accuracy.



To provide a comprehensive assessment of the proposed optimization strategy, four benchmark problems of increasing complexity are considered. These include a nonhomogeneous linear equation, a nonlinear scalar problem, a nonlinear coupled system, and a damped oscillatory linear system. Together, these examples evaluate the accuracy, convergence, and computational efficiency of the proposed methods on representative classes of ordinary differential equations.

5. NUMERICAL EXPERIMENTS

5.1. Nonhomogeneous linear equation. Consider the Nonhomogeneous linear initial value problem

$$y' = -5y + \sin x, \quad y(0) = 0, \quad (5.1)$$

whose exact solution is

$$y(x) = \frac{1}{26} (5 \sin x - \cos x + e^{-5x}). \quad (5.2)$$

To evaluate the proposed optimization strategy of nonhomogeneous linear equation and to investigate the influence of the step size, additional computations were performed using the step sizes $h = 0.1, 0.05, 0.025$ and 0.0125 . These values provide a representative range for assessing both the convergence behavior and the computational efficiency of the considered methods. Smaller step sizes were not considered because the observed errors are already sufficiently small for a reliable comparison of the methods, whereas further reducing the step size would mainly increase the computational time while providing only marginal additional information. For each method, Table 2 reports the maximum absolute error over the interval $[0, 1]$, together with the corresponding numbers of integration steps and function evaluations. Since all the considered fifth-order Runge–Kutta methods employ six stages, they require the same number of function evaluations for a given step size, thereby allowing a direct comparison of numerical accuracy at an identical computational cost.

TABLE 2. Maximum absolute errors and computational cost for the nonhomogeneous linear test problem using different step sizes.

h	Steps	No. of function evaluations	Butcher	Dormand–Prince	Fehlberg	New (S_6^2)	New (S_7^2)	New (S_8^2)
0.1	10	60	3.87×10^{-7}	2.80×10^{-7}	6.11×10^{-7}	1.76×10^{-7}	8.56×10^{-8}	7.17×10^{-8}
0.05	20	120	1.03×10^{-8}	5.96×10^{-9}	1.57×10^{-8}	3.41×10^{-9}	2.12×10^{-9}	1.92×10^{-9}
0.025	40	240	2.95×10^{-10}	1.52×10^{-10}	4.42×10^{-10}	8.21×10^{-11}	6.10×10^{-11}	5.52×10^{-11}
0.0125	80	480	8.83×10^{-12}	4.28×10^{-12}	1.32×10^{-11}	2.23×10^{-12}	1.82×10^{-12}	1.65×10^{-12}

Table 2 confirms the expected convergence behavior of all the considered methods. As the step size decreases, the maximum absolute error decreases significantly for every method, while the computational cost increases proportionally with the numbers of integration steps and function evaluations. The optimized methods exhibit a systematic improvement in accuracy as the optimization order increases from $q = 6$ to $q = 8$, with $S_6^2 > S_7^2 > S_8^2$ for all the tested step sizes. Moreover, each of the optimized methods consistently outperforms the classical Butcher, Dormand–Prince, and Fehlberg methods throughout the entire range of step sizes. In particular, the method obtained by minimizing S_8^2 achieves the smallest maximum absolute errors, whereas the methods associated with S_6^2 and S_7^2 also provide substantial improvements over the classical fifth-order schemes. Since all the methods require exactly the same number of function evaluations for a given step size, the improved numerical accuracy is achieved without any additional computational cost. These results demonstrate that minimizing higher-order residual functions is an effective strategy for constructing more accurate fifth-order Runge–Kutta methods while preserving their computational efficiency.

To complement the quantitative results reported in Table 2, Fig. 1 illustrates the variation of the maximum absolute error with respect to the step size for all the considered methods. The logarithmic scale on the vertical axis provides a clearer visualization of the convergence behavior and facilitates the comparison of the numerical accuracy achieved by the classical and optimized fifth-order Runge–Kutta methods.

Figure 1 confirms the conclusions drawn from Table 2. As expected, the maximum absolute error decreases monotonically with the step size for all the considered methods, illustrating the convergence of the fifth-order Runge–Kutta



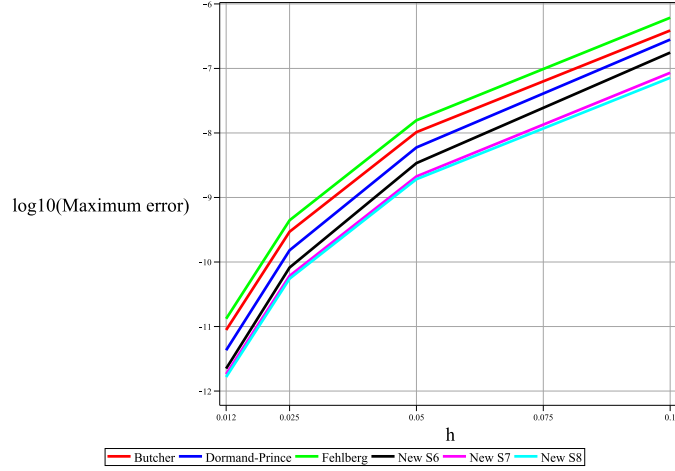


FIGURE 1. Maximum absolute errors versus step size for the nonhomogeneous linear test problem.

schemes. Since all the methods employ the same number of stages, reducing the step size increases the computational cost proportionally through the corresponding increase in the numbers of integration steps and function evaluations. The figure also illustrates the progressive improvement achieved by the residual-based optimization strategy as the optimization order increases from $q = 6$ to $q = 8$. The optimized methods satisfy $S_6^2 > S_7^2 > S_8^2$ for all the tested step sizes. Moreover, the three optimized methods consistently produce smaller maximum absolute errors than the classical Butcher, Dormand–Prince, and Fehlberg methods over the entire range of step sizes. In particular, the method obtained by minimizing S_8^2 achieves the highest accuracy while requiring exactly the same number of function evaluations as the other fifth-order Runge–Kutta methods. These results demonstrate that minimizing higher-order residual functions provides a systematic and effective way of improving the numerical accuracy without increasing the computational cost.

5.2. Nonlinear scalar problem. Consider the nonlinear initial value problem

$$y' = \sqrt{1 - (y - 1)^2}, \quad y(0) = 1, \quad (5.3)$$

whose exact solution on the interval $[0, 1]$ is

$$y(x) = 1 + \sin x. \quad (5.4)$$

To evaluate the proposed optimization strategy of nonlinear scalar problem and to investigate the influence of the step size, additional computations were performed using the step sizes $h = 0.1, 0.05, 0.025$ and 0.0125 . These values provide a representative range for assessing both the convergence behavior and the computational efficiency of the considered methods. Smaller step sizes were not considered because the observed errors are already sufficiently small for a reliable comparison of the methods, whereas further reducing the step size would mainly increase the computational time while providing only marginal additional information. For each method, Table 3 reports the maximum absolute error over the interval $[0, 1]$, together with the corresponding numbers of integration steps and function evaluations. These results allow both the numerical accuracy and the computational efficiency of the considered fifth-order Runge–Kutta methods to be assessed.

The results presented in Table 3 show that the maximum absolute error decreases steadily as the step size is reduced, confirming the expected convergence behavior of all the considered fifth-order Runge–Kutta methods. As expected, this improvement in accuracy is accompanied by a proportional increase in the computational cost, since halving the step size doubles both the number of integration steps and the number of function evaluations.

Regarding accuracy, the optimized methods exhibit progressively better performance as the optimization order increases from $q = 6$ to $q = 8$. Although the method obtained by minimizing S_6^2 does not systematically outperform



TABLE 3. Maximum absolute errors and computational cost for different step sizes.

h	Steps	No. of function evaluations	Butcher	Dormand–Prince	Fehlberg	New (S_6^2)	New (S_7^2)	New (S_8^2)
0.1	10	60	8.85×10^{-9}	3.18×10^{-9}	4.42×10^{-8}	5.61×10^{-9}	3.28×10^{-9}	2.13×10^{-9}
0.05	20	120	2.87×10^{-10}	1.12×10^{-10}	1.30×10^{-9}	1.52×10^{-10}	8.17×10^{-11}	4.67×10^{-11}
0.025	40	240	9.06×10^{-12}	3.75×10^{-12}	3.89×10^{-11}	4.38×10^{-12}	2.22×10^{-12}	1.16×10^{-12}
0.0125	80	480	2.84×10^{-13}	1.21×10^{-13}	1.19×10^{-12}	1.31×10^{-13}	6.44×10^{-14}	3.14×10^{-14}

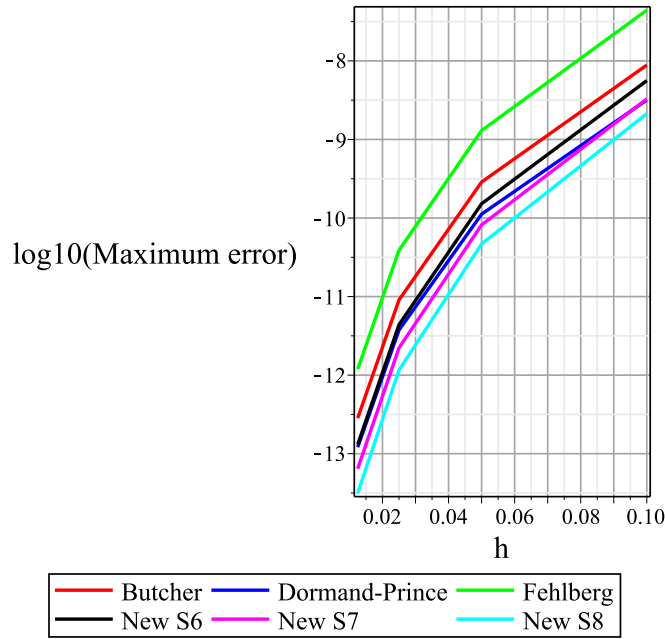


FIGURE 2. Maximum absolute errors versus step size.

the Dormand–Prince method, the optimizations based on S_7^2 and especially on S_8^2 provide significantly smaller errors. In particular, the method associated with S_8^2 consistently achieves the highest accuracy for all the tested step sizes while requiring exactly the same computational effort as the other fifth-order Runge–Kutta methods. These results demonstrate that incorporating higher-order residuals into the optimization process leads to a substantial improvement in numerical accuracy without increasing the computational cost.

To complement the quantitative results reported in Table 3, Fig. 2 illustrates the variation of the maximum absolute error with respect to the step size for all the considered methods. The logarithmic scale on the vertical axis facilitates the comparison of the convergence behavior and highlights the differences in accuracy between the classical and optimized Runge–Kutta methods.

Figure 2 confirms the conclusions drawn from Table 3. As the step size decreases, the maximum absolute error decreases for all the considered methods, demonstrating the expected convergence behavior of the fifth-order schemes. Moreover, the optimized methods become progressively more accurate as the optimization order increases from $q = 6$ to $q = 8$. While the method obtained from minimizing S_6^2 does not consistently outperform the Dormand–Prince method, the optimization based on S_7^2 yields comparable accuracy and generally improves upon the S_6^2 method. In particular, the method obtained by minimizing S_8^2 consistently achieves the smallest maximum absolute errors over the entire range of step sizes. Since all the methods require the same number of function evaluations for a given step size, the improved accuracy achieved by the S_8^2 method is obtained without any additional computational cost.

TABLE 4. Maximum absolute errors and computational cost for the nonlinear coupled test problem using different step sizes.

h	Steps	No. of function evaluations	Butcher	Dormand–Prince	Fehlberg	New (S_6^2)	New (S_7^2)	New (S_8^2)
0.1	10	60	3.12×10^{-8}	2.31×10^{-8}	1.26×10^{-7}	3.72×10^{-8}	2.62×10^{-8}	2.08×10^{-8}
0.05	20	120	9.60×10^{-10}	6.78×10^{-10}	3.55×10^{-9}	1.03×10^{-9}	7.14×10^{-10}	5.55×10^{-10}
0.025	40	240	2.97×10^{-11}	2.06×10^{-11}	1.05×10^{-10}	3.04×10^{-11}	2.08×10^{-11}	1.60×10^{-11}
0.0125	80	480	9.21×10^{-13}	6.37×10^{-13}	3.20×10^{-12}	9.19×10^{-13}	6.25×10^{-13}	4.78×10^{-13}

5.3. **Nonlinear coupled test problem.** Let's consider the following system

$$\begin{cases} y_1' = \sqrt{1 - (y_1 - 1)^2} + y_2 - e^{-x}, \\ y_2' = -y_2 + y_1 - (1 + \sin x), \\ y_1(0) = 1, \quad y_2(0) = 1, \end{cases} \quad (5.5)$$

The nonlinear coupled test problem given by (5.5) is constructed so that its exact solution is given by

$$y_1(x) = 1 + \sin x, \quad y_2(x) = e^{-x}. \quad (5.6)$$

To evaluate the proposed optimization strategy of nonlinear coupled test problem and to investigate the influence of the step size, additional computations were performed using the step sizes $h = 0.1, 0.05, 0.025$ and 0.0125 . These values provide a representative range for assessing both the convergence behavior and the computational efficiency of the considered methods. Smaller step sizes were not considered because the observed errors are already sufficiently small for a reliable comparison of the methods, whereas further reducing the step size would mainly increase the computational time while providing only marginal additional information. For each method, Table 4 reports the maximum absolute error over the interval $[0, 1]$, together with the corresponding numbers of integration steps and function evaluations. Since all the considered fifth-order Runge–Kutta methods employ six stages, they require the same number of function evaluations for a given step size. Consequently, the reported results allow a direct comparison of the numerical accuracy achieved at an identical computational cost.

The results presented in Table 4 confirm the expected convergence behavior of all the considered methods, as the maximum absolute error decreases consistently with the step size. This increase in accuracy is accompanied by a proportional increase in the computational cost, reflected by the growth in the numbers of integration steps and function evaluations. Regarding accuracy, the optimized methods exhibit a clear improvement as the optimization order increases from $q = 6$ to $q = 8$, with the errors satisfying $S_6^2 > S_7^2 > S_8^2$ for all the tested step sizes. Although the method obtained by minimizing S_6^2 is generally less accurate than the classical Butcher method, and the Dormand–Prince method occasionally produces slightly smaller errors than the method associated with S_7^2 , the optimization based on S_8^2 consistently yields the smallest errors among all the considered methods. These results indicate that incorporating higher-order residuals into the optimization process significantly enhances the numerical accuracy of the resulting fifth-order Runge–Kutta method. In particular, the method obtained by minimizing S_8^2 achieves this improvement without increasing the computational cost, since it requires exactly the same number of function evaluations as the classical methods.

To complement the quantitative results reported in Table 4, Fig. 3 illustrates the variation of the maximum absolute error with respect to the step size for all the considered methods. The logarithmic scale on the vertical axis provides a clearer visualization of the convergence behavior and facilitates the comparison of the numerical accuracy achieved by the classical and optimized fifth-order Runge–Kutta methods.

Figure 3 confirms the numerical trends observed in Table 4. For all the considered methods, the maximum absolute error decreases monotonically as the step size is reduced, demonstrating the expected convergence behavior. The optimized methods also exhibit progressively improved accuracy as the optimization order increases from $q = 6$ to $q = 8$, with $S_6^2 > S_7^2 > S_8^2$ throughout the range of step sizes. It is worth noting that the method obtained by minimizing S_6^2 does not consistently outperform the classical Butcher method, while the Dormand–Prince method occasionally yields slightly smaller errors than the method associated with S_7^2 . Nevertheless, the method obtained



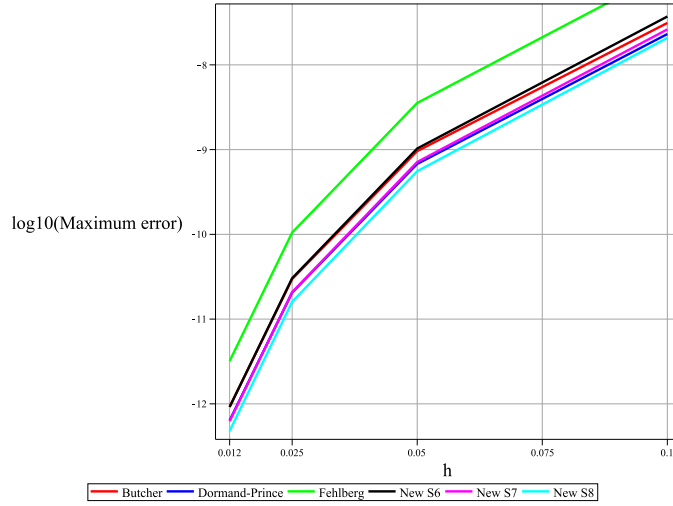


FIGURE 3. Maximum absolute errors versus step size.

from minimizing S_2^2 consistently produces the smallest errors among all the methods considered. Since all the methods employ the same number of stages and therefore require the same number of function evaluations for a given step size, the observed improvement in accuracy is achieved without any additional computational cost. These results provide further numerical evidence that incorporating higher-order residuals into the optimization process leads to more effective fifth-order Runge–Kutta methods.

5.4. Damped oscillatory linear system. To illustrate the impact on numerical performance, we consider the test problem

$$\begin{cases} y_1' = -10y_1 + 3y_2, \\ y_2' = -3y_1 - 10y_2 \\ y_1(0) = 0, \quad y_2(0) = 1, \end{cases} \quad (5.7)$$

whose exact solution is given by

$$y_1(x) = e^{-10x} \sin 3x, \quad y_2(x) = e^{-10x} \cos 3x. \quad (5.8)$$

To further evaluate the robustness of the proposed optimization strategy, we consider the damped rotational system (5.7). This coupled linear problem combines oscillatory and exponentially decaying behaviors, making it a more demanding benchmark than the previous scalar test problems. To evaluate the proposed optimization and to investigate the influence of the step size, additional computations were performed using the step sizes $h = 0.1, 0.05, 0.025$ and 0.0125 . These values provide a representative range for assessing both the convergence behavior and the computational efficiency of the considered methods. Smaller step sizes were not considered because the observed errors are already sufficiently small for a reliable comparison of the methods, whereas further reducing the step size would mainly increase the computational time while providing only marginal additional information. For each method, Table 5 reports the maximum absolute error over the interval $[0, 1]$, together with the corresponding numbers of integration steps and function evaluations. Since all the fifth-order Runge–Kutta methods considered employ six stages, they require the same number of function evaluations for a given step size, allowing a direct comparison of numerical accuracy at an identical computational cost.

Table 5 confirms the expected convergence behavior of all the considered methods. As the step size decreases, the maximum absolute error is significantly reduced for every method, while the computational cost increases proportionally with the number of integration steps and function evaluations. Regarding accuracy, the optimized methods exhibit

TABLE 5. Maximum absolute errors and computational cost for different step sizes.

h	Steps	No. of function evaluations	Butcher	Dormand–Prince	Fehlberg	New (S_6^2)	New (S_7^2)	New (S_8^2)
0.1	10	60	5.60×10^{-4}	5.93×10^{-4}	9.48×10^{-4}	4.14×10^{-4}	2.62×10^{-4}	1.88×10^{-4}
0.05	20	120	1.26×10^{-5}	9.14×10^{-6}	2×10^{-5}	5.74×10^{-6}	2.85×10^{-6}	1.47×10^{-6}
0.025	40	240	3.34×10^{-7}	1.95×10^{-7}	5.14×10^{-7}	1.12×10^{-7}	4.12×10^{-8}	9.70×10^{-9}
0.0125	80	480	9.61×10^{-9}	4.98×10^{-9}	1.46×10^{-8}	2.69×10^{-9}	7.31×10^{-10}	2.82×10^{-10}

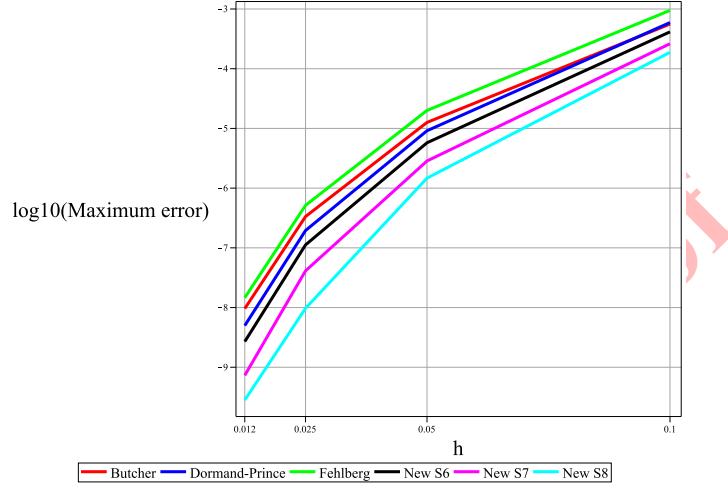


FIGURE 4. Comparison of the maximum absolute errors of the classical and optimized fifth-order Runge–Kutta methods for different step sizes.

a clear improvement as the optimization order increases from $q = 6$ to $q = 8$, with $S_6^2 > S_7^2 > S_8^2$ for all the tested step sizes. Unlike the previous nonlinear scalar problem, the method obtained by minimizing S_6^2 already outperforms both the classical Butcher and Dormand–Prince methods for every step size considered. Furthermore, the methods associated with S_7^2 and especially S_8^2 provide progressively smaller errors, the latter consistently achieving the highest accuracy among all the compared methods. Since all the methods require the same number of function evaluations, these results demonstrate that the proposed residual-based optimization strategy improves the numerical accuracy without increasing the computational cost. They also indicate that incorporating higher-order residual conditions becomes increasingly beneficial as the optimization order increases.

To further assess the accuracy of the considered methods, Figure 4 presents the maximum absolute error as a function of the step size. The comparison includes the classical Butcher, Dormand–Prince, and Fehlberg methods together with the three optimized methods obtained by minimizing the higher-order residuals.

As expected, the maximum error decreases as the step size is reduced for all methods. The optimized methods consistently outperform the classical schemes over the entire range of step sizes. In particular, the method obtained by minimizing the eighth-order residuals (New S8) produces the smallest errors, followed by the methods New S7 and New S6. These results confirm that incorporating higher-order residual information into the optimization process significantly improves the accuracy of the sixth-order Runge–Kutta method.

5.5. Efficiency comparison. An important aspect in assessing the performance of numerical methods is their computational efficiency, that is, the accuracy achieved for a given computational cost. Since all the considered fifth-order explicit Runge–Kutta methods employ six stages, they require exactly the same number of function evaluations for a given step size. Consequently, comparing the maximum absolute errors obtained with identical numbers of function evaluations provides a fair and meaningful measure of efficiency.



TABLE 6. Summary of the efficiency comparison between the classical and optimized fifth-order Runge–Kutta methods.

Test problem	Same computational cost	Most accurate method
Nonhomogeneous linear	Yes	New (S_8^2)
Nonlinear scalar	Yes	New (S_8^2)
Nonlinear coupled	Yes	New (S_8^2)
Damped rotational	Yes	New (S_8^2)

Table 6 summarizes the efficiency comparison for the four benchmark problems considered in this work.

As shown in Table 6, all the methods are compared under identical computational conditions, since they require the same number of function evaluations. The numerical experiments indicate that the optimized methods generally provide smaller maximum absolute errors than the classical Butcher, Dormand–Prince, and Fehlberg methods without increasing the computational cost. Among the optimized schemes, the method obtained by minimizing the eighth-order residual function (S_8^2) consistently exhibits the best overall accuracy over the benchmark problems considered. These results demonstrate that the proposed residual-based optimization strategy improves the efficiency of the underlying fifth-order Runge–Kutta family by achieving higher accuracy while preserving the same computational cost.

This comparison confirms that the observed gain in accuracy is achieved through a more effective choice of the free parameter rather than through additional computational effort.

6. CONCLUSION

In this paper, we have constructed a one-parameter family of explicit fifth-order Runge–Kutta methods depending on the free coefficient $a_{5,3}$. The classical Butcher fifth-order method is recovered as the particular case corresponding to $a_{5,3} = \frac{3}{8}$.

A residual-based optimization strategy was proposed to determine the optimal value of the free parameter by minimizing the sum of squares of the residuals associated with higher-order conditions. Unlike classical approaches based on minimizing principal error coefficients, the proposed methodology provides a systematic criterion for selecting the free parameter while reducing the algebraic defects associated with unsatisfied higher-order conditions.

The numerical experiments performed on four benchmark problems, including a nonhomogeneous linear problem, a nonlinear scalar problem, a nonlinear coupled system, and a damped rotational system, demonstrate the effectiveness of the proposed approach. Convergence studies carried out with several step sizes show that all the considered methods exhibit the expected fifth-order convergence behavior. The optimized methods become progressively more accurate as the optimization order increases from $q = 6$ to $q = 8$, with the method obtained by minimizing S_8^2 consistently providing the best overall performance. Moreover, these improvements are achieved without increasing the computational cost, since the optimized methods require the same number of function evaluations as the corresponding classical fifth-order Runge–Kutta methods.

Overall, the numerical results indicate that incorporating higher-order residual conditions into the optimization process constitutes an effective strategy for improving the accuracy of explicit Runge–Kutta methods while preserving their computational efficiency.

In principle, further improvements could be obtained by extending the optimization to the ninth-order residual conditions. However, this remains computationally demanding because of the large number of order conditions (485 equations) involved at order 9. Future work will therefore focus on developing efficient optimization techniques for higher-order explicit Runge–Kutta methods and on extending the proposed residual-based strategy to adaptive time-stepping schemes.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions, which have significantly improved the quality and clarity of this manuscript.



ANNEX PAGE

The 200 equations of order 8 with s stages

Order 5

$$\begin{aligned}
E_1 : \sum_{i=1}^s b_i &= 1 \\
E_2 : \sum_{i=1}^s b_i c_i &= \frac{1}{2} \\
E_3 : \sum_{i=1}^s b_i c_i^2 &= \frac{1}{3} \\
E_4 : \sum_{i,j=1}^s b_i a_{ij} c_j &= \frac{1}{6} \\
E_5 : \sum_{i=1}^s b_i c_i^3 &= \frac{1}{4} \\
E_6 : \sum_{i,j=1}^s b_i c_i a_{ij} c_j &= \frac{1}{8} \\
E_7 : \sum_{i,j=1}^s b_i a_{ij} c_j^2 &= \frac{1}{12} \\
E_8 : \sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k &= \frac{1}{24} \\
E_9 : \sum_{i=1}^s b_i c_i^4 &= \frac{1}{5} \\
E_{10} : \sum_{i,j=1}^s b_i c_i^2 a_{ij} c_j &= \frac{1}{10} \\
E_{11} : \sum_{i,j,k=1}^s b_i a_{ij} c_j a_{ik} c_k &= \frac{1}{20} \\
E_{12} : \sum_{i,j=1}^s b_i c_i a_{ij} c_j^2 &= \frac{1}{15} \\
E_{13} : \sum_{i,j,k=1}^s b_i c_i a_{ij} a_{jk} c_k &= \frac{1}{30} \\
E_{14} : \sum_{i,j=1}^s b_i a_{ij} c_j^3 &= \frac{1}{20} \\
E_{15} : \sum_{i,j,k=1}^s b_i a_{ij} c_j a_{jk} c_k &= \frac{1}{40} \\
E_{16} : \sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k^2 &= \frac{1}{60} \\
E_{17} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} a_{kl} c_l &= \frac{1}{120}
\end{aligned}$$

Order 6

$$\begin{aligned}
E_{18} : \sum_{i=1}^s b_i c_i^5 &= \frac{1}{6} \\
E_{19} : \sum_{i,j=1}^s b_i c_i^3 a_{ij} c_j &= \frac{1}{12} \\
E_{20} : \sum_{i,j=1}^s b_i c_i^2 a_{ij} c_j^2 &= \frac{1}{18} \\
E_{21} : \sum_{i,j,k=1}^s b_i c_i^2 a_{ij} a_{jk} c_k &= \frac{1}{36} \\
E_{22} : \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{ik} c_k &= \frac{1}{24} \\
E_{23} : \sum_{i,j=1}^s b_i c_i a_{ij} c_j^3 &= \frac{1}{24} \\
E_{24} : \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{jk} c_k &= \frac{1}{48} \\
E_{25} : \sum_{i,j,k=1}^s b_i c_i a_{ij} a_{jk} c_k^2 &= \frac{1}{72} \\
E_{26} : \sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} a_{kl} c_l &= \frac{1}{144} \\
E_{27} : \sum_{i,j,k=1}^s b_i a_{ij} c_j^2 a_{ik} c_k &= \frac{1}{36} \\
E_{28} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{ik} a_{kl} c_l &= \frac{1}{72} \\
E_{29} : \sum_{i,j=1}^s b_i a_{ij} c_j^4 &= \frac{1}{30} \\
E_{30} : \sum_{i,j,k=1}^s b_i a_{ij} c_j^2 a_{jk} c_k &= \frac{1}{60} \\
E_{31} : \sum_{i,j,k=1}^s b_i a_{ij} c_j a_{jk} c_k^2 &= \frac{1}{90} \\
E_{32} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} a_{kl} c_l &= \frac{1}{180} \\
E_{33} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k a_{jl} c_l &= \frac{1}{120} \\
E_{34} : \sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k^3 &= \frac{1}{120} \\
E_{35} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k a_{kl} c_l &= \frac{1}{240} \\
E_{36} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} a_{kl} c_l^2 &= \frac{1}{360} \\
E_{37} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m &= \frac{1}{720}
\end{aligned}$$

Order 7

$$\begin{aligned}
E_{38} : \sum_{i=1}^s b_i c_i^6 &= \frac{1}{7} \\
E_{39} : \sum_{i,j=1}^s b_i c_i^4 a_{ij} c_j &= \frac{1}{14} \\
E_{40} : \sum_{i,j=1}^s b_i c_i^3 a_{ij} c_j^2 &= \frac{1}{21} \\
E_{41} : \sum_{i,j,k=1}^s b_i c_i^3 a_{ij} a_{jk} c_k &= \frac{1}{42} \\
E_{42} : \sum_{i,j,k=1}^s b_i c_i^2 a_{ij} c_j a_{ik} c_k &= \frac{1}{28} \\
E_{43} : \sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j a_{ik} a_{kl} c_l &= \frac{1}{84} \\
E_{44} : \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{ik} c_k^2 &= \frac{1}{42} \\
E_{45} : \sum_{i,j=1}^s b_i c_i^2 a_{ij} c_j^3 &= \frac{1}{28} \\
E_{46} : \sum_{i,j,k=1}^s b_i c_i^2 a_{ij} c_j a_{jk} c_k &= \frac{1}{56} \\
E_{47} : \sum_{i,j,k=1}^s b_i c_i^2 a_{ij} a_{jk} c_k^2 &= \frac{1}{84} \\
E_{48} : \sum_{i,j,k,l=1}^s b_i c_i^2 a_{ij} a_{jk} a_{kl} c_l &= \frac{1}{168} \\
E_{49} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{ik} a_{il} c_j c_k c_l &= \frac{1}{56} \\
E_{50} : \sum_{i,j=1}^s b_i c_i a_{ij} c_j^4 &= \frac{1}{35} \\
E_{51} : \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j^2 a_{jk} c_k &= \frac{1}{70} \\
E_{52} : \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{jk} c_k^2 &= \frac{1}{105} \\
E_{53} : \sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j a_{jk} a_{kl} c_l &= \frac{1}{210} \\
E_{54} : \sum_{i,j,k=1}^s b_i c_i a_{ij} a_{jk} c_k^3 &= \frac{1}{140} \\
E_{55} : \sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} a_{kl} c_l &= \frac{1}{280} \\
E_{56} : \sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} a_{kl} c_l^2 &= \frac{1}{420} \\
E_{57} : \sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} a_{kl} a_{lm} c_m &= \frac{1}{840} \\
E_{58} : \sum_{i,j,k=1}^s b_i a_{ij} c_j^3 a_{ik} c_k &= \frac{1}{56} \\
E_{59} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} c_k a_{il} c_l &= \frac{1}{112} \\
E_{60} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k^2 a_{il} c_l &= \frac{1}{168} \\
E_{61} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} c_l a_{im} c_m &= \frac{1}{336} \\
E_{62} : \sum_{i,j,k=1}^s b_i a_{ij} a_{ik} c_j^2 c_k^2 &= \frac{1}{63} \\
E_{63} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{ik} a_{kl} c_l &= \frac{1}{126} \\
E_{64} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{il} a_{lm} c_m &= \frac{1}{252} \\
E_{65} : \sum_{i,j=1}^s b_i a_{ij} c_j^5 &= \frac{1}{42} \\
E_{66} : \sum_{i,j,k=1}^s b_i a_{ij} c_j^3 a_{jk} c_k &= \frac{1}{84} \\
E_{67} : \sum_{i,j,k=1}^s b_i a_{ij} c_j^2 a_{jk} c_k^2 &= \frac{1}{126} \\
E_{68} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{jk} a_{kl} c_l &= \frac{1}{252} \\
E_{69} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} a_{jl} c_k c_l &= \frac{1}{168} \\
E_{70} : \sum_{i,j,k=1}^s b_i a_{ij} c_j a_{jk} c_k^3 &= \frac{1}{168} \\
E_{71} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} c_k a_{kl} c_l &= \frac{1}{336} \\
E_{72} : \sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} a_{kl} c_l^2 &= \frac{1}{504} \\
E_{73} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{jk} a_{kl} a_{lm} c_m &= \frac{1}{1008} \\
E_{74} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k a_{jl} c_l^2 &= \frac{1}{252} \\
E_{75} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{jl} a_{lm} c_m &= \frac{1}{504}
\end{aligned}$$



$$\begin{aligned}
E_{77} &: \sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k^4 = \frac{1}{210} \\
E_{78} &: \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k^2 a_{kl} c_l = \frac{1}{420} \\
E_{79} &: \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k a_{kl} c_l^2 = \frac{1}{630} \\
E_{80} &: \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{kl} a_{lm} c_m = \frac{1}{1260} \\
E_{81} &: \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} a_{km} c_l c_m = \frac{1}{840} \\
E_{82} &: \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} a_{kl} c_l^3 = \frac{1}{840} \\
E_{83} &: \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} c_l a_{lm} c_m = \frac{1}{1680} \\
E_{84} &: \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m^2 = \frac{1}{2520} \\
E_{85} &: \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} c_n = \frac{1}{5040}
\end{aligned}$$

Order 8

$$\begin{aligned}
E_{86} &: \sum_{i=1}^s b_i c_i^7 = \frac{1}{8} \\
E_{87} &: \sum_{i,j=1}^s b_i c_i^5 a_{ij} c_j = \frac{1}{16} \\
E_{88} &: \sum_{i,j=1}^s b_i c_i^4 a_{ij} c_j^2 = \frac{1}{24} \\
E_{89} &: \sum_{i,j,k=1}^s b_i c_i^4 a_{ij} a_{jk} c_k = \frac{1}{48} \\
E_{90} &: \sum_{i,j,k=1}^s b_i c_i^3 a_{ij} a_{ik} c_j c_k = \frac{1}{32} \\
E_{91} &: \sum_{i,j=1}^s b_i c_i^3 a_{ij} c_j^3 = \frac{1}{32} \\
E_{92} &: \sum_{i,j,k=1}^s b_i c_i^3 a_{ij} c_j a_{jk} c_k = \frac{1}{64} \\
E_{93} &: \sum_{i,j,k=1}^s b_i c_i^3 a_{ij} a_{jk} c_k^2 = \frac{1}{96} \\
E_{94} &: \sum_{i,j,k,l=1}^s b_i c_i^3 a_{ij} a_{jk} a_{kl} c_l = \frac{1}{192} \\
E_{95} &: \sum_{i,j,k=1}^s b_i c_i^2 a_{ij} c_j a_{ik} c_k^2 = \frac{1}{48} \\
E_{96} &: \sum_{i,j,k,l=1}^s b_i c_i^2 a_{ij} c_j a_{ik} a_{kl} c_l = \frac{1}{96} \\
E_{97} &: \sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{ik} a_{il} c_j c_k c_l = \frac{1}{64} \\
E_{98} &: \sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{ik} a_{il} c_k c_l = \frac{1}{96} \\
E_{99} &: \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{ik} c_j c_k a_{il} a_{lm} c_m = \frac{1}{192} \\
E_{100} &: \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{ik} c_k^3 = \frac{1}{64} \\
E_{101} &: \sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{ik} c_k a_{kl} c_l = \frac{1}{128} \\
E_{102} &: \sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j a_{ik} a_{kl} c_l^2 = \frac{1}{192} \\
E_{103} &: \sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} c_j a_{ik} a_{kl} a_{lm} c_m = \frac{1}{384} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j^2 a_{ik} a_{kl} c_l &= \frac{1}{144} \\
\sum_{i,j,k=1}^s b_i c_i a_{ij} c_j^2 a_{ik} c_k^2 &= \frac{1}{72} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} c_k a_{il} a_{lm} c_m &= \frac{1}{288} \\
\sum_{i,j=1}^s b_i c_i^2 a_{ij} c_j^4 &= \frac{1}{40} \\
\sum_{i,j,k=1}^s b_i c_i^2 a_{ij} c_j^2 a_{jk} c_k &= \frac{1}{80} \\
\sum_{i,j,k,l=1}^s b_i c_i^2 a_{ij} a_{jk} a_{jl} c_k c_l &= \frac{1}{160} \\
\sum_{i,j,k=1}^s b_i c_i^2 a_{ij} c_j a_{jk} c_k^2 &= \frac{1}{120} \\
\sum_{i,j,k,l=1}^s b_i c_i^2 a_{ij} c_j a_{jk} a_{kl} c_l &= \frac{1}{240} \\
\sum_{i,j,k=1}^s b_i c_i^2 a_{ij} a_{jk} c_k^3 &= \frac{1}{160} \\
\sum_{i,j,k,l=1}^s b_i c_i^2 a_{ij} a_{jk} c_k a_{kl} c_l &= \frac{1}{320} \\
\sum_{i,j,k,l=1}^s b_i c_i^2 a_{ij} a_{jk} a_{kl} c_l^2 &= \frac{1}{480} \\
\sum_{i,j,k,l,m=1}^s b_i c_i^2 a_{ij} a_{jk} a_{kl} a_{lm} c_m &= \frac{1}{960} \\
\sum_{i,j=1}^s b_i c_i a_{ij} c_j^5 &= \frac{1}{48} \\
\sum_{i,j,k=1}^s b_i c_i a_{ij} c_j^3 a_{jk} c_k &= \frac{1}{96}
\end{aligned}$$

$$\begin{aligned}
\sum_{i,j,k=1}^s b_i c_i a_{ij} c_j^2 a_{jk} c_k^2 &= \frac{1}{144} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j^2 a_{jk} a_{kl} c_l &= \frac{1}{288} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j a_{jk} a_{jl} c_k c_l &= \frac{1}{192} \\
\sum_{i,j,k=1}^s b_i c_i a_{ij} c_j a_{jk} c_k^3 &= \frac{1}{192} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j a_{jk} c_k a_{kl} c_l &= \frac{1}{384} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} c_j a_{jk} a_{kl} c_l^2 &= \frac{1}{576} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} c_j a_{jk} a_{kl} a_{lm} c_m &= \frac{1}{1152} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} c_k^2 a_{jl} c_l &= \frac{1}{288} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} c_k a_{jl} a_{lm} c_m &= \frac{1}{576} \\
\sum_{i,j,k=1}^s b_i c_i a_{ij} a_{jk} c_k^4 &= \frac{1}{240} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} c_k^2 a_{kl} c_l &= \frac{1}{480} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} c_k a_{kl} c_l^2 &= \frac{1}{720} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} c_k a_{kl} a_{lm} c_m &= \frac{1}{1440} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} a_{kl} a_{km} c_l c_m &= \frac{1}{960} \\
\sum_{i,j,k,l=1}^s b_i c_i a_{ij} a_{jk} a_{kl} c_l^3 &= \frac{1}{960} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} a_{kl} c_l a_{lm} c_m &= \frac{1}{1920} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} a_{kl} a_{lm} c_m^2 &= \frac{1}{2880} \\
\sum_{i,j,k,l,m=1}^s b_i c_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} c_n &= \frac{1}{5760} \\
\sum_{i,j,k=1}^s b_i a_{ij} c_j a_{ik} c_k^4 &= \frac{1}{80} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{ik} c_k^2 a_{kl} c_l &= \frac{1}{160} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{ik} a_{kl} a_{km} c_l c_m &= \frac{1}{320} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{ik} c_k a_{kl} c_l^2 &= \frac{1}{240} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{ik} c_k a_{kl} a_{lm} c_m &= \frac{1}{480} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{ik} a_{kl} c_l^3 &= \frac{1}{320} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{ik} a_{kl} c_l a_{lm} c_m &= \frac{1}{640} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{ik} a_{kl} a_{lm} c_m^2 &= \frac{1}{960} \\
\sum_{i,j,k,l,m,n=1}^s b_i a_{ij} c_j a_{ik} a_{kl} a_{lm} a_{mn} c_n &= \frac{1}{1920} \\
\sum_{i,j,k=1}^s b_i a_{ij} c_j^3 a_{ik} c_k^2 &= \frac{1}{96} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^3 a_{ik} a_{kl} c_l &= \frac{1}{192} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{ik} c_k a_{kl} c_l &= \frac{1}{192} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{il} c_l a_{lm} c_m &= \frac{1}{384} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{ik} a_{kl} c_l^2 &= \frac{1}{288} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{il} a_{lm} c_m^2 &= \frac{1}{576} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j^2 a_{ik} a_{kl} a_{lm} c_m &= \frac{1}{576} \\
\sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} c_k a_{il} a_{lm} a_{mn} c_n &= \frac{1}{1152} \\
\sum_{i,j=1}^s b_i a_{ij} c_i^6 &= \frac{1}{56} \\
\sum_{i,j,k=1}^s b_i a_{ij} c_j^4 a_{jk} c_k &= \frac{1}{112} \\
\sum_{i,j,k=1}^s b_i a_{ij} c_j^3 a_{jk} c_k^2 &= \frac{1}{168} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^3 a_{jk} a_{kl} c_l &= \frac{1}{336} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{jk} c_l a_{jl} c_k &= \frac{1}{224} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{jk} c_k a_{jl} a_{lm} c_m &= \frac{1}{672} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} c_k a_{jl} c_l^2 &= \frac{1}{336} \\
\sum_{i,j,k=1}^s b_i a_{ij} c_j^2 a_{jk} c_k^3 &= \frac{1}{224}
\end{aligned}$$

$$\begin{aligned}
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{jk} c_k a_{kl} c_l &= \frac{1}{448} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j^2 a_{jk} a_{kl} c_l^2 &= \frac{1}{672} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j^2 a_{jk} a_{kl} a_{lm} c_m &= \frac{1}{1344} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{jl} a_{jm} c_k c_l c_m &= \frac{1}{448} \\
\sum_{i,j,k=1}^s b_i a_{ij} c_j a_{jk} c_k^4 &= \frac{1}{280} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} c_k^2 a_{kl} c_l &= \frac{1}{560} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} a_{kl} a_{km} c_m c_l &= \frac{1}{1120} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} c_k a_{kl} c_l^2 &= \frac{1}{840} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{jk} c_k a_{kl} a_{lm} c_m &= \frac{1}{1680} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} c_j a_{jk} a_{kl} c_l^3 &= \frac{1}{1120} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{jk} a_{kl} c_l a_{lm} c_m &= \frac{1}{2240} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} c_j a_{jk} a_{kl} a_{lm} c_m^2 &= \frac{1}{3360} \\
\sum_{i,j,k,l,m,n=1}^s b_i a_{ij} c_j a_{jk} a_{kl} a_{lm} a_{mn} c_n &= \frac{1}{6720} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k a_{jl} c_l^3 &= \frac{1}{448} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{jl} c_l a_{lm} c_m &= \frac{1}{896} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{jl} a_{lm} c_m^2 &= \frac{1}{1344} \\
\sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} c_k a_{jl} a_{lm} a_{mn} c_n &= \frac{1}{2688} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k^2 a_{jl} c_l^2 &= \frac{1}{504} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k^2 a_{jl} a_{lm} c_m &= \frac{1}{1008} \\
\sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} a_{kl} c_l a_{jm} a_{mn} c_n &= \frac{1}{2016}
\end{aligned}$$

$$\begin{aligned}
\sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k^5 &= \frac{1}{336} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k^3 a_{kl} c_l &= \frac{1}{672} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k^2 a_{kl} c_l^2 &= \frac{1}{1008} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k^2 a_{kl} a_{lm} c_m &= \frac{1}{2016} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{kl} a_{km} c_m c_l &= \frac{1}{1344} \\
\sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} c_k a_{kl} c_l^3 &= \frac{1}{1344} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{kl} c_l a_{lm} c_m &= \frac{1}{2688} \\
\sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} c_k a_{kl} a_{lm} c_m^2 &= \frac{1}{4032} \\
\sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} c_k a_{kl} a_{lm} a_{mn} c_n &= \frac{1}{8064} \\
E_{190} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} c_l^2 a_{km} c_m &= \frac{1}{2016} \\
E_{191} : \sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} a_{kl} c_l a_{km} a_{mn} c_m &= \frac{1}{4032} \\
E_{192} : \sum_{i,j,k,l=1}^s b_i a_{ij} a_{jk} a_{kl} c_l^4 &= \frac{1}{1680} \\
E_{193} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} c_l^2 a_{lm} c_m &= \frac{1}{3360} \\
E_{194} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} c_l^2 c_m &= \frac{1}{5040} \\
E_{195} : \sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} c_n c_l &= \frac{1}{10080} \\
E_{196} : \sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{ln} c_n c_m &= \frac{1}{6720} \\
E_{197} : \sum_{i,j,k,l,m=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m^3 &= \frac{1}{6720} \\
E_{198} : \sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} c_n c_m &= \frac{1}{13440} \\
E_{199} : \sum_{i,j,k,l,m,n=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} c_n^2 &= \frac{1}{20160} \\
E_{200} : \sum_{i,j,k,l,m,n,o=1}^s b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} a_{no} c_o &= \frac{1}{40320}
\end{aligned}$$

REFERENCES

- [1] K. A. M. Akpini, K. R. Assui, G. Yoro, and B. Bailly, *Optimal Method of Runge–Kutta of Order 5*, Journal of Mathematics Research, 11(1) (2019), 93–101.
- [2] A. Başhan, S. B. G. Karakoç, and T. Geyikli, *Approximation of the KdVB equation by the quintic B-spline differential quadrature method*, Kuwait Journal of Science, 42(2) (2015), 67–92.
- [3] A. Başhan, N. M. Yağmurlu, Y. Uçar, and A. Esen, *An effective approach to numerical soliton solutions for the Schrödinger equation via modified cubic B-spline differential quadrature method*, Chaos, Solitons & Fractals, 100 (2017), 45–56.
- [4] A. Başhan, N. M. Yağmurlu, and Y. U. A. Esen, *Numerical solutions for the fourth order extended Fisher–Kolmogorov equation with high accuracy by differential quadrature method*, Sigma Journal of Engineering and Natural Sciences, 9(3) (2018), 273–284.
- [5] A. Başhan, *An effective application of differential quadrature method based on modified cubic B-splines to numerical solutions of the KdV equation*, Turkish Journal of Mathematics, 42(1) (2018), 373–393.
- [6] E. Bronasco, J. Fregin, D. Ruprecht, and G. Vilmart, *Spectral Deferred Corrections in the Framework of Runge–Kutta Methods*, arXiv preprint arXiv:2604.03013 (2026).
- [7] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd ed., John Wiley & Sons, New York, 2008.
- [8] J. C. Butcher, *An algebraic theory of integration methods*, Mathematics of Computation, 26(117) (1972), 79–106.
- [9] J. R. Dormand and P. J. Prince, *A family of embedded Runge–Kutta formulae*, Journal of Computational and Applied Mathematics, 6(1) (1980), 19–26.
- [10] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, John Wiley & Sons, New York, 2009.
- [11] S. Khashin, *List of some known Runge–Kutta methods family*, Available at: http://math.ivanovo.ac.ru/dalgebra/Khashin/rk/sh_rk.html
- [12] R. Mahnken, *New low order Runge–Kutta schemes for asymptotically exact global error estimation of embedded methods without order reduction*, Computer Methods in Applied Mechanics and Engineering, 401 (2022), 115553.



- [13] B. Martens, *Error Estimates for Runge–Kutta Schemes of Optimal Control Problems with Index 1 DAEs*, Computational Optimization and Applications, 86(3) (2023), 1–27.
- [14] H. Séka and K. R. Assui, *A New Eighth Order Runge–Kutta Family Method*, Journal of Mathematics Research, 11(2) (2019), 190–199.
- [15] Y. Uçar, N. M. Yağmurlu, and A. Başhan, *Numerical solutions and stability analysis of modified Burgers equation via modified cubic B-spline differential quadrature methods*, Sigma Journal of Engineering and Natural Sciences, 37(1) (2019), 129–142.

Uncorrected Proof

