



Article

# A Neural Network Technique for the Derivation of Runge–Kutta Pairs Adjusted for Scalar Autonomous Problems

Vladislav N. Kovalnogov <sup>1</sup>, Ruslan V. Fedorov <sup>1</sup>, Yuri A. Khakhalev <sup>1</sup>, Theodore E. Simos <sup>1,2,3,4,5,6,\*</sup>   
and Charalampos Tsitouras <sup>7,8</sup> 

- <sup>1</sup> Laboratory of Inter-Disciplinary Problems of Energy Production, Ulyanovsk State Technical University, 32 Severny Venetz Street, 432027 Ulyanovsk, Russia; kvn@ulstu.ru (V.N.K.); r.fedorov@ulstu.ru (R.V.F.); ulstu-td-ua@mail.ru (Y.A.K.)
- <sup>2</sup> College of Applied Mathematics, Chengdu University of Information Technology, Chengdu 610225, China
- <sup>3</sup> Department of Medical Research, China Medical University Hospital, China Medical University, Taichung City 40402, Taiwan
- <sup>4</sup> Data Recovery Key Laboratory of Sichuan Province, Neijiang Normal University, Neijiang 641100, China
- <sup>5</sup> Department of Civil Engineering, Section of Mathematics, Democritus University of Thrace, 67100 Xanthi, Greece
- <sup>6</sup> Department of Mathematics, University of Western Macedonia, 50100 Kozani, Greece
- <sup>7</sup> General Department, National & Kapodistrian University of Athens, GR34400 Euripus Campus, Greece; tsitourasc@uoa.gr
- <sup>8</sup> Administration of Businesses and Organizations Department, Hellenic Open University, 26335 Patras, Greece
- \* Correspondence: tsimos.conf@gmail.com
- † Correspondence Address: T. E. Simos, 10 Konitsis Street, 17564 Athens, Greece.



**Citation:** Kovalnogov, V.N.; Fedorov, R.V.; Khakhalev, Y.A.; Simos, T.E.; Tsitouras, C. A Neural Network Technique for the Derivation of Runge–Kutta Pairs Adjusted for Scalar Autonomous Problems. *Mathematics* **2021**, *9*, 1842. <https://doi.org/10.3390/math9161842>

Academic Editor: Anton Iliev

Received: 23 June 2021

Accepted: 2 August 2021

Published: 4 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** We consider the scalar autonomous initial value problem as solved by an explicit Runge–Kutta pair of orders 6 and 5. We focus on an efficient family of such pairs, which were studied extensively in previous decades. This family comes with 5 coefficients that one is able to select arbitrarily. We set, as a fitness function, a certain measure, which is evaluated after running the pair in a couple of relevant problems. Thus, we may adjust the coefficients of the pair, minimizing this fitness function using the differential evolution technique. We conclude with a method (i.e. a Runge–Kutta pair) which outperforms other pairs of the same two orders in a variety of scalar autonomous problems.

**Keywords:** initial value problem; scalar autonomous; Runge–Kutta; differential evolution; functionally fitted methods

**MSC:** 65L05; 65L06; 90C26; 90C30

## 1. Introduction

The Initial Value Problem (IVP) is given as

$$\begin{aligned}x' &= f(t, x), \\x(t_0) &= x_0\end{aligned}\tag{1}$$

with  $t, t_0 \in \mathbb{R}$ ,  $x, x' \in \mathbb{R}^m$  and  $f: \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ .

Amongst the most celebrated numerical methods for dealing with (1) are Runge–Kutta (RK) pairs. The following Butcher tableau [1,2] characterizes these methods.

$$\begin{array}{c|c}a & B \\ \hline & c \\ & \hat{c}\end{array}$$

with  $c^T, \hat{c}^T, a \in \mathbb{R}^s$  and  $B \in \mathbb{R}^{s \times s}$ . In the above case, the pair shares  $s$  stages that are evaluated explicitly when  $B$  is strictly lower triangular. The approximation of the solution, forwards from  $(t_n, x_n)$  to  $t_{n+1} = t_n + h_n$  is furnished by two estimations of  $x(t_{n+1})$ . These are  $x_{n+1}$  and  $\hat{x}_{n+1}$ , which are given by

$$x_{n+1} = x_n + h_n \cdot \sum_{i=1}^{i=s} c_i k_i$$

with

$$\hat{x}_{n+1} = x_n + h_n \cdot \sum_{i=1}^{i=s} \hat{c}_i k_i$$

and

$$k_i = f(t_n + a_i h_n, x_n + h_n \cdot \sum_{j=1}^{j=i-1} b_{ij} k_j),$$

for  $i = 1, 2, \dots, s$ . The approximations  $x_{n+1}$  and  $\hat{x}_{n+1}$  are of algebraic orders  $p$  and  $q < p$ , respectively. Thus, a local error estimation

$$\epsilon_n = h_n^{p-q-1} \cdot \|x_{n+1} - \hat{x}_{n+1}\|,$$

is formed in every step. This helps in forming the following step changing algorithm

$$h_{n+1} = d \cdot h_n \cdot \left(\frac{\sigma}{\epsilon_n}\right)^{1/p},$$

where  $\sigma$  is a tolerance, chosen by the user, and  $d = 0.9$  is a safety factor. In the case that  $\epsilon_n < \sigma$ , the above formula is used as the next step-length. On the contrary, we also use this formula, but the approximate solution is not forwarded and  $h_{n+1}$  is used as new version of  $h_n$ . Details of this issue can be retrieved from [3] or even [4] (pg. 167–168). These methods are usually abbreviated as RKp(q) pairs.

Runge–Kutta methods first appeared in the late nineteenth century [5,6], while pairs were introduced after 1960. The first-celebrated such pairs, of orders 5(4), 6(5) and 8(7), were presented by Fehlberg [7,8]. Then, in the early 1980’s, Dormand and Prince followed [9,10]. In addition, our research group has derived a number of such pairs [11–14].

Runge–Kutta pairs are well suited for efficiently approximating the solution of non-stiff problems of the form in (1). Such problems arise, for example, when creating digital twins of clean-energy production technologies, for which functionally fitted finite element methods, with one-step and multi-step forms and improved dispersive and dissipative properties, have become critically important [15]. The precision demanded explains the wide diversity of pairs. As a result, the lower the accuracy on demand, the more efficient the lower RK pairs are. In reverse, a higher-order pair should be used for stringent accuracies at quadruple precision. [16]. The effort to construct better pairs is the subject of current research [17,18].

Here, we focus on RK6(5) pairs, which are used for high to modest accuracies. We are especially interested in problems (1) of the form

$$x' = f(x), \quad x(t_0) = x_0, \tag{2}$$

with  $f : \mathbb{R} \rightarrow \mathbb{R}$ . These problems are called scalar autonomous, and we will derive a particular RK6(5) pair tuned specially for this type of problem.

The paper is organized into sections as follows:

1. Introduction;
2. Theory of Runge–Kutta Pairs of Orders 6(5);
3. Training the coefficients;
4. Numerical Tests;

5. Conclusion.

2. Theory of Runge–Kutta Pairs of Orders 6(5)

Runge–Kutta pairs of orders six and five are, almost, the most-used ones. Their coefficients have to satisfy 54 order conditions, and families of solutions have been discovered through the years. We have selected the Verner-DLMP [19,20] family, which has the advantage of being solved linearly. This is an FSAL (First Stage As Last) family. Although  $s = 9$ , the pairs use only eight stages each step, since the last (the 9th) stage is used as the first stage of the next step.

Then we choose freely the coefficients  $a_2, a_4, a_5, a_6, a_7$  and  $\hat{c}_9$ . This family’s pairs have been shown to perform the best in a variety of problems [21].

We may proceed by evaluating explicitly the remaining coefficients.

In the following algorithm  $a^i$  is the vector with the components of  $a$  raised to the  $i$ -th power and  $e = [1, 1, \dots, 1]^T \in \mathbb{R}^s$ .  $(B \cdot a)_5$  is the 5-th component of  $B \cdot a$ . See [22,23] for more details.  $A = \text{diag}(a)$  and  $I$  is the identity matrix of proper dimension.

ALGORITHM: The free parameters are  $a_2, a_4, a_5, a_6, a_7, \hat{c}_9$ . It is known that for this family  $c_2 = c_3 = \hat{c}_2 = \hat{c}_3 = 0$  and  $b_{i2} = 0, i = 4, 5, 6, 7, 8$ . Consecutively execute the following instructions.

1. Solve  $c \cdot e = 1, c \cdot a = \frac{1}{2}, c \cdot a^2 = \frac{1}{3}, c \cdot a^3 = \frac{1}{4}, c \cdot a^4 = \frac{1}{5}, c \cdot a^5 = \frac{1}{6}$ , for  $c_2, c_4, c_5, c_6, c_7, c_8$ .
2. Put  $a_3 = \frac{2}{3c_4}, b_{43} = \frac{a_4^2}{2a_3}, b_{32} = \frac{a_3^2}{2a_2}$ .
3. Solve  $(b \cdot a)_5 = \frac{a_5^2}{2}$  and  $(b \cdot a^2)_5 = \frac{a_5^3}{3}$  for  $b_{53}, b_{54}$
4. Substitute  $b_{87}$  from  $(c \cdot (A + B - I))_7 = 0$ .
5. Since  $(c \cdot (A - I) \cdot B)_3 = 0$  find  $b_{76}$  from

$$c \cdot (A - I) \cdot B \cdot (A - a_4I) \cdot (A - a_5I) \cdot a - (c(A - I)B)_3 = \int_0^1 (x - 1) \int_0^x (y - c_4)(y - c_5)ydydx.$$

6.  $b_{86}$  is given from  $(c \cdot (B + A - I))_6 = 0$
7. Solve simultaneously for  $\hat{c}_1, \hat{c}_4, \hat{c}_5, \hat{c}_6, \hat{c}_7, \hat{c}_8, b_{63}, b_{73}$ , and  $b_{83}$  the equations:  
 $\hat{c} \cdot e = 1, \hat{c} \cdot a = \frac{1}{2}, \hat{c} \cdot a^2 = \frac{1}{3}, \hat{c} \cdot a^3 = \frac{1}{4}, \hat{c} \cdot a^4 = \frac{1}{5}, (c \cdot (B + A - I))_3 = 0, (\hat{c} \cdot B)_3 = 0,$

$$c \cdot (A - I) \cdot B \cdot (A - a_4I) \cdot (A - a_5I) \cdot a = \int_0^1 (x - 1) \int_0^x (y - a_5)(y - a_4)ydydx,$$

$$\hat{c} \cdot B \cdot (A - a_5I) \cdot (A - a_4I) \cdot a = \int_0^1 \int_0^x (y - a_5)(y - a_4)ydydx.$$

8. From  $(B \cdot a)_6 = \frac{a_6^2}{2}, (B \cdot a^2)_6 = \frac{a_6^3}{3}$  evaluate  $b_{64}$  and  $b_{65}$ .
9. From  $(B \cdot a)_7 = \frac{c_7^2}{2}, (B \cdot a^2)_7 = \frac{c_7^3}{3}$  evaluate  $b_{74}$  and  $b_{75}$ .
10. From  $(B \cdot a)_8 = \frac{c_8^2}{2}, (B \cdot a^2)_8 = \frac{c_8^3}{3}$  evaluate  $b_{84}$  and  $b_{85}$ .
11. From  $B \cdot e = a$  evaluate  $b_{21}, b_{31}, \dots, b_{81}$ .
12. Finally, using FSAL (First Stage As Last) property, substitute  $b_{9j} = c_j, j = 1, 2, \dots, 8$ .

All equations are solved explicitly and straightforwardly. No back substitutions or implicit equations are present.

The question now raised is how to select the free parameters. Traditionally, we attempt to reduce the norm of the principal term of the local truncation error, i.e., the coefficients of  $h^7$  in the residual of Taylor error expansions corresponding to the sixth order method of the underlying RK pair [11].

Another choice is examined in [24], where we dealt with a class of seven stages, as well as FSAL pairs of orders six and four, that are specially tuned for addressing the problems of interest here. We presented the reduced set of order conditions for the case of interest; then we solved it in order to furnish a certain pair ST6(4). We didn't considered this case, since there are only two free parameters that are non-linearly dependent and it is not expected to make serious improvement. However, we will include this pair in our numerical tests.

### 3. Training the Coefficients

Here, our approach is to train the coefficients of a method. In this view, we say that we are utilizing a Neural Network technique. We consider, as input, the free parameters of the Runge–Kutta pair. Then, the steps taken for solving an Initial Value Problem can be seen as internal layers, while the output is a particular efficiency measure of the results.

We intend to derive a particular RK6(5) pair belonging to the studied family above. The resulting pair has to perform best on scalar autonomous problems (2). First, let us say that some pair was run in a certain problem (2) for some tolerance.

We then record the number  $\mu$  of function evaluations (stages) needed and the global error  $\varepsilon$  observed over the whole mesh (grid-points) in the interval of integration. Then, we form the efficiency measure

$$r = \mu \cdot \varepsilon^{1/6}. \quad (3)$$

Here we choose the following couple of scalar autonomous problems.

$$\text{1st problem : } x' = e^{-x}, x(0) = 1, t \in [0, 20],$$

with theoretical solution  $x(t) = \log(e + t)$ , and

$$\text{2nd problem : } x' = x^{1/3}, x(0) = 1, t \in [0, 20],$$

with theoretical solution  $x(t) = \frac{2}{3} \sqrt{\frac{2}{3}} \cdot x^{3/2}$ .

After using as tolerance  $\sigma = 10^{-11}$ , we ran DLMP6(5) for the above problems. We recorded

$$\mu_1 = 369, \varepsilon_1 \approx 1.9 \cdot 10^{-12}, {}^1r_{\text{DLMP65}} = 4.09,$$

for the first problem and

$$\mu_2 = 433, \varepsilon_2 \approx 7.6 \cdot 10^{-11}, {}^2r_{\text{DLMP65}} = 8.92,$$

for the second problem.

Let us suppose that any new pair NEW6(5) furnishes corresponding efficiency measures  ${}^1r_{\text{NEW65}}$  and  ${}^2r_{\text{NEW65}}$  for the same runs. We then form, as a fitness function, the sum

$$\hat{r} = \frac{{}^1r_{\text{DLMP65}}}{{}^1r_{\text{NEW65}}} + \frac{{}^2r_{\text{DLMP65}}}{{}^2r_{\text{NEW65}}}, \quad (4)$$

and try to maximize it. That is, the compound fitness function is actually two whole runs of Initial Value Problems. The value  $\hat{r}$  changes according to the selection of the free parameters  $a_2, a_4, a_5, a_6, a_7$  and  $\hat{c}_9$ .

The original idea is based on [25]. For the minimization of  $\hat{r}$  we used the differential evolution technique [26]. We have already tried this approach and previously acquired some interesting results when producing Numerov-type methods for integrating orbits [27]. In that work, we trained the coefficients of a Numerov-type method on a Kepler orbit. We then observed excellent results over a set of Kepler orbits, as well other known orbital problems.

Software [28] was used for implementing our approach. As the objective (i.e. fitness) function we added (4). Actually,  ${}^1r_{\text{DLMP65}}$  and  ${}^2r_{\text{DLMP65}}$  were found in advance;  ${}^1r_{\text{NEW65}}$  and  ${}^2r_{\text{NEW65}}$  were adjusted according to the selection of the free parameters.

Finally, we derived an optimal method, named hereafter the NEW6(5) pair, sharing the following free parameters

$$\begin{aligned}
 a_2 &= 0.010190841992960, a_4 = 0.119497020307147, \\
 a_5 &= 0.4156202137620401, a_6 = 0.574431750193581, \\
 a_7 &= 0.802904404563573, \hat{c}_9 = 0.010038977481306
 \end{aligned}$$

We ran this pair for tolerance  $\sigma = 10^{-11}$ , and we recorded,

$$\mu_1 = 305, \varepsilon_1 \approx 4.4 \cdot 10^{-16}, {}^1r_{\text{NEW65}} = 0.84,$$

for the first problem and

$$\mu_2 = 297, \varepsilon_2 \approx 8.5 \cdot 10^{-14}, {}^2r_{\text{NEW65}} = 1.97,$$

for the second problem; i.e.,

$$\hat{r} = \frac{4.09}{0.84} + \frac{8.92}{1.97} = 4.86 + 4.53.$$

The above means that DLMP6(5) is about 386% and 353% more expensive than NEW6(5) for the problems under consideration, respectively (for  $\sigma = 10^{-11}$  in the specified interval). In reverse, NEW6(5) furnishes for the same costs

$$\log_{10} 4.86^6 \approx 4.12, \text{ and } \log_{10} 4.53^6 \approx 3.94,$$

more digits of accuracy, respectively.

The resulting pair is presented in Table 1.

**Table 1.** Coefficients of the here-proposed NEW6(5) pair, accurate for double precision computations.

$a_2 = 0.010190841992960,$	$a_3 = 0.079664680204765,$	$a_4 = 0.119497020307147,$
$a_5 = 0.4156202137620401,$	$a_6 = 0.574431750193581,$	$a_7 = 0.802904404563573,$
$a_8 = a_9 = 1,$	$c_1 = 0.0271498589320027,$	$c_2 = c_3 = 0,$
$c_4 = 0.219287409614054,$	$c_5 = 0.3291830326685719,$	$c_6 = 0.0671726795393684,$
$c_7 = 0.2983955751678166,$	$c_8 = 0.058811444078187,$	$c_9 = 0,$
$\hat{c}_1 = 0.04405860112075145,$	$\hat{c}_2 = \hat{c}_3 = 0,$	$\hat{c}_4 = 0.179304147231661,$
$\hat{c}_5 = 0.4167401045500195,$	$\hat{c}_6 = -0.028810314749226,$	$\hat{c}_7 = 0.338870205765215,$
$\hat{c}_8 = 0.039798278600273,$	$\hat{c}_9 = 0.010038977481306,$	
$b_{21} = 0.010190841992960,$	$b_{31} = -0.231715933708755,$	$b_{32} = 0.311380613913519,$
$b_{41} = 0.029874255076787,$	$b_{42} = 0,$	$b_{43} = 0.089622765230360,$
$b_{51} = 1.122557183457524,$	$b_{52} = 0,$	$b_{53} = -4.289151529341307,$
$b_{54} = 3.582214559645822,$	$b_{61} = -1.943165983475119,$	$b_{62} = 0,$
$b_{63} = 7.4677564003897048,$	$b_{64} = -5.495873107952286,$	$b_{65} = 0.545714441231281,$
$b_{71} = -2.803379493731238,$	$b_{72} = 0,$	$b_{73} = 10.105223718871366,$
$b_{74} = -7.165351732976296,$	$b_{75} = 0.058381490301930,$	$b_{76} = 0.6080304220978117,$
$b_{81} = 10.510126812245035,$	$b_{82} = 0,$	$b_{83} = -36.1276707396443543,$
$b_{84} = 25.865046568980085,$	$b_{85} = 2.3514136197972213,$	$b_{86} = -2.598933426151360,$
$b_{87} = 1.000017164773373,$	$b_{9j} = c_j, j = 1, 2, \dots, 8.$	

The norm of the principal truncation error coefficients is  $\|T^{(7)}\|_2 \approx 1.00 \cdot 10^{-4}$ , which is much greater than the corresponding value  $\|T^{(7)}\|_2 \approx 3.91 \cdot 10^{-5}$  for DLMP6(5). The interval of absolute stability is  $(-4.7, 0)$ , which is of the same magnitude as that of DLMP6(5), being  $(-4.2, 0)$ . We also mention that the corresponding principal truncation error norm for ST6(4) is  $\|T^{(7)}\|_2 \approx 7.69 \cdot 10^{-4}$ , while this pair shares a small stability interval of  $(-3.3, 0)$ . All truncation errors were measured in the reduced set valid for problems of the form (2); see [24].

In conclusion, no extra property seems to hold. The pair given in Table 1 does not provide anything interesting. No extended interval of stability is observed, no minimal truncation error exists, nor anything else. It is hard to believe its special performance after seeing its traditional characteristics.

**4. Numerical Tests**

We tested the following pairs chosen from the family studied above.

1. DLMP6(5) 9-stages FSAL pair given in [19].
2. ST6(4) 7-stages FSAL pair given in [24].
3. NEW6(5) 9-stages FSAL presented here.

All the pairs were run for tolerances  $10^{-6}, 10^{-7}, \dots, 10^{-11}$ , in the interval  $[0, 20]$ , except the last one which was run in the interval  $[\frac{\pi}{6}, \frac{\pi}{3}]$ . The efficiency measures (3) were recorded.

The problems we tested are listed in Table 2. We can verify that some of them describe real life problems, e.g., problem 1 demonstrates radioactivity decay. These problems cover various cases. Thus, we have included problems with slowly varying solutions, e.g., see problems 4 and 5. On the contrary, problems 7 and 8 share solutions that are constantly and clearly increasing. Problems with periodic solutions also exist, such as problem 9.

**Table 2.** Problems tested.

	<b>Problem</b>	<b>Solution</b>
1	$x' = -x, x(0) = 1$	$x(t) = e^{-t}$
2	$x' = \cos x, x(0) = 0$	$x(t) = 2 \arctan(\tanh(t/2))$
3	$x' = -x(1 - x/20)/4, x(0) = 1$	$x(t) = 20/(19e^{t/4} + 1)$
4	$x' = x^2 - x, x(0) = \frac{1}{2}$	$x(t) = (1 + e^{-t})^{-1}$
5	$x' = e^{-x}, x(0) = 1$	$x(t) = \log(e + t)$
6	$x' = \sin x, x(0) = \frac{1}{10}$	$x(t) = 2 \cot^{-1}(e^{-t} \cot 0.05)$
7	$x' = x^{1/3}, x(0) = 1$	$x(t) = \frac{2}{3} \sqrt{\frac{2}{3}} \cdot x^{3/2}$
8	$x' = \tanh 2x, x(0) = 2$	$x(t) = \frac{1}{2} \arcsin h(e^{2t} \sinh 4)$
9	$x' = \sqrt{ 1 - x^2 }, x(\frac{\pi}{6}) = \frac{1}{2}$	$x = \sin t$

We estimated 54 (i.e. 9 problems times 6 tolerances) efficiency measures for each pair. We set NEW65 as a reference pair. Then, we divided each efficiency measure of DLMP6(5) with the corresponding efficiency measure of NEW6(5). The results can be found in Table 3. The figures underlined for problems 5 and 7 are the numbers we found in the original training (in the previous section) with tolerance  $10^{-11}$ . It is obvious that our results are in favor of the second pair. On average, we observed a ratio of 1.75, meaning that DLMP6(5) is about 75% more expensive. This is quite remarkable, since much effort has been spent over the years to achieve 10 – 20% efficiency [23,29]. In reverse, this means that about  $\log_{10} 1.75^6 \approx 1.46$  digits were gained on average, for the same costs.

**Table 3.** Efficiency measures ratios of DLMP6(5) vs NEW6(5).

Problem	Tolerances					
	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	$10^{-10}$	$10^{-11}$
1	1.74	1.53	1.61	2.33	2.27	2.31
2	1.53	1.69	1.69	1.55	1.54	1.42
3	1.66	1.92	2.35	2.04	1.88	1.82
4	1.43	1.51	1.47	1.57	1.41	1.40
5	0.91	1.18	1.06	0.98	3.81	<u>4.86</u>
6	1.68	1.67	1.70	1.71	1.44	1.31
7	0.97	1.24	1.27	1.54	2.22	<u>4.53</u>
8	1.35	1.25	1.28	1.41	1.46	1.40
9	1.85	1.75	1.79	1.82	1.97	1.51

In Table 4, we present the ratios of efficiency measures of ST6(4), with the corresponding efficiency measures of NEW6(5). On average, we observed a ratio of 1.76 meaning that ST6(4) is about 76% more expensive. In reverse, this means that about  $\log_{10} 1.76^6 \approx 1.47$  digits were gained on average, with the same costs. The result is remarkable. ST6(4) was specially designed to address problems of the form (2). NEW6(5) outperformed the other pairs even in clearly non-linear problems. Finally, we highlight that we achieved more or less similar results for longer integrations.

**Table 4.** Efficiency measures ratios of ST6(4) vs NEW6(5).

Problem	Tolerances					
	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	$10^{-10}$	$10^{-11}$
1	1.59	1.54	1.56	2.37	2.28	2.48
2	1.58	1.44	1.51	1.42	1.49	1.33
3	1.49	1.65	2.30	2.08	1.92	1.91
4	1.51	1.62	1.52	1.61	1.43	1.44
5	1.55	2.09	2.12	1.77	5.91	5.42
6	1.57	1.35	1.50	1.60	1.41	1.30
7	0.93	1.17	1.19	1.38	1.68	2.61
8	0.95	0.87	0.89	0.84	0.88	0.80
9	2.35	1.86	1.97	2.05	1.99	1.73

For illustration purposes we have included a couple of efficiency plots for Problems 1 and 6 in Figures 1 and 2, respectively. In these figures, we record the stages used by each pair vs the accuracy achieved. By drawing horizontal lines, we may justify the efficiency ratios for the corresponding problems in the tables above.

The results are very promising. Some future research may use optimization on a wider range of tolerances and model problems. Perhaps a seven-stage pair, that is specially constructed after solving the reduced set of order conditions for problems (2), and then trained properly, would furnish even more interesting results.

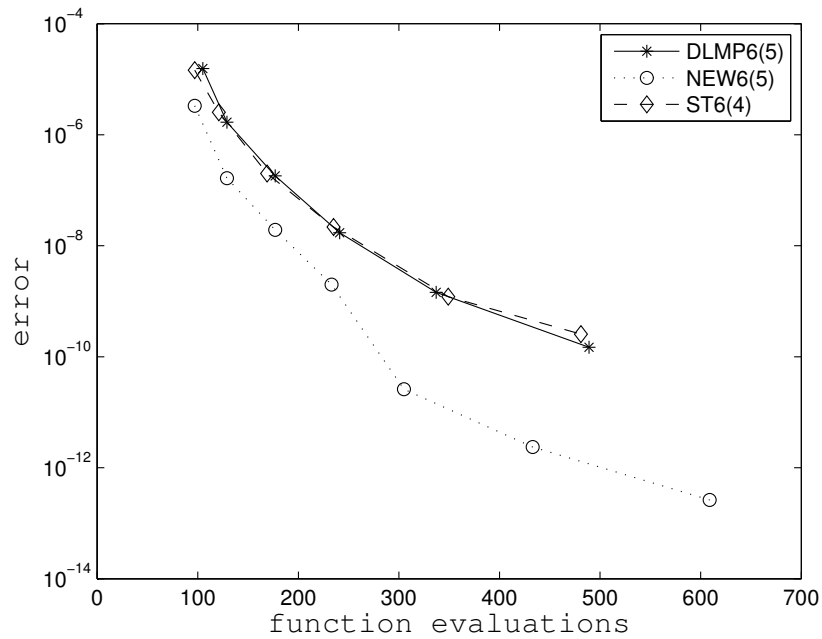


Figure 1. Efficiency plots for Problem-1.

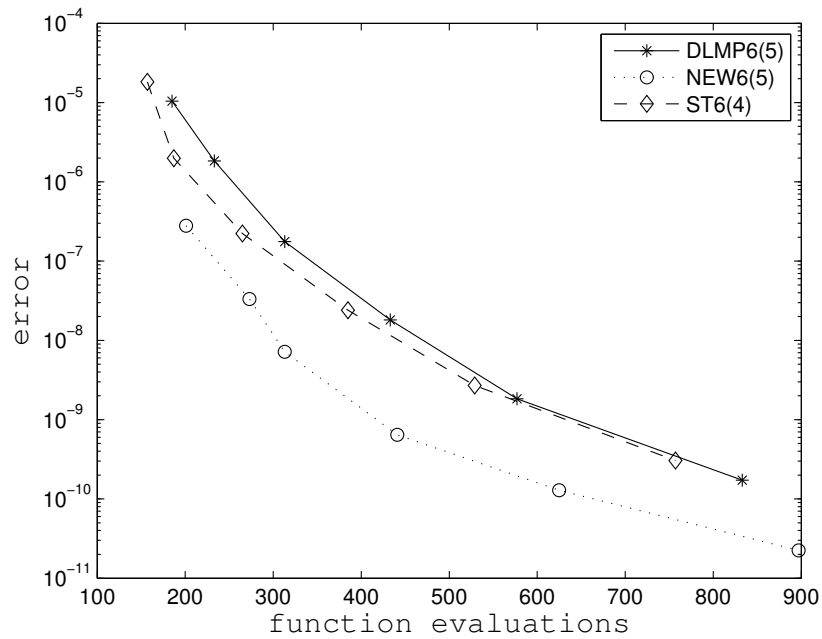


Figure 2. Efficiency plots for Problem-6.

### 5. Conclusions

The training of the coefficients of a Runge–Kutta pair, for addressing a particular kind of problem, is considered. We concentrated on scalar autonomous problems and an extensively studied family of Runge–Kutta pairs of orders 5 and 6. After optimizing the free parameters (coefficients) of the pair with a couple of runs on certain scalar autonomous problems, we proposed a new pair. This pair was found to outperform other representatives from this family in a wide range of relevant problems.



The topic we presented in this paper may expand into many other cases. It can be easily applied to all kinds of Runge–Kutta methods and many other types of Initial Value Problems, such as Hamiltonian problems, Oscillatory problems, and more.

**Author Contributions:** All authors contributed equally. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research was supported by a Mega Grant from the Government of the Russian Federation within the framework of federal project No. 075-15-2021-584.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Butcher, J.C. On Runge–Kutta processes of high order. *J. Austral. Math. Soc.* **1964**, *4*, 179–194. [CrossRef]
- Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons: Chichester, UK, 2003.
- Tsitouras, C.; Papakostas, S.N. Cheap Error Estimation for Runge–Kutta pairs. *SIAM J. Sci. Comput.* **1999**, *20*, 2067–2088. [CrossRef]
- Hairer, E.; Norsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I, Nonstiff Problems*; Springer: Berlin/Heidelberg, Germany, 1993.
- Runge, C. Ueber die numerische Auflösung von Differentialgleichungen. *Math. Ann.* **1895**, *46*, 167–178. [CrossRef]
- Kutta, W. Beitrag zur näherungsweise Integration von Differentialgleichungen. *Z. Math. Phys.* **1901**, *46*, 435–453.
- Fehlberg, E. Klassische Runge–Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle. *Computing* **1969**, *4*, 93–106. [CrossRef]
- Fehlberg, E. Klassische Runge–Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme. *Computing* **1970**, *6*, 61–71. [CrossRef]
- Dormand, J.R.; Prince, P.J. A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26. [CrossRef]
- Prince, P.J.; Dormand, J.R. High order embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **1981**, *7*, 67–75. [CrossRef]
- Tsitouras, C. A parameter study of explicit Runge–Kutta pairs of orders 6(5). *Appl. Math. Lett.* **1998**, *11*, 65–69. [CrossRef]
- Famelis, I.T.; Papakostas, S.N.; Tsitouras, C. Symbolic derivation of Runge–Kutta order conditions. *J. Symbolic Comput.* **2004**, *37*, 311–327. [CrossRef]
- Tsitouras, C. Runge–Kutta pairs of orders 5(4) satisfying only the first column simplifying assumption. *Comput. Math. Appl.* **2011**, *62*, 770–775. [CrossRef]
- Medvedev, M.A.; Simos, T.E.; Tsitouras, C. Fitted modifications of Runge–Kutta pairs of orders 6(5). *Math. Meth. Appl. Sci.* **2018**, *41*, 6184–6194. [CrossRef]
- Simos, T.E.; Kovalnogov, V.N.; Shevchuk, I.V. Perspective of mathematical modeling and research of targeted formation of disperse phase clusters in working media for the next-generation power engineering technologies. In *AIP Conference Proceedings*; AIP Publishing LLC: Melville, NY, USA, 2017; Volume 1863, p. 560099.
- Tsitouras, C. Optimized explicit Runge–Kutta pair of orders 9(8). *Appl. Numer. Math.* **2001**, *38*, 121–134. [CrossRef]
- Shen, Y.C.; Lin, C.L.; Simos, T.E.; Tsitouras, C. Runge–Kutta Pairs of Orders 6 (5) with Coefficients Trained to Perform Best on Classical Orbits. *Mathematics* **2021**, *9*, 1342. [CrossRef]
- Kovalnogov, V.N.; Simos, T.E.; Tsitouras, C. Runge–Kutta pairs suited for SIR-type epidemic models. *Math. Meth. Appl. Sci.* **2021**, *44*, 5210–5216. [CrossRef]
- Dormand, J.R.; Lockyer, M.A.; McGorrigan, N.E.; Prince, P.J. Global error estimation with Runge–Kutta triples. *Comput. Math. Appl.* **1989**, *18*, 835–846. [CrossRef]
- Verner, J.H. Some Runge–Kutta formula pairs. *SIAM J. Numer. Anal.* **1991**, *28*, 496–511. [CrossRef]
- Simos, T.E.; Tsitouras, C. Evolutionary derivation of Runge–Kutta pairs for addressing inhomogeneous linear problems. *Numer. Algor.* **2021**, *62*, 2101–2111. [CrossRef]
- Papakostas, S. N. RhD Dissertation, National Technical University of Athens, Athens. 1996. Available online: <https://www.didaktorika.gr/eadd/handle/10442/6561> (accessed on 23 June 2021)
- Papakostas, S.N.; Tsitouras, C.; Papageorgiou, G. A general family of explicit Runge–Kutta pairs of orders 6(5). *SIAM J. Numer. Anal.* **1996**, *33*, 917–936. [CrossRef]
- Simos, T.E.; Tsitouras, C. 6th order Runge–Kutta pairs for scalar autonomous IVP. *Appl. Comput. Math.*, **2020**, *19*, 412–421.
- Tsitouras, C. Neural Networks With Multidimensional Transfer Functions. *IEEE T. Neural Nets*, **2002**, *13*, 222–228. [CrossRef]
- Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]

- 
27. Liu, C.; Hsu, C.W.; Tsitouras, C.; Simos, T.E. Hybrid Numerov-type methods with coefficients trained to perform better on classical orbits. *Bull. Malays. Math. Sci. Soc.* **2019**, *42*, 2119–2134. [[CrossRef](#)]
  28. DeMat. Available online: <https://www.swmath.org/software/24853> (accessed on 23 June 2021)
  29. Papakostas, S.N.; Tsitouras, C. High phase-lag order Runge-Kutta and Nyström pairs. *SIAM J. Sci. Comput.* **1999**, *21*, 747–763. [[CrossRef](#)]