

# Using Neural Networks for the derivation of Runge-Kutta-Nyström pairs for integration of orbits.

Ch. Tsitouras<sup>a</sup>, I. Th. Famelis<sup>b</sup>

<sup>a</sup>*TEI of Chalkis, Dept. of Applied Sciences, GR34400, Psahna, Greece*

<sup>b</sup>*TEI of Athens, Dept. of Mathematics, GR12210, Egaleo, Greece*

---

## Abstract

In this paper we present Runge–Kutta–Nyström (RKN) pairs of orders 4(3) and 6(4). We choose a test orbit from the Kepler problem to integrate for a specific tolerance. Then we train the free parameters of the above RKN4(3) and RKN6(4) families to perform optimally. For that we form a neural network approach and minimize its objective function using a differential evolution optimization technique. Finally we observe that the produced pairs outperform standard pairs from the literature for Pleiades orbits and Kepler problem over a wide range of eccentricities and tolerances.

*Keywords:* Neural Networks, Runge–Kutta, Kepler problem, Differential Evolution

*PACS:* 02.60.Lj 07.05.Mh

---

## 1. Introduction

Explicit Runge–Kutta–Nyström pairs are widely used for the numerical solution of the initial value problem

$$y'' = f(x, y), \quad y(x_0) = y_0 \in \mathbb{R}^m, \quad y'(x_0) = y_0' \in \mathbb{R}^m, \quad x \in [x_0, x_e]$$

where  $f : \mathbb{R} \times \mathbb{R}^m \mapsto \mathbb{R}^m$ . We usually use the extended Butcher tableau [1] of the method's coefficients :

$$\begin{array}{c|c} c & A \\ \hline & \hat{b}, \hat{b}' \\ & \tilde{b}, \tilde{b}' \end{array}$$

to present the RKN pair. In such a tableau  $b^T, \hat{b}^T, b'^T, \hat{b}'^T, c \in \mathbb{R}^s$  and  $A \in \mathbb{R}^{s \times s}$  is strictly lower triangular.

---

*Email addresses:* [tsitoura@teihal.gr](mailto:tsitoura@teihal.gr) (Ch. Tsitouras), [ifamelis@teiath.gr](mailto:ifamelis@teiath.gr) (I. Th. Famelis )

*URL:* <http://users.ntua.gr/tsitoura/> (Ch. Tsitouras),  
<http://users.teiath.gr/ifamelis/> (I. Th. Famelis )

Such a method implementing the following formulae:

$$y_{n+1} = y_n + h_n y'_n + h_n^2 \sum_{i=1}^s b_i f_{ni}$$

and

$$\hat{y}_{n+1} = y_n + h_n y'_n + h_n^2 \sum_{i=1}^s \hat{b}_i f_{ni}$$

advances the solution from  $x_n$  to  $x_{n+1} = x_n + h_n$  computing at each step approximations  $y_{n+1}, \hat{y}_{n+1}$  to  $y(x_{n+1})$  of orders  $p$  and  $p - 1$  respectively.

It also produces two approximations  $y'_{n+1}, \hat{y}'_{n+1}$  to  $y'(x_{n+1})$  of orders  $p$  and  $p - 1$ , given by

$$y'_{n+1} = y'_n + h_n \sum_{i=1}^s b'_i f_{ni}$$

and

$$\hat{y}'_{n+1} = y'_n + h_n \sum_{i=1}^s \hat{b}'_i f_{ni}.$$

Here

$$f_{ni} = f(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{ij} f_{nj}) \in \mathbb{R}^m$$

for  $i = 1, 2, \dots, s \geq p$ . These embedded form methods (called RKN $p(p - 1)$ ) are implemented with variable step-sizes as we can obtain an estimate

$$u_{n+1} = \max(\|y_{n+1} - \hat{y}_{n+1}\|_\infty, \|y'_{n+1} - \hat{y}'_{n+1}\|_\infty)$$

of the local truncation error of the  $p - 1$  order formula. If this error estimation is greater than a requested tolerance TOL it is common to apply the step-size control algorithm

$$h_{n+1} = 0.9 h_n \cdot \left(\frac{\text{TOL}}{u_{n+1}}\right)^{1/p},$$

to compute the next step-size. If it is not, we use the same formula to recompute the current step. See [12] for more details on the implementation of these type of step size policies.

## 2. Derivation of the RKN pairs

The derivation of better RKN pairs is of continued interest the last 30 - 40 years, see [11] and references therein. The main framework for the construction of RKN pairs is matching Taylor series expansions of  $y(x + h) - y_{n+1}$  and  $y'(x + h) - y'_{n+1}$  after we have expanded the various  $f_{ni}$ 's.

1  
2  
3  
4  
5  
6  
7  
8  
9  
2.1. *RKN<sub>4</sub>(3) pairs*

10 A pair of orders four and three as the one that interests us has to satisfy the  
11 following equations of condition:  
12

$$13 \quad b'e = 1, b'c = \frac{1}{2}, b'c^2 = \frac{1}{3}, b'c^3 = \frac{1}{4} \text{ and } b'Ac = \frac{1}{24} \quad (1)$$

14 when we set

$$15 \quad Ae = \frac{c^2}{2} \quad (2)$$

16 and

$$17 \quad b = b'(e - c) \quad (3)$$

18 with  $e = [1, 1 \dots 1]^T \in \mathbb{R}^s$ .

19 Here we consider the family of Dormand et. al. [2] that needs four stages  
20 per step ( $s = 4$ ). This family uses FSAL (First Stage As Last) device so it  
21 effectively needs only three stages per step. FSAL demands  $c_4 = 1$  and  $a_{4i} = b_i$ ,  
22  $i = 1, 2, 3$ . Thus the parameters available for fulfilling the above mentioned five  
23 equations of condition are:  $c_2, c_3, b'_1, b'_2, b'_3, b'_4$  and  $a_{32}$ . Two of them are free  
24 to choose, namely  $c_2$ , and  $c_3$ . The simplifying assumptions define all the other  
25 coefficients.  
26

27 Similarly, for the coefficients of the lower order formulas after choosing a  $\hat{b}'_4$   
28 we solve

$$29 \quad \hat{b}'e = 1, \hat{b}'c = \frac{1}{2}, \hat{b}'c^2 = \frac{1}{3}$$

30 for  $\hat{b}'_1, \hat{b}'_2, \hat{b}'_3$ .

31 Finally, we set  $\hat{b}_3 = 0.15$  and  $\hat{b}_4 = -1/20$  and solve

$$32 \quad \hat{b} \cdot e = \frac{1}{2} \text{ and } \hat{b} \cdot c = \frac{1}{6}$$

33 for  $\hat{b}_i, i = 1, 2$ . The fixed coefficients for the lower order formulas affect mainly  
34 the step size. For example, smaller values may produce smaller estimations for  
35 the error and in consequence this is equivalent to using more lax tolerances. So  
36 for reasons of comparison we use the ones chosen in [2].  
37

38 When we solve all the equations we conclude to the following expressions  
39 with respect to  $c_2$  and  $c_3$  [10]:  
40

$$41 \quad a_{21} = \frac{c_2^2}{2}, a_{31} = \frac{(c_3(c_2^2(1 - 12c_3) + 6c_2^3c_3 - c_3^2 + 3c_2c_3(1 + c_3)))}{(6c_2(1 - 3c_2 + 2c_2^2))},$$

$$42 \quad a_{32} = \frac{((c_2 - c_3)c_3(-c_3 + c_2(-1 + 3c_3)))}{(6c_2(1 - 3c_2 + 2c_2^2))}, a_{41} = \frac{(1 - 2c_2 - 2c_3 + 6c_2c_3)}{(12c_2c_3)},$$

$$43 \quad a_{42} = \frac{(1 - 2c_3)}{(12c_2^2 - 12c_2c_3)}, a_{43} = \frac{(-1 + 2c_2)}{(12(c_2 - c_3)c_3)}, b'_1 = \frac{(1 - 2c_2 - 2c_3 + 6c_2c_3)}{(12c_2c_3)},$$

$$b'_2 = \frac{(-1 + 2c_3)}{(12c_2(c_2^2 + c_3 - c_2(1 + c_3)))}, b'_3 = \frac{(1 - 2c_2)}{(12(c_2 - c_3)(-1 + c_3)c_3)},$$

$$b'_4 = \frac{(3 - 4c_3 + c_2(-4 + 6c_3))}{(12(-1 + c_2)(-1 + c_3))}, \hat{b}_1 = \frac{-13 + 24c_2 + 9c_3}{60c_2},$$

$$\hat{b}_2 = \frac{13 - 9c_3}{60c_2}, \hat{b}'_1 = \frac{4 - 5c_2 - 5c_3 + 8c_2c_3}{6c_2c_3}, \hat{b}'_2 = \frac{4 - 5c_3}{6c_2^2 - 6c_2c_3}, \hat{b}'_3 = -\frac{4 - 5c_2}{6c_2c_3 - 6c_3^2}.$$

## 2.2. RKN6(4) pairs

For the derivation of such type of pairs we need to solve more equations of condition along with those in (1). If assumptions (2-3) hold, to satisfy algebraic order five the additional conditions are:

$$b'c^4 = \frac{1}{5}, b'Ac^2 = \frac{1}{60}, b'cAc = \frac{1}{30},$$

and

$$b'c^5 = \frac{1}{6}, b'Ac^3 = \frac{1}{120}, b'A^2c = \frac{1}{720}, b'cAc^2 = \frac{1}{72}, b'c^2Ac = \frac{1}{36}$$

to satisfy algebraic order six.

We consider again the family studied in [2, 5] that needs six stages per step ( $s = 6$ ). This family also uses FSAL device so it effectively needs only five stages per step. FSAL device enforces  $c_6 = 1$  and  $a_{6i} = b_i, i = 1, 2, \dots, 5$ . Among the parameters available for fulfilling the above mentioned equations of condition, we choose  $c_2, c_3$ , and  $c_4$  freely. All the other coefficients are defined by the conditions and the simplifying assumptions. More details and the algorithm are given in [5].

The lower order weights have to satisfy

$$\hat{b}'e = 1, \hat{b}'c = \frac{1}{2}, \hat{b}'c^2 = \frac{1}{3}, \hat{b}'c^3 = \frac{1}{4} \text{ and } \hat{b}'Ac = \frac{1}{24}.$$

These are five linear equations with six unknowns. We set  $\hat{b}'_5 = -\frac{2}{5}$  [2] and  $\hat{b}'_6 = b'_6$  and solve four of the above equations for  $\hat{b}'_1, \hat{b}'_2, \hat{b}'_3, \hat{b}'_4$  while we choose the remaining as in [2]. We finally derive the coefficients of the vector  $\hat{b}$  using the assumption  $\hat{b} = \hat{b}'(e - c)$

## 2.3. On the derivation of the pairs

The main question raising now is how to select the free parameters i.e.  $c_2$  and  $c_3$  for the 4(3) pair and  $c_2, c_3$ , and  $c_4$  for the 6(4) pair. For a  $p$ -order RKN method, the minimization of the  $p + 1$  order term in the truncation error expansion seems the best choice for the solution of a general problem. This technique does not consider the nature of each specific problem we want solved. Thus many authors considered many other approaches utilizing various properties of the problems. Such classes of problems are Hamiltonian, orbit, periodic,

Schrödinger and many others. For example periodic problems have been studied extensively, when optimizing the numerical procedure for a specific test problem, and very promising methods have been produced for them [5]. In other cases we deal with some side properties such as symplectiness [8].

Unfortunately in most cases analytical consideration of test problems to be solved produces complicated algebra and enforces us to proceed after making oversimplifications.

Our purpose here is to produce a RKN4(3) and a RKN6(4) pair that have optimal performance for the two body problem. Unfortunately, it is very difficult to derive simple algebraic formulas for the coefficients that may produce better pairs for this problem.

An interesting alternative can be the consideration of Runge-Kutta type neural networks, where the various new families pairs are tested on some model problems to give good predictions for their coefficients.

### 3. The new Runge–Kutta–Nyström pairs

We consider the well known Kepler problem

$$\begin{aligned} y_1'' &= -\frac{y_1}{\sqrt{y_1^2+y_2^2}^3} \\ y_2'' &= -\frac{y_2}{\sqrt{y_1^2+y_2^2}^3} \end{aligned}$$

$x \geq 0$ ,  $y(0) = [1 - \epsilon, 0]^T$ ,  $y'(0) = [0, \sqrt{\frac{1+\epsilon}{1-\epsilon}}]^T$  with  $\epsilon$  the eccentricity of the orbit.

We construct a Neural Network (NN) similar to the one given in [9] for Runge–Kutta methods. In the input we give the eccentricity  $\epsilon$ , the tolerance TOL, the integration interval endpoint  $x_e$  and the two parameters  $c_2$  and  $c_3$  (or  $c_2$ ,  $c_3$  and  $c_4$  for the 6(4) pair). Then the corresponding problem is integrated and we record the error  $ge$  and the total number of the function evaluations  $N$ . Here with  $ge$  we denote the maximum of the norm of the global error on the mesh points of the whole interval of integration. The output is a measure of the numerical method efficiency. We use

$$\text{eff} = N \cdot (ge)^{\frac{1}{4}}, \tag{4}$$

for the 4(3) pair and

$$\text{eff} = N \cdot (ge)^{\frac{1}{6}},$$

for the 6(4) pair. For this pair we choose the step-changing algorithm presented in [12].

We test DEP4(3) pair for  $c_2 = 0.25$ ,  $c_3 = 0.7$ , TOL=  $10^{-4}$  and  $x_e = 20\pi$ . We chose  $\epsilon = 0.25$  as representative eccentricity. This seems to be natural choice as it is the mean value of the existing asteroids in the solar system. Higher order pairs perform better when we choose stringent tolerances. Running with low tolerances, a case that a 4(3) pair is preferable, we experience large errors after some integration point. Increasing the eccentricity we get even larger errors.

Thus such pairs are better suited for crude tolerances, small eccentricities and small integration intervals.

We record

$$\text{eff}_{\text{DEP43}}(\text{TOL} = 10^{-4}, \epsilon = 0.25) = 289.87$$

Then we train the coefficients for the NN described above. We set  $\hat{b}_3 = \frac{3}{2}$ ,  $\hat{b}_4 = -\frac{1}{20}$  and  $\hat{b}'_4 = 0.19$  similar to DEP43. These values do not affect seriously the efficiency of the pairs but it is important to tune the error estimator in a way so the various 4(3) pairs have similar costs for similar tolerances. In such a case efficiency is more influenced by the global error achieved.

When doing so, we get

$$\text{eff}_{\text{NEW43}}(\text{TOL} = 10^{-4}, \epsilon = 0.25) = 164.50$$

for

$$c_2 = \frac{50}{99} \text{ and } c_3 = \frac{19}{40}.$$

We list the coefficients of the new pair in Table-1.

Table 1: Coefficients of NEW4(3)

0				
$\frac{50}{99}$	$\frac{1250}{9801}$			
$\frac{19}{40}$	$-\frac{5935429}{44800000}$	$\frac{10989429}{44800000}$		
1	$\frac{949}{5700}$	$\frac{3267}{11900}$	$\frac{400}{6783}$	
$b$	$\frac{949}{5700}$	$\frac{3267}{11900}$	$\frac{400}{6783}$	
$b'$	$\frac{949}{5700}$	$\frac{323433}{583100}$	$\frac{16000}{142443}$	$\frac{514}{3087}$
$\hat{b}$	$\frac{4483}{40000}$	$\frac{11517}{40000}$	$\frac{3}{20}$	$-\frac{1}{20}$
$\hat{b}'$	$\frac{13399}{95000}$	$-\frac{153549}{595000}$	$\frac{2096}{2261}$	$\frac{19}{100}$

The main issue of our effort is to get a method using the least information. A sole integration is expected to give a pair that performs best over all eccentricities and tolerances. TOL was not included in the definition of  $eff$  since it is not a qualification output for the pairs. Another approach could be to define  $eff$  as an average of a variety of combinations of  $e$  and TOL.

The neural networks are actually nonlinear optimizers. There exist other nonlinear optimization techniques like conjugate gradient, back-propagation, or other Newton-type methods based on some kind estimation of derivatives. Such "standard" techniques are not worth here to be used in our case where we optimize a whole integration over the coefficients of a method. The derivatives (i.e. Jacobians) of the object functions are really difficult to estimate. A two-dimensional curve of  $eff$  with respect to  $c_2$  and  $c_3$  is too expensive though is not informative since there are very small regions where peaks in  $eff$  can not be observed.

Thus we use a differential evolution (DE) technique for this purpose. DE is a population based method which seems to perform better here, where the

output comes after a complete run of the Initial Value Problem. We implement DeMat software for Matlab as DE method, see [6].

Finally we test the new pair for a wide range of tolerances and eccentricities. Actually for

$$TOL = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$$

and

$$\epsilon = 0.00, 0.05, 0.10, \dots, 0.70$$

we record the values  $\text{eff}_{DEP43}(TOL, \epsilon)$  for DEP4(3) pair and  $\text{eff}_{NEW43}(TOL, \epsilon)$  for the new pair. We avoided larger eccentricities because they rarely occur in practice. The corresponding quotients

$$\text{eff}_{NEW43}(TOL, \epsilon) / \text{eff}_{DEP43}(TOL, \epsilon)$$

are given in Table-2. The average ratio is 1.32 which means that the new pair is about 32% more efficient in the family of Kepler problems. In this table the underlined number  $1.76 \approx \frac{289.87}{164.50}$ .

Table 2:  $\text{eff}_{DEP43}(TOL, \epsilon) / \text{eff}_{NEW43}(TOL, \epsilon)$

$TOL$	$\epsilon$														
	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70
$10^{-2}$	0.98	1.15	1.17	1.06	1.09	1.19	1.34	1.76	2.44	1.87	1.65	1.60	1.80	2.96	1.57
$10^{-3}$	1.01	1.02	1.18	1.25	1.27	1.43	1.58	1.87	1.76	1.50	1.42	1.43	1.58	2.59	1.64
$10^{-4}$	1.02	1.02	1.08	1.18	1.33	<u>1.76</u>	1.61	1.31	1.20	1.16	1.15	1.15	1.17	1.29	1.61
$10^{-5}$	1.03	1.02	1.02	1.06	1.12	1.27	1.50	1.39	1.16	1.06	1.01	0.99	1.01	1.08	1.37
$10^{-6}$	1.03	1.01	1.00	1.01	1.01	1.04	1.13	1.22	1.30	1.28	1.12	1.06	1.06	1.13	1.47

We proceed testing DEP6(4) pair for  $c_2 = 0.1$ ,  $c_3 = 0.3$ ,  $c_4 = 0.7$ ,  $TOL = 10^{-6}$  and  $x_e = 20\pi$ . We tried here  $\epsilon = 0.35$  since the pair is of higher order and is expected to perform better for large eccentricities and smaller tolerances. We find

$$\text{eff}_{DEP64}(TOL = 10^{-7}, \epsilon = 0.35) = 335.86$$

Then we train the free parameters of this family for this NN in the lines described for the previous case keeping the same values of  $\hat{b}'_5, \hat{b}'_6$  as in DEP6(4) for tuning reasons and get

$$\text{eff}_{NEW64}(TOL = 10^{-7}, \epsilon = 0.35) = 138.33$$

for

$$c_2 = \frac{2}{11}, c_3 = \frac{4}{9} \text{ and } c_4 = \frac{5}{8}.$$

The coefficients of the new pair are listed in Table-3.

Finally, we test the new pair for a wide range of tolerances and eccentricities. Actually for

$$TOL = 10^{-4}, 10^{-5}, \dots, 10^{-9}$$

Table 3: Coefficients of NEW6(4)

0																
$\frac{2}{11}$	$\frac{2}{121}$															
$\frac{4}{9}$	$\frac{404}{67797}$	$\frac{6292}{67797}$														
$\frac{5}{8}$	$\frac{1056145}{77201408}$	$\frac{5544825}{38600704}$	$\frac{2932605}{77201408}$													
$\frac{163}{244}$	$\frac{328961}{14300210}$	$\frac{6078447}{41566085}$	$\frac{3270629}{65455319}$	$\frac{444237}{113127028}$												
1	$\frac{8707}{130400}$	$\frac{3059969}{17643600}$	$\frac{745767}{2655328}$	$-\frac{150784}{266175}$	$\frac{221533456}{406167475}$											
$b$	$\frac{8707}{130400}$	$\frac{3059969}{17643600}$	$\frac{745767}{2655328}$	$-\frac{150784}{266175}$	$\frac{221533456}{406167475}$											
$b'$	$\frac{8707}{130400}$	$\frac{33659659}{158792400}$	$\frac{6711903}{13276640}$	$-\frac{1206272}{798525}$	$\frac{54054163264}{32899565475}$	$\frac{1}{12}$										
$\hat{b}$	$-\frac{5337191}{43580352}$	$\frac{4180502689}{6137566240}$	$-\frac{1158531525}{2455026496}$	$\frac{62835974}{115079367}$	$-\frac{81}{610}$											
$\hat{b}'$	$-\frac{5337191}{43580352}$	$\frac{45985529579}{55238096160}$	$-\frac{2085356745}{2455026496}$	$\frac{502687792}{345238101}$	$-\frac{2}{5}$	$\frac{1}{12}$										

and  $\epsilon = 0.00, 0.05, 0.10, \dots, 0.70$  we record the values  $\text{eff}_{DEP64}(\text{TOL}, \epsilon)$  for DEP6(4) pair and  $\text{eff}_{NEW64}(\text{TOL}, \epsilon)$  for the new pair. Here we integrated for a longer period,  $x_{end} = 60\pi$ . For even longer periods it is better to use lower tolerances and in consequence higher order pairs. The corresponding quotients

$$\text{eff}_{DEP64}(\text{TOL}, \epsilon) / \text{eff}_{NEW64}(\text{TOL}, \epsilon)$$

are shown in Table-4. The average of these ratios is 1.45 which means that the new pair is about 45% more efficient in the family of Kepler problems.

Table 4:  $\text{eff}_{NEW64}(\text{TOL}, \epsilon) / \text{eff}_{DEP64}(\text{TOL}, \epsilon)$

$TOL$	$\epsilon$														
	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70
$10^{-4}$	1.13	1.09	1.15	1.16	1.23	1.33	1.53	1.86	1.34	1.34	1.40	1.19	1.19	1.22	1.23
$10^{-5}$	1.20	1.12	1.13	1.19	1.24	1.28	1.39	1.87	1.60	1.53	1.57	1.51	1.46	1.46	1.39
$10^{-6}$	1.32	1.20	1.10	1.16	1.31	1.44	1.50	1.60	1.69	1.90	2.46	2.18	1.87	1.62	1.51
$10^{-7}$	1.45	1.38	1.16	1.13	1.19	1.37	1.58	2.23	1.74	1.57	1.66	1.71	1.73	1.76	1.70
$10^{-8}$	1.31	1.34	1.29	1.17	1.21	1.31	1.44	1.77	1.84	1.56	1.51	1.49	1.47	1.46	1.53
$10^{-9}$	1.19	1.21	1.32	1.28	1.26	1.34	1.45	1.66	2.13	1.59	1.49	1.45	1.41	1.37	1.33

We note that when integrating the two body problem, the ratio of the global error of methods can be sensitive to small changes in eccentricity, [7]. Complete smoothness of the ratios is not guaranteed.

We extend our tests for the well known orbit Pleiades taken from [4]. This is a celestial mechanics problem of seven stars in the plane of coordinates  $x_i, y_i$  and masses  $m_i, (i = 1, \dots, 7)$  which has the form:

$$z'' = f(z), z(0) = z_0, z'(0) = z'_0$$

with

$$z \in \mathbb{R}^{14}, 0 \leq t \leq 3.$$

1  
2  
3  
4  
5  
6  
7  
8  
9 Defining  $z := (x^T, y^T)^T$ ,  $x, y \in \mathbb{R}^7$ , the function  $f : \mathbb{R}^{14} \rightarrow \mathbb{R}^{14}$  is given by  
10  $f(z) = f(x, y) = (f^{(1)}(x, y)^T, f^{(2)}(x, y)^T)^T$  where  $f^{(1,2)} : \mathbb{R}^{14} \rightarrow \mathbb{R}^7$  read

$$11 \quad f_i^{(1)} = \sum_{i \neq j} m_j (x_j - x_i) / r_{ij}^{\frac{3}{2}}, \quad f_i^{(2)} = \sum_{i \neq j} m_j (y_j - y_i) / r_{ij}^{\frac{3}{2}},$$

12  
13  
14 where,  $m_i = i$  and

$$15 \quad r_{ij} = (x_j - x_i)^2 + (y_j - y_i)^2.$$

16  
17 The initial values are  $z_0 = (3, 3, -1, -3, 2, -2, 2, 3, -3, 2, 0, 0, -4, 4)^T$  and  
18  $z'_0 = (0, 0, 0, 0, 0, 1.75, -1.5, 0, 0, 0, -1.25, 1, 0, 0)^T$  and the reference solution at  
19 the end of the integration interval can be found in [3].

20 For this problem, for the 4(3) pairs the average ratio of

$$21 \quad \text{eff}_{NEW43}(\text{TOL}) / \text{eff}_{DEP43}(\text{TOL})$$

22  
23 for

$$24 \quad \text{TOL} = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$$

25  
26 is 1.14 which means that the new pair is about 14% more efficient in this specific  
27 problem.

28 For the 6(4) pairs the average ratio of

$$29 \quad \text{eff}_{NEW64}(\text{TOL}) / \text{eff}_{DEP64}(\text{TOL})$$

30  
31 for

$$32 \quad \text{TOL} = 10^{-4}, 10^{-5}, \dots, 10^{-9}$$

33  
34 is 1.22 which means that the new pair is about 22% more efficient in this specific  
35 problem.

#### 36 37 38 **4. Conclusion**

39 We present a fairly new general technique in this paper. We construct nu-  
40 merical methods tuned to integrate specific classes of nonlinear problems. This  
41 is achieved by training their coefficients to perform optimally at a chosen repre-  
42 sentative of this class. We record some first promising results for Runge–Kutta–  
43 Nyström pairs which solve orbits and especially the Kepler problem.  
44  
45

#### 46 47 **References**

- 48  
49 [1] J. C. Butcher, The Numerical Analysis of Ordinary Differential Equations,  
50 John Wiley & Sons Inc., New York (1987).  
51  
52 [2] J. R. Dormand, M. E. El-Mikkawy and P. J. Prince, Families of Runge-  
53 Kutta-Nyström formulae, IMA J. Numer. Anal., **7**, 235-250 (1987).  
54  
55 [3] Francesca Mazzia and Cecilia Magherini, Test Set for Initial Value Problem  
56 Solvers Release 2.4, **Report 4/2008**, Department of Mathematics University  
57 of Bari.  
58

- 1  
2  
3  
4  
5  
6  
7  
8  
9 [4] E. Hairer, S. P. Nørsett and G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer-Verlag, 2nd ed. Berlin (1993).  
10  
11  
12 [5] S. N. Papakostas and Ch. Tsitouras, High Phase-Lag-Order Runge–Kutta  
13 and Nyström Pairs, SIAM J. Sci. comput., **21**, 747-763 (1999).  
14  
15 [6] K. V. Price, R. M. Storn and J. A. Lampinen, Differential Evolution: A  
16 practical approach to global optimization, Springer, Berlin (2005).  
17  
18 [7] P.W Sharp, J.C. Castillo-Rogezb, K.R. Grazier, The performance of phase-  
19 lag enhanced ERKN pairs on N-body problems, J. Comput. Appl. Maths, to  
20 appear.  
21  
22 [8] Ch. Tsitouras, A Tenth Order Symplectic Runge–Kutta–Nyström Method,  
23 Cele. Mech. Dynam. Astron. **74**, 223-230 (1999).  
24  
25 [9] Ch. Tsitouras, Neural Networks with multidimensional transfer functions,  
26 IEEE T. Neural Networks, **20**, 222-228 (2002).  
27  
28 [10] Ch. Tsitouras, Using Neural Networks for the Derivation of Runge-Kutta-  
29 Nystrom pairs, Amer. Inst. Phys., CP-**1048**, 1034-1036 (2008).  
30  
31 [11] Ch. Tsitouras and I. Th. Famelis, Symbolic derivation of Runge-Kutta-  
32 Nyström order conditions. J. Math. Chem., **46**, 896-912 (2009).  
33  
34 [12] Ch. Tsitouras and S. N. Papakostas, Cheap Error Estimation for Runge-  
35 Kutta pairs, SIAM J. Sci. Comput., **20**, 2067-2088 (1999).  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65