mathematics

MDPI

Article

Eighth Order Two-Step Methods Trained to Perform Better on Keplerian-Type Orbits

Vladislav N. Kovalnogov [1], Ruslan V. Fedorov [1], Andrey V. Chukalin [1], Theodore E. Simos [1,2,3,4,5,6,*] and Charalampos Tsitouras [7,8]

1 Laboratory of Inter-Disciplinary Problems of Energy Production, Ulyanovsk State Technical University, 432027 Ulyanovsk, Russia; kvn@ulstu.ru (V.N.K.); r.fedorov@ulstu.ru (R.V.F.); chukalin.andrej@mail.ru (A.V.C.)
2 College of Applied Mathematics, Chengdu University of Information Technology, Chengdu 610225, China
3 Department of Medical Research, China Medical University Hospital, China Medical University, Taichung City 40402, Taiwan
4 Data Recovery Key Laboratory of Sichuan Province, Neijiang Normal University, Neijiang 641100, China
5 Section of Mathematics, Department of Civil Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
6 Department of Mathematics, University of Western Macedonia, 50100 Kozani, Greece
7 General Department, National & Kapodistrian University of Athens, 10679 Athens, Greece; tsitourasc@uoa.gr
8 Department of Business Administration, Hellenic Open University, 26335 Patras, Greece
* Correspondence: simost@susu.ru

Abstract: The family of Numerov-type methods that effectively uses seven stages per step is considered. All the coefficients of the methods belonging to this family can be expressed analytically with respect to four free parameters. These coefficients are trained through a differential evolution technique in order to perform best in a wide range of Keplerian-type orbits. Then it is observed with extended numerical tests that a certain method behaves extremely well in a variety of orbits (e.g., Kepler, perturbed Kepler, Arenstorf, Pleiades) for various steplengths used by the methods and for various intervals of integration.

Keywords: initial value problem (IVP; second-order IVP); Numerov-type methods; two-body problem; perturbed Kepler; differential evolution

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

Citation: Kovalnogov, V.N.; Fedorov, R.V.; Chukalin, A.V.; Simos, T.E.; Tsitouras, C. Eighth Order Two-Step Methods Trained to Perform Better on Keplerian-Type Orbits. *Mathematics* **2021**, *9*, 3071. https://doi.org/10.3390/math9233071

Academic Editors: Junseok Kim, Hongyu Liu and Danny Barash

Received: 12 October 2021
Accepted: 25 November 2021
Published: 29 November 2021

1. Preliminary Discussion

We are interested in the particular version of the Initial Value Problem (IVP):

$$y'' = f(t, y), \; y(t_0) = y_0, \; y'(t_0) = y_0', \tag{1}$$

where $f : \mathbb{R} \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$ and $y_0, \; y_0' \in \mathbb{R}^m$. This problem is used to simulate a variety of physical models. Observe that y' is not a part of (1).

The Numerov scheme is one of the most known methods for addressing (1). It aids in propagation of the numerical approximation of the solution from t_λ to $t_{\lambda+1} = t_\lambda + h$, in accordance to the following formula:

$$y_{\lambda+1} = 2y_\lambda - y_{\lambda-1} + \frac{h^2}{12}(f_{\lambda+1} + 10f_\lambda + f_{\lambda-1}),$$

with $y_\lambda \approx y(t_\lambda)$ and $f_\lambda \approx y_\lambda'' = f(x_\lambda, y_\lambda)$. Remark also that $y_\lambda, f_\lambda \in \mathbb{R}^m$.

Implicit Numerov-type methods that use points off-step were proposed firstly almost four decades ago, beginning with the work of Hairer [1] and followed by Cash [2] and

Chawla [3]. The main focus at the time was on the P-stability feature. This later property was demanded when dealing stiff problems with periodic solutions.

A little later, Chawla [4], made a modification to the Numerov producing the explicit scheme:

$$\begin{aligned} w_1 &= y_{\lambda-1}, \\ w_2 &= y_\lambda, \\ w_3 &= 2y_\lambda - y_{\lambda-1} + h^2 f(t_\lambda, w_2), \end{aligned} \tag{2}$$

$$y_{\lambda+1} - 2y_\lambda + y_{\lambda-1} = \tfrac{1}{12}h^2 \cdot (f(x_{\lambda+1}, w_3) + 10f(x_\lambda, w_2) + f(x_{\lambda-1}, w_1)),$$

where h the steplength used which remains constant through the integration:

$$h = t_{\lambda+1} - t_\lambda = t_\lambda - t_{\lambda-1} = \cdots = t_1 - t_0.$$

The vectors $y_{\lambda-1}$, y_λ and $y_{\lambda+1}$ approximate $y(t_\lambda - h)$, $y(x_\lambda)$ and $y(x_\lambda + h)$ respectively while $w_1 \in \mathbb{R}^m$, $w_2 \in \mathbb{R}^m$ and $w_3 \in \mathbb{R}^m$ are the stages of the method.

Following the common technique we use information known at grid by setting:

$$w_1 = y_{\lambda-1},\ w_2 = y_\lambda.$$

After $f(t_{\lambda-1}, w_1)$ has already evaluated in the previous step, only $f(t_{\lambda+1}, w_3)$ and $f(t_\lambda, w_2)$ are the new stages (function evaluations) wasted every step.

Subsequently, Tsitouras proposed an approach in the form of Runge–Kutta–Nyström (RKN)-type methods [5]. The technique he proposed lowered significantly the cost. This results in creating a sixth-order method by using only four stages (see [5]). On the contrary, older implementations necessitated the evaluation of six stages per step [6].

In the years thereafter, our research group has delved deeper into the subject. Tsitouras [7] produced methods that attained eighth algebraic order using only nine stages each step. Simultaneously, a group of Spanish researchers working on this subject, presented very interesting results as well [8–10].

2. Theory Numerov-Type Methods Using Off-Step Nodes

For addressing numerically problem (1) we are interested in using higher-order schemes. Here t which is the independent variable is incorporated as an extra component of y. Then, we concentrate to $y'' = f(y)$ which is an autonomous system without losing the generality of the approach. Thus, an s-stage hybrid Numerov scheme gets the form:

$$\begin{aligned} y_{k+1} &= 2y_k - y_{k-1} + h^2 \cdot (b \otimes I_s) \cdot f(u) \\ w &= (e + c) \otimes y_k - c \otimes y_{k-1} + h^2 \cdot (A \otimes I_s) \cdot f(w) \end{aligned} \tag{3}$$

where $I_s \in \mathbb{R}^{s \times s}$ is the identity matrix. In expression (3) above $A \in \mathbb{R}^{s \times s}, b^T \in \mathbb{R}^s, c \in \mathbb{R}^s$ are the matrices and vectors containing the coefficients of the method while

$$e = [1\ 1\ \cdots 1]^T \in \mathbb{R}^s.$$

Butcher tableau [11,12] is in common use in formulating the coefficients,

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

The method given by Formula (2) can be represented using matrices. The function evaluations are computed in sequence and the methods are explicit. The matrix of coefficients A is strictly lower triangular. We assume $s = 8$ and now the coefficients of the method can by tabulated as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} b_1 & b_2 & 0 & b_4 & b_4 & b_5 & b_5 & b_1, \end{bmatrix},$$

and

$$c = \begin{bmatrix} -1 & 0 & c_3 & c_4 & -c_4 & -c_5 & c_5 & 1 \end{bmatrix}^T.$$

Then only seven function evaluations are evaluated per step since $f(w_1)$ is known from the last step. Our aim now is to achieve eighth order and for accomplishing this, all order conditions of the same and of lower order have to be eliminated, see [13].

The scheme we consider here possesses 34 parameters. Namely, 27 entries through matrix A (i.e., $a_{31}, a_{32}, \cdots, a_{87}$), another 4 coefficients for vector b, and finally another 3 entries coming from c. Using these parameters we have to satisfy 62 order conditions for attaining 8–th order (follow Table 5 from [13]).

The offered parameters are much less than the equations. This is a usual occurrence while deriving Runge–Kutta-type schemes. Using simplifying assumptions is a frequent way to get around issue. Here we only use,

$$A \cdot e = \frac{1}{2}(c + c^2), \; A \cdot c = \frac{1}{6}(-c + c^3).$$

We waste only the six parameters $a_{31}, \; a_{32}, \; a_{41}, \; a_{42}, \; a_{51}$, and a_{52} to address these assumptions. Then all terms that contain the subexpressions $A \cdot e$ and $A \cdot c$ can be dropped from the corresponding order conditions listed in [13]. As a result, the remaining 11 coefficients must now fulfill only nine order criteria.

We may solve this problem analytically with respect to a quadruplet of parameters that can be chosen arbitrarily. Let us choose c_3, c_4, c_5, and a_{64}. The rest coefficients are then presented below.

$$b_1 = \frac{14c_4^2\left(5c_5^2 - 2\right) - 28c_5^2 + 15}{840\left(c_4^2 - 1\right)\left(c_5^2 - 1\right)}, \; b_2 = \frac{14c_4^2\left(25c_5^2 - 3\right) - 42c_5^2 + 13}{420c_4^2c_5^2},$$

$$b_4 = \frac{42c_5^2 - 13}{840c_4^2\left(c_4^2 - 1\right)\left(c_4^2 - c_5^2\right)}, \; b_5 = \frac{42c_5^2 - 13}{840c_4^2\left(c_4^2 - 1\right)\left(c_4^2 - c_5^2\right)},$$

$$a_{31} = \frac{1}{6}\left(c_3 - c_3^3\right), \; a_{32} = \frac{1}{6}c_3\left(c_3^2 + 3c_3 + 2\right), \; a_{43} = \frac{c_4\left(c_4^3 + 2c_4^2 - 1\right)}{12c_3(c_3 + 1)},$$

$$a_{53} = -\frac{c_4\left(c_3\left(c_4^3 + 2c_4^2 - 1\right) + c_4^4 - 4c_4^3 + c_4\right)}{12c_3(c_3 + 1)(c_3 - c_4)}, \; a_{54} = \frac{(c_3 - 1)c_4^3}{6(c_4 + 1)(c_3 - c_4)},$$

$$a_{63} = \frac{c_5\left(\begin{array}{c} -c_3\left(c_4^2 + c_4 - 1\right)\left(42c_5^4 - 55c_5^2 + 13\right) + 42c_4^4\left(c_5^3 - 2c_5^2 + 1\right) + 51c_4^3\left(c_5^2 - 1\right) \\ +c_4^2\left(42c_5^4 - 55c_5^3 + 4c_5^2 + 9\right) + c_4\left(-9c_5^4 - 4c_5^2 + 13\right) + c_5^2\left(9c_5^2 + 13c_5 - 22\right) \end{array}\right) - 24a_{64}c_4^2\left(42c_4^4 - 55c_4^2 + 13\right)}{12c_3(c_3 + 1)\left(42c_4^3 - 42c_4^2 - 13c_4 + 13\right)(c_3 + c_4)},$$

$$a_{65} = \frac{\begin{array}{c}(c_5-1)c_5(c_4+c_5)\left(\begin{array}{c} c_4^2\big(42c_3(c_5^2-c_5-1)-3(14c_5^2+3c_5+3)\big) \\ +c_4(c_5+1)(c_3(42c_5+29)+9c_5+22) \\ -c_5(c_3(42c_5+29)+9c_5+22) \end{array}\right) \\ -12a_{64}c_4\big(42c_4^4-55c_4^2+13\big)(c_3-c_4)\end{array}}{12(c_4-1)^2c_4\big(42c_4^2-13\big)(c_3+c_4)},$$

$$a_{73} = \frac{\left(\begin{array}{c}(c_5+1)\left(\begin{array}{c}(c_5-1)c_5\left(\begin{array}{c} c_3^2\big(c_4^2+c_4-1\big)\big(42c_5^3-42c_5^2-13c_5+13\big) \\ +c_3\left(\begin{array}{c} 42c_4^4\big(c_5^2-c_5-1\big) \\ +2c_4^3\big(63c_5^3+21c_5^2+6c_5+19\big) \\ +c_4^2\big(29c_5^2+4c_5-22\big) \\ -51c_4c_5^2(c_5+1)+c_5^2(9c_5+22) \end{array}\right) \\ +c_4\left(\begin{array}{c} -42c_4^4\big(c_5^2-c_5-1\big)-51c_4^3(c_5+1) \\ +c_4^2\big(-210c_5^3+13c_5^2+61c_5+9\big) \\ +c_4\big(93c_5^3+9c_5^2-13c_5+13\big)-c_5^2(9c_5+22) \end{array}\right)\end{array}\right)\end{array}\right) \\ -24a_{64}c_4^2\big(42c_4^4-55c_4^2+13\big)(c_3-c_4)\end{array}\right)}{12c_3(c_3+1)\big(42c_4^3-42c_4^2-13c_4+13\big)(c_5-1)\big(c_3^2-c_4^2\big)},$$

$$a_{74} = -\frac{\left(\begin{array}{c}(c_5+1)\left(\begin{array}{c}(c_5-1)c_5^2\left(\begin{array}{c} c_3\left(\begin{array}{c} c_4^3\big(84c_5^2-42c_5-17\big) \\ +c_4^2\big(42c_5^2+42c_5-20\big) \\ +c_4\big(-51c_5^2+11c_5+11\big) \\ +c_5(9c_5-11) \end{array}\right) \\ +c_4\left(\begin{array}{c} c_4^3(42c_5-9)-2c_4^2\big(63c_5^2+21c_5-23\big) \\ +c_4\big(51c_5^2-11c_5-11\big)+(11-9c_5)c_5 \end{array}\right)\end{array}\right)\end{array}\right) \\ -12a_{64}c_4^2\big(42c_4^4-55c_4^2+13\big)(c_3-c_4)\end{array}\right)}{12c_4^2\big(42c_4^4-55c_4^2+13\big)(c_5-1)(c_3-c_4)},$$

$$a_{75} = \frac{\left(\begin{array}{c}(c_5+1)\left(\begin{array}{c}(c_5-1)c_5(c_4+c_5)\left(\begin{array}{c} c_4^3\big(42c_3\big(c_5^2-c_5-1\big)-84c_5^2-9\big) \\ +c_4^2\big(c_3(80c_5+29)+42c_5^2+33c_5+22\big) \\ -3c_4c_5(3c_3(c_5+3)+11)+c_3c_5(9c_5-11) \end{array}\right)\end{array}\right) \\ -12a_{64}c_4^2\big(42c_4^4-55c_4^2+13\big)(c_3-c_4)\end{array}\right)}{12(c_4-1)^2c_4^2\big(42c_4^2-13\big)(c_5-1)(c_3+c_4)},$$

$$a_{76} = \frac{\big(42c_4^2-11\big)c_5(c_5+1)}{6\big(42c_4^2-13\big)(c_5-1)},$$

$$a_{83} = -\frac{\begin{array}{c} 24a_{64}c_4^2\big(42c_4^3-42c_4^2-13c_4+13\big)(c_4+1)^2(c_3-c_4) \\ +(c_5-1)c_5(c_4+c_5)\left(\begin{array}{c} c_3\left(\begin{array}{c} -42c_4^4\big(c_5^2-c_5-1\big) \\ -c_4^3\big(42c_5^3+168c_5^2+25c_5-17\big) \\ +c_4^2\big(42c_5^3+9c_5^2-33c_5-29\big) \\ +c_4\big(42c_5^2+38c_5+22\big)-c_5(9c_5+22) \end{array}\right) \\ +c_4\left(\begin{array}{c} 42c_4^4\big(c_5^2-c_5-1\big) \\ +c_4^3\big(-42c_5^3+84c_5^2+51c_5+9\big) \\ +c_4^2\big(42c_5^3+75c_5^2+7c_5+3\big) \\ -2c_4\big(21c_5^2+19c_5+11\big)+c_5(9c_5+22) \end{array}\right)\end{array}\right)\end{array}}{6c_3(c_3+1)(c_5-1)c_5\big(c_3^2-c_4^2\big)\big(14c_4^4\big(5c_5^2-2\big)+c_4^2\big(15-70c_5^4\big)+c_5^2\big(28c_5^2-15\big)\big)},$$

$$a_{84} = \frac{\begin{array}{c} 24a_{64}c_4^2(c_4+1)^2\big(42c_4^3-42c_4^2-13c_4+13\big)(c_3-c_4) \\ -c_5\big(c_5^2-1\big)\left(\begin{array}{c} c_3\left(\begin{array}{c} c_4^4\big(84c_5^2-42c_5-17\big) \\ +c_4^3\big(112c_5^3+14c_5^2-32c_5-5\big) \\ +c_4^2\big(28c_5^3-37c_5^2+47c_5-3\big) \\ +c_4\big(-28c_5^3-14c_5^2+6c_5+5\big) \\ -28c_5^3+37c_5^2-5c_5-6 \end{array}\right) \\ +c_4\left(\begin{array}{c} c_4^4(42c_5-9)+c_4^3\big(-28c_5^3-98c_5^2+6c_5+31\big) \\ +c_4^2\big(-112c_5^3+37c_5^2-21c_5+3\big) \\ +c_4\big(28c_5^3+14c_5^2-6c_5-5\big)+28c_5^3-37c_5^2+5c_5+6 \end{array}\right)\end{array}\right)\end{array}}{12c_4^2(c_4+1)(c_5-1)c_5(c_3-c_4)\big(14c_4^4\big(5c_5^2-2\big)+c_4^2\big(15-70c_5^4\big)+c_5^2\big(28c_5^2-15\big)\big)},$$

$$a_{85}=\frac{\left(\begin{array}{c}(c_4+1)\left(\begin{array}{c}(c_5-1)c_5\left(\begin{array}{c}c_3\left(\begin{array}{c}84c_4^4(c_5^2-c_5-1)\\+c_4^3(84c_5^3-42c_5^2+25c_5+67)\\+c_4^2(-28c_5^4+42c_5^3+92c_5^2+44c_5-4)\\+c_4(56c_5^4-51c_5^3-82c_5^2+1)\\-28c_5^4+9c_5^3+32c_5^2-11c_5-6\end{array}\right)\\+c_4\left(\begin{array}{c}-3c_4^3(42c_5^2+17c_5+3)\\+c_4^2(-28c_5^4-126c_5^3+34c_5^2+88c_5+40)\\+c_4(56c_5^4+51c_5^3+20c_5^2+1)\\-28c_5^4-9c_5^3-12c_5^2-11c_5-6\end{array}\right)\end{array}\right)\\-24a_{64}c_4^2(42c_4^4-55c_4^2+13)(c_3-c_4)\end{array}\right)\end{array}\right)}{12(c_4-1)c_4^2(c_5-1)c_5(c_3+c_4)\left(\begin{array}{c}14c_4^4(5c_5^2-2)\\+c_4^2(15-70c_5^4)+c_5^2(28c_5^2-15)\end{array}\right)},$$

$$a_{86}=-\frac{(c_4^2-1)(c_5+1)(14c_4^2(c_5^2+c_5+1)-3c_5^2-5c_5-3)}{6(c_5-1)c_5^2(c_4^4(28-70c_5^2)+5c_4^2(14c_5^4-3)-28c_5^4+15c_5^2)},$$

$$a_{87}=-\frac{(14c_4^4-17c_4^2+3)(c_5^2-1)}{6c_5^2(c_4^4(28-70c_5^2)+5c_4^2(14c_5^4-3)-28c_5^4+15c_5^2)},$$

$$a_{41}=\frac{1}{6}\left(6a_{43}c_3-c_4^3+c_4\right),\ a_{42}=\frac{1}{6}\left(-6a_{43}c_3-6a_{43}+c_4^3+3c_4^2+2c_4\right),$$

$$a_{51}=\frac{1}{6}\left(6a_{53}c_3+6a_{54}c_4+c_4^3-c_4\right),\ a_{52}=\frac{1}{6}\left(\begin{array}{c}-6a_{53}c_3-6a_{53}-6a_{54}c_4-6a_{54}\\-c_4^3+3c_4^2-2c_4\end{array}\right),$$

$$a_{61}=\frac{1}{6}\left(\begin{array}{c}6a_{63}c_3+6a_{64}c_4\\-6a_{65}c_4+c_5^3-c_5\end{array}\right),\ a_{62}=\frac{1}{6}\left(\begin{array}{c}-6a_{63}c_3-6a_{63}-6a_{64}c_4-6a_{64}\\+6a_{65}c_4-6a_{65}-c_5^3+3c_5^2-2c_5\end{array}\right),$$

$$a_{71}=\frac{1}{6}\left(\begin{array}{c}6a_{73}c_3+6a_{74}c_4-6a_{75}c_4\\-6a_{76}c_5-c_5^3+c_5\end{array}\right),\ a_{72}=\frac{1}{6}\left(\begin{array}{c}-6a_{73}c_3-6a_{73}-6a_{74}c_4-6a_{74}\\+6a_{75}c_4-6a_{75}+6a_{76}c_5\\-6a_{76}+c_5^3+3c_5^2+2c_5\end{array}\right),$$

$$\begin{aligned}a_{81}&=a_{83}c_3+a_{84}c_4-a_{85}c_4-a_{86}c_5+a_{87}c_5,\\a_{82}&=a_{83}(-c_3)-a_{83}-a_{84}c_4-a_{84}+a_{85}c_4-a_{85}+a_{86}c_5-a_{86}-a_{87}c_5-a_{87}+1.\end{aligned}$$

In our previous work [13] we present exhausting details about truncation error terms and their derivation. Coleman [14] proposed the expression of local truncation terms by using B2 series and connected its representation with T2 rooted trees.

In [13] we made the choice

$$c_3=-\frac{17}{19},\ c_4=\frac{5}{6},\ c_5=-\frac{17}{19}.\ a_{64}=\frac{2}{3},$$

and attained the as norm of the principal truncation error terms

$$\|T^{(9)}\|_2\approx 6.7\times 10^{-4}.$$

We name the resulting method ACM17.

A little later in [15] we selected

$$c_3=\frac{10,061,236,723,712,997}{11,558,051,517,695,875},\ c_4=-\frac{2,829,529,861,714,855}{10,654,190,333,740,618},$$

$$c_5=-\frac{27,575,926,752,714,835}{24,688,741,064,860,472},\ a_{64}=-\frac{22,870,801,009,117,007}{9,387,744,870,410,575}.$$

and achieved high phase lag for best integrating problems with periodic solutions. We named this method PL18. Traditionally we try to get better methods by minimizing the

principal truncation error. Assuming that all nodes lay in the interval $[-1, 1]$ we may attain the almost global minimum
$$\|T^{(9)}\|_2 \approx 1.39 \times 10^{-4},$$
selecting
$$c_3 = -0.3868070797478156,\ c-4 = -0.6584162991759234,$$
$$c_5 = -0.2932375941564522,\ a_{64} = -0.1290369411904927$$

We name this method MIN.

The technique of minimizing some norm of the terms of principal truncation error terms is very general. Here, we suggest a different technique for choosing c_3, c_4, c_5, and a_{64} that exploits on the behavior of the scheme in a certain class of problems.

3. Performance of the Schemes in a Set of Keplerian-like Problems

We intend to build a special Numerov-type method from the preceding family. The resulting scheme must outperform all other methods on Keplerian-type problems. For testing we have chosen the following problems.

1. *The Kepler problem*

$$\begin{aligned} {}^1y'' &= -\frac{{}^1y}{\left(\sqrt{({}^1y)^2 + ({}^2y)^2}\right)^3}, \\ {}^2y'' &= -\frac{{}^2y}{\left(\sqrt{({}^1y)^2 + ({}^2y)^2}\right)^3}, \end{aligned}$$

with $t \in [0, 10\pi]$, $y(0) = [1-\tau, 0]^T$ and $y'(0) = \left[0, \sqrt{\frac{1+\tau}{1-\tau}}\,\right]^T$. The theoretical solution is [16]
$${}^1y(t) = \cos(v) - \tau,\ {}^2y(t) = \sin(v)\sqrt{1-\tau^2}.$$

In the above, $v = \tau \cdot \sin(v) + x$, τ is the eccentricity, and the the left superscript denotes the components of y. They shall not be confused with $y_1 = \left[{}^1y_1, {}^2y_1, {}^3y_1, {}^4y_1\right]^T$, $y_2 = \left[{}^1y_2, {}^2y_2, {}^3y_2, {}^4y_2\right]^T, y_3, \cdots$, that correspond to the vectors approximating the solution at $t_1, t_2, t_3, \cdots$.

We ran this problem for five different eccentricities (i.e., $\tau = 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}$). In parallel we recorded the function evaluations wasted along with errors at the endpoint $t_{end} = 10\pi$. The results for these problems made by the methods ACM17, PL18, and MIN can be found in Table 1.

Table 1. Accurate digits observed at the end-point by the three methods on classical Kepler for $x \in [0, 10\pi]$.

e	Steps	ACM17	PL18	MIN
0.0	60	2.8	2.6	2.5
	120	5.4	5.0	5.1
	180	6.9	6.4	6.7
	240	8.0	7.4	7.9
	300	8.8	8.2	8.8
	360	9.5	8.8	9.5
	420	10.1	9.4	10.1
0.2	80	2.2	1.8	2.4
	160	4.8	4.3	4.9
	240	6.4	5.8	6.5
	320	7.5	6.9	7.6
	400	8.4	7.8	8.5
	480	9.1	8.5	9.2
	560	9.8	9.0	9.8

Table 1. *Cont.*

e	Steps	ACM17	PL18	MIN
0.4	150	2.3	1.9	2.6
	300	4.9	4.4	5.1
	450	6.5	5.9	6.7
	600	7.6	7.0	7.8
	750	8.5	7.9	8.7
	900	9.2	8.6	9.4
	1050	9.9	9.2	10.0
0.6	200	0.7	0.3	1.3
	400	3.1	2.6	3.4
	600	4.6	4.1	4.9
	800	5.7	5.2	6.0
	1000	6.6	6.0	6.8
	1200	7.3	6.7	7.5
	1400	7.9	7.3	8.1
0.8	500	0.0	−0.2	1.1
	1000	1.9	1.5	2.3
	1500	3.4	2.9	3.7
	2000	4.5	4.	4.8
	2500	5.4	4.8	5.6
	3000	6.1	5.5	6.3
	3500	6.7	6.1	6.9

2. *The Perturbed Kepler*

According to Einstein's general relativity theory, the Schwarzschild potential is utilized to describe the motion of a planet. Then the equations are given by:

$$
\begin{aligned}
{}^1y'' &= -\frac{{}^1y}{\sqrt{({}^1y)^2+({}^2y)^2}^3} - (2+\delta)\delta\frac{{}^1y}{\sqrt{({}^1y)^2+({}^2y)^2}^5}, \\
{}^2y'' &= -\frac{{}^2y}{\sqrt{({}^1y)^2+({}^2y)^2}^3} - (2+\delta)\delta\frac{{}^2y}{\sqrt{({}^1y)^2+({}^2y)^2}^5},
\end{aligned}
$$

and the analytical solution is

$$
{}^1y = \cos(t+\delta t),\ {}^2y = \sin(t+\delta t).
$$

We solved this problem for $\delta = 0.01, 0.03, 0.05, 0.07, 0.09$ in the interval $[0, \frac{10\pi}{1+\delta}]$. The results for these problems made by the methods ACM17, PL18, and MIN can be found in Table 2.

Table 2. Accurate digits observed at the end-point by the three methods on perturbed Kepler.

δ	Steps	ACM17	PL18	MIN
0.01	50	2.0	2.1	1.8
	100	4.7	4.3	4.4
	150	6.2	5.7	6.0
	200	7.3	6.7	7.1
	250	8.1	7.5	8.0
	300	8.8	8.2	8.7
	350	9.4	8.7	9.3

Table 2. *Cont.*

δ	Steps	ACM17	PL18	MIN
0.03	50	2.0	2.0	1.7
	100	4.6	4.2	4.3
	150	6.2	5.6	5.9
	200	7.2	6.7	7.0
	250	8.1	7.5	7.9
	300	8.7	8.1	8.7
	350	9.3	8.7	9.3
0.05	50	1.9	2.0	1.6
	100	4.5	4.2	4.2
	150	6.1	5.6	5.8
	200	7.2	6.6	7.0
	250	8.0	7.4	7.9
	300	8.7	8.1	8.6
	350	9.3	8.6	9.2
0.07	60	2.5	2.5	2.2
	120	5.2	4.8	4.9
	180	6.7	6.2	6.5
	240	7.8	7.2	7.6
	300	8.6	8.0	8.5
	360	9.3	8.7	9.3
	420	9.9	9.2	9.9
0.09	60	2.4	2.4	2.2
	120	5.1	4.7	4.8
	180	6.6	6.2	6.4
	240	7.7	7.2	7.6
	300	8.6	8.0	8.4
	360	9.2	8.6	9.2
	420	9.8	9.2	9.8

3. *The Arenstorf orbit*

Another interesting orbit is a restricted three body problem (sun–earth–moon) described by the equations of motion (see [17], p. 296).

$$\begin{aligned} {}^1y'' &= \zeta' \cdot \frac{q_1(t) - {}^1y}{P_1} + \zeta \cdot \frac{d_1(t) - {}^1y}{P_2}, \\ {}^2y'' &= \zeta' \cdot \frac{q_2(t) - {}^2y}{P_1} + \zeta \cdot \frac{d_2(t) - {}^2y}{P_2}, \end{aligned}$$

where

$$\begin{aligned} P_1 &= \left(\sqrt{({}^1y - q_1(t))^2 + ({}^2y - q_2(t))^2}\right)^3, \\ P_2 &= \left(\sqrt{({}^1y - d_1(t))^2 + ({}^2y - d_2(t))^2}\right)^3, \end{aligned}$$

with

$$\begin{aligned} \zeta &= 0.012277471, \zeta' = 0.987722529, \\ q_1(t) &= -\zeta \cos t, \quad q_2(t) = -\zeta \sin t, \\ d_1(t) &= \zeta' \cos t, \quad d_2(t) = \zeta' \sin t. \end{aligned}$$

The initial values

$$ {}^1y(0) = 0.994,\ {}^1y'(0) = 0,\ {}^2y(0) = 0,\ {}^2x'(0) = -1.00758510637908252, $$

and with $t_A = 17.0652165601579625589, 2t_A,\ 3t_A, \cdots$ the solution is periodic.

The results for this problems made by the methods ACM17, PL18, and MIN can be found in Table 3.

Table 3. Accurate digits observed at the end-point by the three methods on Arenstorf. On the top the results after one period t_A, on the bottom after two periods $2t_A$.

t_{end}	Steps	ACM17	PL18	MIN
t_A	10,000	3.7	2.7	4.2
	15,000	4.8	4.0	5.4
	20,000	5.8	5.0	6.4
	25,000	6.5	5.8	7.2
	30,000	7.2	6.4	7.8
	35,000	7.7	6.9	8.4
	40,000	8.2	7.4	8.9
$2t_A$	10,000	−1.9	−1.0	0.1
	20,000	1.3	0.2	1.8
	30,000	2.5	1.8	3.1
	40,000	3.4	2.6	4.0
	50,000	4.2	3.4	4.8
	60,000	4.8	4.0	5.5
	70,000	5.3	4.5	6.0

4. *The Pleiades*

Finally, we tried the "Pleiades" motion which is presented in ([17], p. 245).

$$ {}^{i}x'' = \sum_{i\neq j} \frac{\mu_j\left({}^{j}x - {}^{i}x\right)}{\rho_{ij}},\ {}^{i}z'' = \sum_{i\neq j} \frac{\mu_j\left({}^{j}z -{}^{i} z\right)}{\rho_{ij}}, $$

with

$$ \rho_{ij} = \sqrt{\left({}^{i}x - {}^{j}x\right)^2 + \left({}^{i}z - {}^{j}z\right)^2}^{3}, i, j = 1, \cdots, 7. $$

The initial values are

$$ {}^{1}x(0) = 3,\ {}^{2}x(0) = 3,\ {}^{3}x(0) = -1,\ {}^{4}x(0) = -3,\ {}^{5}x(0) = 2,\ {}^{6}x(0) = -2,\ {}^{7}x(0) = 2, $$

$$ {}^{1}z(0) = 3,\ {}^{2}z(0) = -3,\ {}^{3}z(0) = 2,\ {}^{4}z(0) = 0,\ {}^{5}z(0) = 0,\ {}^{6}z(0) = -4,\ {}^{7}z(0) = 4, $$

$$ {}^{1}x'(0) = 0,\ {}^{2}x'(0) = 0,\ {}^{3}x'(0) = 0,\ {}^{4}x'(0) = 0,\ {}^{5}x'(0) = 0,\ {}^{6}x'(0) = 1.75,\ {}^{7}x'(0) = -1.5, $$

$$ {}^{1}z'(0) = 0,\ {}^{2}z'(0) = 0,\ {}^{3}z'(0) = 0,\ {}^{4}z'(0) = -1.25,\ {}^{5}z'(0) = 1,\ {}^{6}z'(0) = 0,\ {}^{7}z'(0) = 0, $$

We set $\mu_j = j,\ j = 1, \cdots, 7$. We tried $t_{end} = 3$ and 4 as end-points. The solution there was made with `Mathematica` [18] at high precision. Then we recorded the errors made at $t_{end} = 3$ and $t_{end} = 4$ by the various methods. The results for this problems made by the methods ACM17, PL18, and MIN can be found in Table 4.

Table 4. Accurate digits observed at the end-point by the three methods on Pleiades. On the top for end-point $t_{end} = 3$, on the bottom for $t_{end} = 4$.

t_{end}	Steps	ACM17	PL18	MIN
3	3000	2.4	2.0	3.1
	3000	2.4	2.0	3.1
	4500	3.8	3.3	4.2
	6000	4.9	4.3	5.2
	7500	5.7	5.1	6.0
	9000	6.4	5.8	6.7
	10,500	7.0	6.3	7.3
	12,000	7.5	6.8	7.8
4	4000	1.9	1.4	2.6
	6000	3.3	2.8	3.7
	8000	4.	3.8	4.7
	10,000	5.2	4.6	5.5
	12,000	5.9	5.3	6.2
	14,000	6.5	5.9	6.8
	16,000	7.0	6.3	7.3

In consequence, we set 14 problems (i.e., 5 Keplerian, 5 perturbed Kepler, 3 Arenstorf orbits, and 3 Pleiades) to run for 7 different steplengths each. This sums to 98 runs in total. The average accurate digits (after all these 98 runs) achieved are 5.98, 5.44, and 6.13 for the methods ACM17, PL18, and MIN, respectively. This means that the method with minimal truncation error behaves rather better.

However, our issue here is to find free parameters in the algorithm presented in the previous section so that it provides a method that works as best as possible.

4. Training the Free Parameters in a Wide Set of Keplerian-like Problems

In [19] we may find the origin of our idea. After choosing the free parameters c_3, c_4, c_5, a_{64}, we get a method named NEW8 and form Tables like those presented in the previous section. There, we record the accurate digits achieved using exactly the same steplengths and calculate the average of the 98 digits found there. This average serves as a fitness value and we intend to maximize it. The Differential Evolution algorithm was applied as maximization process [20]. We actually used MATLAB [21] and the suite of routines in DeMat [22] for the implementation of Differential Evolution technique.

We have already used the Differential Evolution process to generate numerical approaches for handling IVP, with extremely intriguing results. However, until now, we tried optimization on a small set (one or two) of problems. On the contrary here we extended the approach by using 98 runs.

The optimization via differential evolution produced a number of quadruplets for the parameters, and we provide the chosen one, in double precision, below,

$$c_3 = -0.4821271178014236, \; c_4 = -0.1599331990972641,$$

$$c_5 = -0.81752579390977, \; a_{64} = 2.118887522290334$$

The resulting pair is presented in the Appendix A as part of a MATLAB function where we implemented the new method.

Using the free parameters listed exactly above, we obtained the end point accuracies tabulated in Tables 5–8.

Table 5. Accurate digits observed at the end-point by the new method on classical Kepler for $x \in [0, 10\pi]$.

e	Steps	NEW8
0.0	60	3.8
	120	6.5
	180	8.2
	240	9.4
	300	10.5
	360	11.6
	420	12.6
0.0	80	4.2
	160	5.8
	240	7.0
	320	8.0
	400	8.7
	480	9.3
	560	9.8
0.0	150	3.5
	300	6.3
	450	7.3
	600	8.2
	750	8.9
	900	9.5
	1050	10.0
0.6	200	1.6
	400	4.2
	600	6.4
	800	7.0
	1000	7.5
	1200	8.0
	1400	8.5
0.8	500	0.6
	1000	2.9
	1500	4.5
	2000	5.9
	2500	7.4
	3000	7.7
	3500	8.7

Table 6. Accurate digits observed at the end-point by the new method on perturbed Kepler.

δ	Steps	NEW8
0.01	50	3.1
	100	5.8
	150	7.4
	200	8.7
	250	9.7
	300	10.6
	350	11.8
0.03	50	3.3
	100	5.9
	150	7.6
	200	8.9
	250	10.0
	300	11.6
	350	11.3
0.05	50	3.6
	100	6.1
	150	7.9
	200	9.5
	250	10.3
	300	10.5
	350	10.9
0.07	60	4.9
	120	8.6
	180	8.7
	240	9.5
	300	10.2
	360	10.8
	420	11.2
0.09	60	4.0
	120	6.7
	180	8.2
	240	9.2
	300	9.9
	360	10.5
	420	11.1

Table 7. Accurate digits observed at the end-point for the new method on Arenstorf. In the left the results after one period t_A, in the right after two periods $2t_A$.

Steps	NEW8	Steps	NEW8
10,000	3.8	10,000	1.1
15,000	5.4	20,000	1.6
20,000	6.7	30,000	3.2
25,000	7.6	40,000	4.5
30,000	8.4	50,000	5.8
35,000	9.1	60,000	7.1
40,000	9.7	70,000	8.8

Table 8. Accurate digits observed at the end-point by the new method on Pleiades. On the top for end-point $t_{end} = 3$, on the bottom for $t_{end} = 4$.

Steps	NEW8
3000	3.1
4500	4.3
6000	5.3
7500	6.1
9000	6.8
10,500	7.3
12,000	7.8
Steps	**NEW8**
4000	2.6
6000	3.8
8000	4.9
10,000	5.7
12,000	6.3
14,000	6.9
16,000	7.4

The average accuracy obtained for the 98 runs is 7.25, i.e., more than one digit was gained over the optimal method MIN. The norm of the principal truncation error coefficients for NEW8 is

$$\|T^{(7)}\|_2 \approx 5.77 \cdot 10^{-4},$$

which is four times bigger than the corresponding norm of method MIN. On the other hand, no extra property seems to hold. No high phase-lag, interval of periodicity, symplecticness, etc.

5. Numerical Tests

We tested NEW8, presented in the previous section along with method MIN which possesses a minimal truncation error. Both pairs were run for the same problems and steplengths. In prior parts and for optimization, we preferred using the end-point error because it decreased evaluation times significantly.

In the numerical testing, the errors are assessed throughout the entire grid on the integration interval. Every step, a parallel integration with strict tolerance and an eighth order Runge–Kutta–Nyström pair [23] is used to estimate the true solution. Thus, an almost true global error is registered. Here, we use the same problems but change the parameters and the integration intervals while we report global errors in the results.

In the following the Kepler orbits were run in the interval $[0, 20\pi]$ with eccentricities $\tau = 0.1, 0.3, 0.5, 0.7$, and 0.9, respectively. The results were recorded in Table 9.

We also run the perturbed Kepler orbits in the interval $[0, \frac{20\pi}{1+\delta}]$ with parameters $\delta = 0.02, 0.04, 0.06, 0.08, 0.1$, respectively. The results for these problems are recorded in Table 10.

The global errors for Arenstorf are recorded in Table 11 over the intervals $[0, 0.75t_A]$ and $[0, 1.25t_A]$. Finally, the global errors for Pleiades are recorded in Table 12 over the intervals $[0, 4.5]$ and $[0, 5.5]$. We did not go any further since quasi-collisions ruin the results.

Table 9. Final numerical tests. Accurate digits observed over all grid points on classical Kepler in the interval $[0, 20\pi]$.

e	Steps	NEW8	MIN
0.1	2· 60	3.1	1.5
	2· 120	5.4	4.0
	2· 180	6.7	5.6
	2· 240	7.6	6.7
	2· 300	8.4	7.6
	2· 360	9.0	8.3
	2· 420	9.5	8.9
0.3	2· 80	1.6	1.0
	2· 160	4.2	3.3
	2· 240	5.8	4.8
	2· 320	7.0	5.9
	2· 400	7.8	6.8
	2· 480	8.4	7.5
	2· 560	8.8	8.1
0.5	2· 150	1.3	1.0
	2· 300	3.8	3.2
	2· 450	5.5	4.7
	2· 600	6.8	5.8
	2· 750	7.8	6.7
	2· 900	8.4	7.4
	2100	8.8	8.0
0.7	2· 200	0.2	-0.3
	2· 400	1.4	1.1
	2· 600	2.8	2.4
	2· 800	3.9	3.4
	2000	4.8	4.2
	2400	5.6	4.9
	2800	6.3	5.5

Table 9. *Cont.*

e	Steps	NEW8	MIN
0.9	2000	2.2	1.8
	3000	3.2	2.7
	4000	4.0	3.5
	5000	4.7	4.1
	6000	5.3	4.6
	7000	5.8	5.1
	8000	6.3	5.5

Table 10. Final numerical tests. Accurate digits observed over all grid points on perturbed Kepler in the interval $[0, \frac{20\pi}{1+\delta}]$.

δ	Steps	NEW8	MIN
0.02	$2 \cdot 50$	2.6	1.1
	$2 \cdot 100$	5.2	3.7
	$2 \cdot 150$	6.7	5.3
	$2 \cdot 200$	7.9	6.4
	$2 \cdot 250$	8.8	7.3
	$2 \cdot 300$	9.6	8.0
	$2 \cdot 350$	10.3	8.6
0.04	$2 \cdot 50$	2.8	1.1
	$2 \cdot 100$	5.3	3.7
	$2 \cdot 150$	6.9	5.2
	$2 \cdot 200$	8.0	6.4
	$2 \cdot 250$	9.0	7.2
	$2 \cdot 300$	9.8	8.0
	$2 \cdot 350$	10.5	8.6
0.06	$2 \cdot 50$	3.6	1.0
	$2 \cdot 100$	5.5	3.6
	$2 \cdot 150$	7.1	5.2
	$2 \cdot 200$	8.4	6.3
	$2 \cdot 250$	9.4	7.2
	$2 \cdot 300$	10.3	7.9
	$2 \cdot 350$	10.7	8.5
0.08	$2 \cdot 60$	3.7	1.6
	$2 \cdot 120$	7.0	4.2
	$2 \cdot 180$	8.3	5.8
	$2 \cdot 240$	9.2	6.9
	$2 \cdot 300$	9.9	7.8
	$2 \cdot 360$	10.5	8.6
	$2 \cdot 420$	11.0	9.2

Table 10. *Cont.*

δ	Steps	NEW8	MIN
0.10	$2 \cdot 60$	3.2	1.5
	$2 \cdot 120$	6.1	4.2
	$2 \cdot 180$	7.6	5.7
	$2 \cdot 240$	8.7	6.9
	$2 \cdot 300$	9.5	7.8
	$2 \cdot 360$	10.1	8.5
	$2 \cdot 420$	10.6	9.1

Table 11. Final numerical tests. Accurate digits observed over all grid points on Arenstorf. On the top the results for the interval $[0, 0.75t_A]$, on the bottom for the interval $[0, 1.25t_A]$.

Steps	NEW8	MIN
5000	2.6	3.2
10,000	5.1	5.2
15,000	6.9	6.5
20,000	7.9	7.5
25,000	8.7	8.4
30,000	9.3	9.0
35,000	9.8	9.6
Steps	**NEW8**	**MIN**
10,000	1.5	2.1
20,000	4.3	4.1
30,000	6.4	5.5
40,000	8.0	6.6
50,000	8.2	7.4
60,000	8.7	8.1
70,000	9.4	8.7

The overall average accuracy observed was 6.6 digits for NEW8 and 5.6 digits for MIN, which is also a very good result. We also remark that we are able to generate a number of schemes with coefficients that are similar to the one shown. This means that the distance was no greater than, say, $10 - 3$. These schemes also performed well. Perhaps only 5% to 10% worse than the method given here. This signifies that no strict property, such as order conditions, conservation rules, or other symplectic property, holds for the new method. Our new recommended method falls inside a range of coefficients that appears to be suitable for the type of Keplerian-type obits we are interested in here.

Table 12. Final numerical tests. Accurate digits observed over all grid points on Pleiades. On the top the results for the interval $[0, 4.5]$, on the bottom for the interval $[0, 5.5]$.

Steps	NEW8	MIN
5000	2.7	2.7
7500	4.1	4.0
10,000	5.1	5.0
12,500	5.9	5.8
15,000	6.6	6.5
17,500	7.2	7.1
20,000	7.6	7.6
Steps	**NEW8**	**MIN**
7000	3.4	3.3
10,500	4.8	4.7
14,000	5.9	5.7
17,500	6.7	6.6
21,000	7.3	7.3
24,500	7.9	7.8
28,000	8.4	8.4

6. Conclusions

In a family of hybrid two-step techniques, we suggested a method for better selecting its free parameters. After testing their performance in a large number of Keplerian-type orbits, these parameters were chosen. The Differential Evolution technique was used for obtaining an almost optimal choice. In other sets of Keplerian-type orbits, the derived scheme outperformed other methods from the same family by a significant margin.

Author Contributions: Conceptualization, V.N.K., T.E.S. and C.T.; Data curation, V.N.K., R.V.F., A.V.C. and T.E.S.; Formal analysis, V.N.K., R.V.F., T.E.S., A.V.C. and C.T.; Funding acquisition, T.E.S.; Investigation, V.N.K., R.V.F., T.E.S., A.V.C. and C.T.; Methodology, T.E.S. and C.T.; Project administration, T.E.S.; Resources, T.E.S., A.V.C. and C.T.; Software, R.V.F., A.V.C. and C.T.; Supervision, T.E.S.; Validation, V.N.K., A.V.C. and T.E.S.; Visualization, V.N.K., R.V.F. and T.E.S.; Writing—original draft, C.T.; Writing— review & editing, T.E.S. and C.T. All authors have 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 the 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.

Appendix A

We present a listing with a naive program for the new method (NEW8). The coefficients are rounded and are suitable for double-precision arithmetic. We made use of the technique described in ([17], p. 471) in order to avoid numerical instabilities. We remark that $y_1 \approx y(t_0 + h)$ represented by the variable `y1` can be found using a Runge–Kutta–Nyström pair [23]. The function `rkn86` can be retrieved from [24].

```
%-------------------------------------------------------------------- begin
function [x,y]=numer8(fucn,t0,te,y0,dy0,n);

% explicit 8th order, 7 stages method, for addressing y'' = f(x,y)
%
% Input
% t0, te : left and right points of integration
% y0, dy0: initial y and y' (at t0)
% n      : number of steps
%
% The coefficients
b=[-0.011910630531427863, -1.4152390130922559, 0, 1.1198831773307117, ...
    1.1198831773307117, 0.099646959746844095, 0.099646959746844095,   ...
                                           -0.011910630531427863];
c=[-1, 0, -0.48212711780142360, -0.15993319909726412,                 ...
     0.15993319909726412, 0.81752579390976997, -0.81752579390976997, 1]';
a=[0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0;
-0.061676388147542510, -0.063163891893415396, 0, 0, 0, 0, 0, 0;
-0.001449407926829631, -0.014860974640587388, -0.050866902894472477,  ...
                                                         0, 0, 0, 0, 0;
0.0012884760471727602, 0.042761762969669080, 0.052439198342644856,    ...
                                   -0.0037335237241120772, 0, 0, 0, 0;
0.036564037809900442, -2.9816788795117797, -0.12349939054047346,      ...
                         2.1188875222903341, 1.6926638187608034, 0, 0, 0;
-0.028514259688726427, 1.1813134649095517, 0.10483959970071562,       ...
  -0.85285968590356044, -0.49075320588562187, 0.011385401766656327, 0, 0;
0.052214784939110816, -6.3487950094855168, -0.0082786720847229343,    ...
3.7999377812747299, 3.6145591840867179, -0.0071926442865628577,       ...
-0.10244542444375599, 0];

s=length(c);                                     % no of stages
h=(te-t0)/n;                                     % step length
m=length(y0);                                    % dimension of system
x=[t0 t0+h zeros(1,n-1)]';                       % output of t
y=zeros(m,n+1);                                  % output of y
[x1,y1]=rkn86(fucn,t0,t0+h,y0,dy0,3e-14);       % initial y1
y(:,1)=y0;
y(:,2)=y1(end,:)';
F=zeros(m,s);
f1=feval(fucn,t0,y0);

hu0=(y(:,2)-y(:,1));             % use device in Hairer et al, pg~471

for k=2:n,
  f0=f1;
  F(:,1)=f0;
  f1=feval(fucn,x(k),y(:,k));    % The first stage
  F(:,2)=f1;
  for o=3:s, % Another s-2 stages
    F(:,o)=feval(fucn,x(k)+c(o)*h,(1+c(o))*y(:,k)-c(o)*y(:,k-1)  ...
                                                 +h*h*F*a(o,:)');
  end;
  hu1=hu0+h*h*F*b';
```

```
  y(:,k+1)=hu1+y(:,k);
  x(k+1)=x(k)+h;
  hu0=hu1;
end;
%--------------------------------------------------------------------  end
```

Then we verify a test presented above with perturbed Kepler, e.g., the last rightmost entry in Table 6 for NEW8 (i.e., for $\delta = 0.09$ and after 420 steps) can be found typing:

```
>> fucn=@(x,y) [-y(1)/sqrt(y(1)^2+y(2)^2)^3                              ...
       -(2+0.09)*0.09*y(1)/sqrt(y(1)^2+y(2)^2)^5;
               -y(2)/sqrt(y(1)^2+y(2)^2)^3                              ...
       -(2+0.09)*0.09*y(2)/sqrt(y(1)^2+y(2)^2)^5]
>> [xout,yout] = numer8(fucn, 0, 10*pi/(1+0.09), [1 0]',[0 1+0.09]',420);
>> -log10(max(abs([1 0]'-yy(:,end))))
ans =
   11.0680
```

which is rounded to 11.1 as shown in that table.

References

1. Hairer, E. Unconditionally stable methods for second order differential equations. *Numer. Math.* **1979**, 32, 373–379. [CrossRef]
2. Cash, J.R. High order P–stable formulae for the numerical integration of periodic initial value problems. *Numer. Math.* **1981**, 37, 355–370. [CrossRef]
3. Chawla, M.M. Two–step fourth order P–stable methods for second order differential equations. *BIT* **1981**, 21, 190–193. [CrossRef]
4. Chawla, M.M. Numerov Made Explicit has Better Stability. *BIT* **1984**, *24*, 117–118. [CrossRef]
5. Tsitouras, C. Explicit Numerov type methods with reduced number of stages. *Comput. Math. Appl.* **2003**, 45, 37–42. [CrossRef]
6. Chawla, M.M.; Rao, P.S. An explicit sixth–order method with phase–lag of order eight for $y'' = f(t, y)$. *J. Comput. Appl. Math.* **1987**, 17, 365–368. [CrossRef]
7. Tsitouras, C. Explicit eighth order two–step methods with nine stages for integrating oscillatory problems. *Int. J. Modern Phys. C* **2006**, 17, 861–876. [CrossRef]
8. Franco, J.M. A class of explicit two-step hybrid methods for second-order IVPs. *J. Comput. Appl. Math.* **2006**, *187*, 41—57. [CrossRef]
9. Franco, J.M.; Randez, L. Explicit exponentially fitted two-step hybrid methods of high order for second-order oscillatory IVPs. *Appl. Math. Comput.* **2016**, *273*, 493–505. [CrossRef]
10. Franco, J.M.; Randez, L. Eighth-order explicit two-step hybrid methods with symmetric nodes and weights for solving orbital and oscillatory IVPs. *Int. J. Modern. Phys. C* **2018**, 29, 1850002. [CrossRef]
11. Butcher, J.C. Implicit Runge Kutta processes. *Math. Comput.* **1964**, 18, 50–64 . [CrossRef]
12. Butcher, J.C. On Runge–Kutta processes of high order. *J. Austral. Math. Soc.* **1994** 4, 179–194. [CrossRef]
13. Simos, T.E.; Tsitouras, Ch.; Famelis, I.T. Explicit Numerov Type Methods with Constant Coefficients: A Review. *Appl. Comput. Math.* **2017**, 16, 89–113.
14. Coleman, J.P. Order conditions for a class of two-step methods for $y'' = f(x, y)$. *IMA J. Numer. Anal.* **2003**, 23, 197–220. [CrossRef]
15. Simos, T.E.; Tsitouras, C. A new family of seven stages, eighth order explicit Numerov–type methods. *Math. Meth. Appl. Sci.* **2017**, 40, 7867–7878. [CrossRef]
16. Hull, T.E; Enright, W.H.; Fellen, B.M.; Sedgwick, A.E. Comparing Numerical Methods for Ordinary differential Equations. *SIAM J. Numer. Anal.* **1972**, 9, 603–637. [CrossRef]
17. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I, Nonstiff Problems*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1993.
18. Wolfram Research, Inc. *Mathematica, Version 11.1*; Wolfram Research, Inc.: Champaign, IL, USA, 2017.
19. Tsitouras, C. Neural Networks With Multidimensional Transfer Functions. *IEEE Trans. Neural Netw.* **2002**, *13*, 222–228. [CrossRef]
20. 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]
21. The Mathworks, Inc. *MATLAB*; version R2019b; The Mathworks, Inc.: Natick, MA, USA, 2019.
22. DeMat. Available online: https://www.swmath.org/software/24853 (accessed on 23 August 2021).
23. Papakostas, S.N.; Tsitouras, C. High phase-lag order Runge-Kutta and Nyström pairs. *SIAM J. Sci. Comput.* **1999**, 21, 747–763. [CrossRef]
24. `rkn86`. Available online: https://www.mathworks.com/matlabcentral/fileexchange/3153-rkn86 (accessed on 12 October 2021).