

Rational Approximation Method for Stiff Initial Value Problems

Artur Karimov ^{1,*}, Denis Butusov ^{1,*}, Valery Andreev ² and Erivelton G. Nepomuceno ³

¹ Youth Research Institute, St. Petersburg Electrotechnical University “LETI”, 5 Professora Popova St., 197376 Saint Petersburg, Russia

² Department of Computer-Aided Design, St. Petersburg Electrotechnical University “LETI”, 5 Professora Popova St., 197376 Saint Petersburg, Russia; vsandreev@etu.ru

³ Centre for Ocean Energy Research, Department of Electronic Engineering, Maynooth University, W23 F2H6 Maynooth, Ireland; erivelton.nepomuceno@mu.ie

* Correspondence: aikarimov@etu.ru (A.K.); dnbutusov@etu.ru (D.B.)

Abstract: While purely numerical methods for solving ordinary differential equations (ODE), e.g., Runge–Kutta methods, are easy to implement, solvers that utilize analytical derivations of the right-hand side of the ODE, such as the Taylor series method, outperform them in many cases. Nevertheless, the Taylor series method is not well-suited for stiff problems since it is explicit and not A -stable. In our paper, we present a numerical-analytical method based on the rational approximation of the ODE solution, which is naturally A - and $A(\alpha)$ -stable. We describe the rational approximation method and consider issues of order, stability, and adaptive step control. Finally, through examples, we prove the superior performance of the rational approximation method when solving highly stiff problems, comparing it with the Taylor series and Runge–Kutta methods of the same accuracy order.

Keywords: rational approximation; numerical integration; stiff problem; variable step size; ODE; stiff systems



Citation: Karimov, A.; Butusov, D.; Andreev, V.; Nepomuceno, E.G.

Rational Approximation Method for Stiff Initial Value Problems.

Mathematics **2021**, *9*, 3185. <https://doi.org/10.3390/math9243185>

Academic Editor: Marianito Rodrigo

Received: 14 November 2021

Accepted: 8 December 2021

Published: 10 December 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/>).

1. Introduction

The basic principle in designing numerical methods for solving the initial value problem

$$\dot{y} = f(t, y), y(t_0) = y_0 \quad (1)$$

is that the numerical method must fit the Taylor series expansion of the solution in a given point with the desired accuracy [1]. The Taylor series can directly be used for numerical integration as well, and a number of algorithms and software have been developed based on Taylor series expansion [2–5].

In real-life applications, the one-step Taylor series method outperforms purely numerical one-step algorithms such as the Dormand–Prince and Gragg–Bulirsch–Stör methods when extended data types are used and small error tolerances are required. Its high computational efficiency on conventional data types can be achieved by using automatic differentiation for computing the derivatives in the Taylor expansion [3,6].

The main drawback of the explicit Taylor series method is that it can fail on stiff ODEs, i.e., lose stability when the step size is not appropriate or may select small step sizes in adaptive time step implementation, being guided more by stability requirements rather than accuracy limits, which results in performance loss. The formal definition of stiffness can be found in [7], and a vital feature of the numerical method suitable for efficient solving stiff equations is A -stability and its generalizations, e.g., $A(\alpha)$ -stability [8]. The Taylor series method is not A -stable. Still, its stability region grows with its order [9], which may increase the performance of high-order Taylor series methods on moderately stiff problems. Recently, a modification of the Taylor series method for semi-linear problems called the exponential Taylor series method was proposed [10]. This idea can be used to derive Taylor-like methods, which are A -stable and L -stable [11].

In general, the Taylor series approximation can be considered a variant of the solution’s polynomial approximation with respect to time step h .

However, the Taylor series is not the only variant of the function decomposition that allows for constructing integration schemes. Several other examples include the Haar wavelet decomposition and hybrid function decomposition for integer-order ODEs [12], and fractional-order general Lagrange scaling functions for fractional-order ODEs [13]. It is also well known that, in some practical cases, using rational approximation instead of polynomial approximation is more suitable, e.g., when the approximated function contains poles. L. Wuytack was among the first researchers to notice it in application to numerical integration and proposed the second-order numerical method based on the Padé approximation for solving problems with singularities [14]. Later works, e.g., [15], developed this idea. It is also notable that there is a well-known relation between multi-step methods and the Padé approximation [16]. The Padé approximation of a matrix exponential in domain-specific implementations of the exponential Taylor method can be highly efficient [17].

An attempt to analyze the applicability of the Padé type rational approximation methods to the one-dimensional stiff problem was made by M. Gadella and L. Lara [18]. However, the high potential of the rational approximation methods in solving stiff multidimensional problems has still not been investigated.

This paper aims to fill this gap by introducing a family of A - and $A(\alpha)$ -stable rational approximation methods and by studying their performance on stiff problems. The paper is organized as follows. In Section 2, we introduce the theoretical background of the proposed rational approximation method. Section 3 presents the stability analysis and proves that the proposed method is A - or $A(\alpha)$ -stable depending on the order of accuracy. In Section 4, we introduce an adaptive step control technique for this method. In Section 5, we present the experimental results. Section 6 gives the conclusions.

The following special notation is accepted in the paper:

- denotes the Hadamard (element-wise) multiplication;
- ⊗ denotes an arbitrary binary vector operation.

2. Basics of Rational Approximation in Numerical Integration

For a given initial value problem (1), where y is a vector-valued phase variable and t is a scalar (time), the approximate solution obtained in the next time point is denoted as $y_{n+1} \approx y(t_{n+1})$. Denote the time step $h = t_{n+1} - t_n$. Then, consider the Taylor series that approximates the solution:

$$y_{n+1} = \sum_{i=0}^{\infty} \frac{y_n^{(i)}}{i!} h^i. \tag{2}$$

Consider a rational approximation of y_{n+1} as a fraction of two polynomials $P_L(h)$ and $Q_M(h)$:

$$\begin{aligned} P_L(h) &= p_0 + p_1h + \dots + p_Lh^L, \\ Q_M(h) &= q_0 + q_1h + \dots + q_Mh^M, \end{aligned}$$

where L and M are powers of these polynomials. The rational approximation of the solution reads:

$$y_{n+1} = \frac{P_L(h)}{Q_M(h)}. \tag{3}$$

A number of reliable algorithms for finding the rational approximant (3) exist [19]. Here, we use the most straightforward approach. Let $M + L = p$, where p is a certain natural number. Assuming that (2) and (3) must give the same result up to the error term $O(h^{p+1})$, it follows that

$$\sum_{i=0}^{\infty} \frac{y^{(i)}}{i!} h^i - \frac{P_L(h)}{Q_M(h)} = O(h^{p+1}), \tag{4}$$

which results in a system of equations:

$$\begin{cases} p_0 = y_n q_0, \\ p_1 = h y'_n q_0 + y_n q_1, \\ p_2 = h^2 \frac{y''_n}{2} q_0 + h y'_n q_1 + y_n q_2, \\ \vdots \\ p_L = h^L \frac{y_n^{(L)}}{L!} q_0 + h^{L-1} \frac{y_n^{(L-1)}}{(L-1)!} q_1 + \dots + y_n q_L, \\ 0 = h^{L+1} \frac{y_n^{(L+1)}}{(L+1)!} q_0 + h^L \frac{y_n^{(L)}}{L!} q_1 + \dots + h^{L-M+1} \frac{y_n^{(L-M+1)}}{(L-M+1)!} q_M, \\ \vdots \\ 0 = h^{L+M} \frac{y_n^{(L+M)}}{(L+M)!} q_0 + h^{L+M-1} \frac{y_n^{(L+M-1)}}{(L+M-1)!} q_1 + \dots + h^L \frac{y_n^{(L)}}{L!} q_M. \end{cases} \tag{5}$$

One can notice that this system is complete only if $L + M = p$, which is exactly the case of the Padé rational approximation. Similarly, the Padé approximant corresponds to the rational approximation of the highest possible order of accuracy. Usually, the notation $R_{L/M}$ or $[L/M]$ is used to denote the powers of polynomials in the Padé approximant with the power of the numerator L and the power of the denominator M . Otherwise, if $L + M - p = r$, then $r > 0$, the last r equations from (5) should be omitted, and the r coefficients in P and Q remain free. This option can be used to obtain rational approximants with some desired properties, but their order of accuracy is lower than that with the Padé approximant.

2.1. Padé Approximation

The second-order numerical integration method proposed by L. Wuytack is the $[1/1]$ Padé approximation similar to the type in (3). After simplification, it reads:

$$y_{n+1} = y_n + \frac{h f^2(t_n, y_n)}{f(t_n, y_n) - \frac{h}{2} f'(t_n, y_n)}, \tag{6}$$

where $f'(t, y)$ is a time derivative of $f(t, y)$.

The fourth-order Padé numerical method of type $[2/2]$ can be obtained from (5) in a similar way [18]. After simplification, it can be presented as follows:

$$y_{n+1} = y_n + \frac{(24 f''_n f_n^2 - 36 f_n'^2 f_n)h + (24 f''_n f_n' f_n - 6 f_n^{(3)} f_n^2 - 18 f_n'^3)h^2}{-36 f_n'^2 + 24 f''_n f_n + (12 f''_n f_n' - 6 f_n^{(3)} f_n)h + (-4 f_n''^2 + 3 f_n^{(3)} f_n')h^2} \tag{7}$$

where $f_n^{(q)} = f^{(q)}(t_n, y_n)$ for more compact notation.

Two problems are evident when working with numerical methods such as (7). First, this formula can be straightforwardly implemented in a one-dimensional case, but when y_n and f_n are vectors, some special techniques are needed to perform the division of two vector values, appearing in the numerator and denominator of (7). The classical definition of vector space does not include vector division, but some authors [14,20,21] propose using element-wise operation for this purpose:

$$\frac{\mathbf{v}}{\mathbf{w}} = \left(\frac{v_1}{w_1} \dots \frac{v_i}{w_i} \dots \frac{v_m}{w_m} \right)^T. \tag{8}$$

Alas, there is no proof that such an operation can preserve A - and $A(\alpha)$ -stability even when the formula is A - or $A(\alpha)$ -stable for a one-dimensional problem [22]. Moreover, it was shown that this approach might notably affect the dynamical properties of the simulated system [21].

The second problem of the Padé approximant is that, starting with the [2/2] formula, it becomes cumbersome and difficult to derive, implement, and compute.

2.2. Proposed Approximation

Reducing the rational approximation order may result in more feasible types of approximation. In this paper, we focus on a class of methods given by $[p - 1]/[p - 1]$ approximants for even p and $[p]/[p - 1]$ for odd p . The method of order 2 of this class is (6), similar to the Padé approximant. The Methods of orders 3–6 are given in Table 1. These formulas were obtained from the equations in (5) with an additional condition: the denominator of these rational approximants Q should represent the Taylor series for $(y(t_n - h) - y(t_n))/h$:

$$Q = f(t_n, y_n) - \frac{h}{2}f'(t_n, y_n) + \frac{h^2}{6}f''(t_n, y_n) + \dots$$

Denoting the numerator P , one can notice that, in such types of approximation, the numerator is always a sum of two terms:

$$P = y_n\Phi_{-h}(t_n, y_n) + h\Theta_h(t_n, y_n),$$

where $\Phi_{-h}(t_n, y_n) = Q$ and $\Theta_h(t_n, y_n)$ is another function of t_n and y_n . Therefore, we can write the general form of these type methods similar to conventional one-step methods:

$$y_{n+1} = y_n + \frac{h\Theta_h(t_n, y_n)}{\Phi_{-h}(t_n, y_n)}. \tag{9}$$

One obvious advantage of the proposed rational approximation (RA) method is that they have a more straightforward formulation than Padé approximation methods. Another advantage is that they not only can be A - or $A(\alpha)$ -stable but also can guarantee stability preservation in the multidimensional case, which is considered in the next sections.

Table 1. RA methods of orders 3–6.

p	Formula
3	$y_{n+1} = y_n + \frac{hf_n^2 + \frac{h^3}{12}(4f_n''f_n - f_n'^2)}{f_n - \frac{h}{2}f_n' + \frac{h^2}{6}f_n''}$
4	$y_{n+1} = y_n + \frac{hf_n^2 + \frac{h^3}{12}(4f_n''f_n - 3f_n'^2)}{f_n - \frac{h}{2}f_n' + \frac{h^2}{6}f_n'' - \frac{h^3}{24}f_n^{(3)}}$
5	$y_{n+1} = y_n + \frac{hf_n^2 + \frac{h^3}{12}(4f_n''f_n - 3f_n'^2) + \frac{h^5}{360}(6f_n^{(4)}f_n + 10f_n''^2 - 15f_n^{(3)}f_n')}{f_n - \frac{h}{2}f_n' + \frac{h^2}{6}f_n'' - \frac{h^3}{24}f_n^{(3)} + \frac{h^4}{120}f_n^{(4)}}$
6	$y_{n+1} = y_n + \frac{hf_n^2 + \frac{h^3}{12}(4f_n''f_n - 3f_n'^2) + \frac{h^5}{360}(6f_n^{(4)}f_n + 10f_n''^2 - 15f_n^{(3)}f_n')}{f_n - \frac{h}{2}f_n' + \frac{h^2}{6}f_n'' - \frac{h^3}{24}f_n^{(3)} + \frac{h^4}{120}f_n^{(4)} - \frac{h^5}{720}f_n^{(5)}}$

2.3. Multidimensional Case

Following [4], we rewrite the non-autonomous ODE (1) in a form of an autonomous ODE:

$$Y(t) = (t, y_1, \dots, y_m)^T,$$

$$F(y) = (1, f_1(Y), \dots, f_m(Y))^T,$$

so Equation (1) becomes

$$Y'(t) = F(Y(t)). \tag{10}$$

The first derivatives of $F(Y)$ are as follows:

$$F' = JF,$$

$$F'' = JJF + \frac{\partial J}{\partial Y}FF,$$

$$F''' = \frac{\partial^2 J}{\partial Y^2}FFF + \frac{\partial J}{\partial Y}JFF + 2\frac{\partial J}{\partial Y}FJF + J\frac{\partial J}{\partial Y}FF + JJJF,$$

$$\dots,$$

where $F = F(Y)$ and $J = \partial F / \partial Y$ is the Jacobian matrix. Higher-order derivatives are given in Appendix A.

Furthermore, we show that the RA method can be written in a matrix form:

$$M_{den}\Delta Y_{n+1} = M_{num}F(Y_n), \tag{11}$$

where $\Delta Y_{n+1} = Y_{n+1} - Y_n$, and M_{num} and M_{den} are $d \times d$ matrices, where d is the system dimension after writing it as an autonomous system: $d = m$ if (1) is autonomous and $d = m + 1$ if (1) is non-autonomous. This allows us to solve (11) via the Gauss method in a manner similar to implicit methods performed for a single iteration of the Newton–Raphson procedure, which is important for stability.

First, we prove the following lemma.

Lemma 1. *The i -th derivative of $F(Y)$ can be represented as $M_i F(Y)$, where M_i is a matrix, $i \in \mathbb{Z}_+$.*

Proof. Due to the rules of differentiation,

$$F'(Y(t)) = \frac{\partial F(Y)}{\partial Y(t)} \frac{dY(t)}{dt}. \tag{12}$$

Substituting (10) into (12) leads to

$$F'(Y(t)) = \frac{\partial F(Y)}{\partial Y(t)} F(Y(t)), \tag{13}$$

which already reads as $M_1(Y)F(Y)$, and after differentiating,

$$\frac{d}{dt}(M_1(Y)F(Y)) = \left(\frac{\partial M_1(Y)}{\partial Y(t)} F(Y) \right) F(Y) + M_1^2(Y)F(Y). \tag{14}$$

Collecting terms by $F(Y)$ in (14) as $M_2 = \left(\left(\frac{\partial M_1(Y)}{\partial Y(t)} F(Y) \right) F(Y) + M_1^2(Y) \right)$, we obtain

$$\frac{d}{dt}(M_1(Y)F(Y)) = M_2(Y)F(Y), \tag{15}$$

which is similar to (12) up to the index by M . Continuing this process, we obtain

$$\frac{d}{dt}(M_i(Y)F(Y)) = M_{i+1}(Y)F(Y), \tag{16}$$

which proves the lemma. \square

Then, the following theorem can be proven.

Theorem 1. *RA method of order p*

$$y_{n+1} = y_n + \frac{hf_n^2 + \frac{h^3}{12}(-4f_n''f_n + 3f_n'^2) + \dots}{f_n - \frac{h}{2}f_n' + \frac{h^2}{6}f_n'' + \dots}, \tag{17}$$

for the multidimensional ODE reads in a matrix form as

$$\Delta Y_{n+1} = (M_{den})^{-1}M_{num}F(Y_n), \tag{18}$$

where M_{num} and M_{den} are matrices.

Proof of Theorem 1. Using the notation in (10) and results of Lemma 1, we rewrite (9) for the one-dimensional case as follows:

$$Y_{n+1} = Y_n + \frac{M_{num}F(Y_n)F(Y_n)}{M_{den}F(Y_n)}, \tag{19}$$

where M_{num} collects all terms by $F(Y_n)$ in the numerator and M_{den} collects all terms by $F(Y_n)$ in the denominator. Equation (19) in the multidimensional case reads

$$\Delta Y_{n+1} = (M_{den}F(Y_n))^{-1}M_{num}F(Y_n) \circ F(Y_n), \tag{20}$$

where $\Delta Y_{n+1} = Y_{n+1} - Y_n$ and \circ denotes the Hadamard (element-wise) multiplication. From (20), implying $\|F(Y_n)\| \neq 0$ and using the commutativity of the Hadamard product, we obtain the following the formula:

$$\Delta Y_{n+1} = (M_{den})^{-1}M_{num}F(Y_n), \tag{21}$$

which proves Theorem 1. \square

The second-order RA method (6) in matrix form is as follows:

$$\Delta Y_{n+1} = \left(I - \frac{h}{2}J(Y_n)\right)^{-1} hF(Y_n), \tag{22}$$

where I is the identity matrix, which gives the linearly implicit midpoint method, also known under the abbreviation LIMP [8].

Denote

$$\begin{aligned} F_1 &= J, \\ F_2 &= \frac{\partial J}{\partial Y}F + J^2, \\ F_3 &= \left(\frac{\partial^2 J}{\partial Y \partial Y}F\right)F + \frac{\partial J}{\partial Y}JF + 2\left(\frac{\partial J}{\partial Y}F\right)J + J\left(\frac{\partial J}{\partial Y}F\right) + J^3. \end{aligned}$$

The RA method of order 4 is as follows:

$$\Delta Y_{n+1} = \left(I - \frac{h}{2}F_1 + \frac{h^2}{6}F_2 - \frac{h^3}{24}F_3\right)^{-1} \left(I + \frac{h}{2}\left(-\frac{1}{4}F_1^2 + \frac{1}{3}F_2\right)\right)hF, \tag{23}$$

where zero elements in the matrix are left blank. From (27), the stability functions of RA methods are

$$R(z) = \frac{1 + \frac{z}{2} + \frac{z^2}{6} + \frac{z^3}{24} + \dots + \frac{z^{p-1}}{p!}}{1 - \frac{z}{2} + \frac{z^2}{6} - \frac{z^3}{24} + \dots - \frac{z^{p-1}}{p!}} \tag{28}$$

for even p and

$$R(z) = \frac{1 + \frac{z}{2} + \frac{z^2}{6} + \frac{z^3}{24} + \dots - \frac{z^{p-1}}{p!} + \frac{2z^p}{(p+1)!}}{1 - \frac{z}{2} + \frac{z^2}{6} - \frac{z^3}{24} + \dots + \frac{z^{p-1}}{p!}} \tag{29}$$

for odd p . Even-order RA methods are A -stable for orders 2 and 4, and $A(\alpha)$ -stable for the higher orders, and odd-order RA methods are not $A(\alpha)$ -stable; see the stability regions of the order 2–7 RA methods in Figure 1.

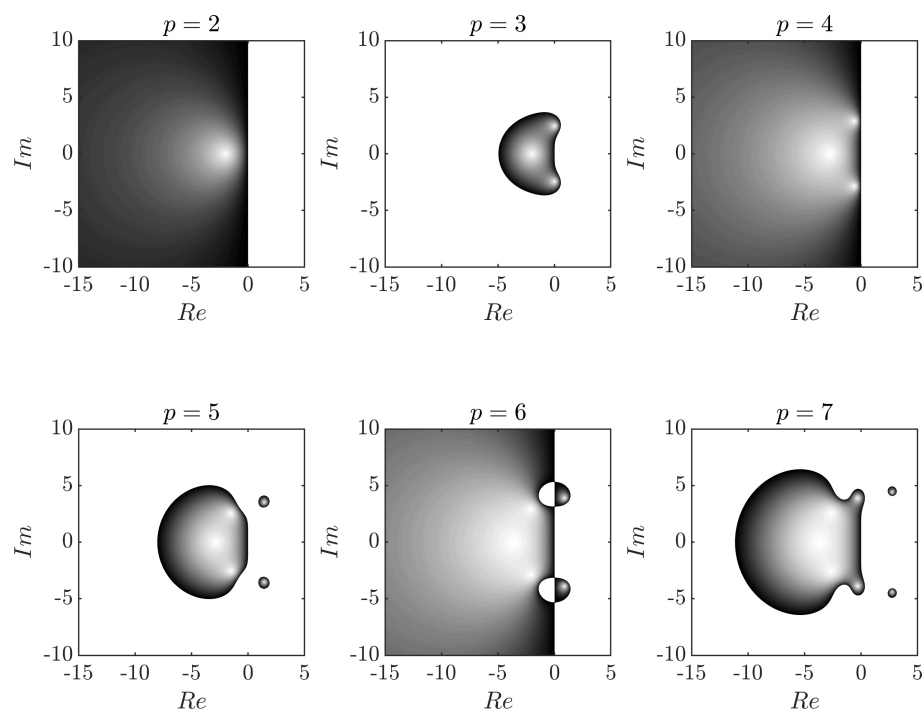


Figure 1. Stability regions of the order 2–7 RA methods.

The essential issue in RA method implementation is using the matrix inverse (18) instead of element-wise operations. Due to Theorem 2, this preserves $A(\alpha)$ -stability.

Theorem 2. For the $A(\alpha)$ -stable rational approximation method,

$$Y_{n+1} = \mathbf{v}(Y_n) \oslash \mathbf{w}(Y_n), \mathbf{v} = PF \circ F, \mathbf{w} = QF, \tag{30}$$

where \mathbf{v} and \mathbf{w} are vectors, P and Q are matrices, and \oslash is a certain binary operator:

1. Treating \oslash as an element-wise division (8) may lead to the loss of $A(\alpha)$ -stability for a certain Y_n , i.e., for a multidimensional linear test problem:

$$Y' = AY, \tag{31}$$

where A is a matrix, $\exists \{Y_n, A, h\} : Y_{n+1} = RY_n, \max |\text{eig}(R)| \geq 1$.

2. Treating \oslash as a matrix inverse-based operation (18) preserves $A(\alpha)$ -stability for any Y_n .

Proof. Consider the n -dimensional test problem in (31). For the method in (30), its matrix form reads:

$$Y_{n+1} = (PY_n \circ Y_n) \oslash (QY_n),$$

and from (28)

$$P = 1 + \frac{hA}{2} + \frac{h^2 A^2}{6} + \dots + \frac{h^{p-1} A^{p-1}}{p!}$$

and

$$Q = 1 - \frac{hA}{2} + \frac{h^2 A^2}{6} - \dots - \frac{h^{p-1} A^{p-1}}{p!}.$$

The stability function can be found depending on the operator \odot :

1. If \odot is an element-wise division, the stability function for each dimension is found independently as

$$R_i(hA, Y_n) = \frac{u_i}{v_i}, u = (PY_n \circ Y_n), v = (QY_n).$$

Let us give one example of how the stability may be violated. Let $Y_n = (1 \ 1 \ 0 \ 0 \dots \ 0)^T$, and A is a matrix of a stable system in the Jordan normal form:

$$A = \begin{pmatrix} \sigma & \omega & & \\ -\omega & \sigma & & \\ & & \ddots & \end{pmatrix},$$

where $\sigma < 0$ and $\omega > 0$ are real numbers. In this example,

$$R_1(hA, Y_n) = \frac{1 + h\frac{\sigma}{2} + h\frac{\omega}{2} + h^2\frac{\sigma^2 - \omega^2}{6} + h^2\frac{2\sigma\omega}{6} + \dots}{1 - \frac{h\sigma}{2} - \frac{h\omega}{2} + h^2\frac{\sigma^2 - \omega^2}{6} + h^2\frac{2\sigma\omega}{6} - \dots}.$$

Let h be small enough to neglect terms by h^2 and its higher powers because

$$R_1(hA, Y_n) = \frac{1 + h\frac{\sigma}{2} + h\frac{\omega}{2} + \mathcal{O}(h^2)}{1 - \frac{h\sigma}{2} - \frac{h\omega}{2} + \mathcal{O}(h^2)},$$

and denote $h\frac{\sigma}{2} = -a, h\frac{\omega}{2} = b$. Then,

$$R_1(hA, Y_n) = \frac{1 - a + b}{1 + a - b}, R_2(hA, Y_n) = \frac{1 - a - b}{1 + a + b}.$$

The stability loss means elongation of the vector Y_{n+1} , i.e., $\|Y_{n+1}\|_2^2 = R_1^2 + R_2^2 > \|Y_n\|_2^2 = 2$. However, for all positive a and b , the condition $\|Y_{n+1}\|_2^2 > 2$ is satisfied. This example explicitly shows that element-wise division may violate $A(\alpha)$ -stability for an arbitrary stable system and initial conditions.

2. If \odot is the matrix inverse, the stability function is

$$R(hA) = Q^{-1}P,$$

which does not depend on Y_n . Therefore, $A(\alpha)$ -stability is preserved in the multidimensional case similar to the other one-step methods [1].

□

4. Adaptive Step Control

Adaptive step control may sufficiently increase the performance of the integration program. In order to control the step size, error estimation is required. RA methods provide three possible error estimation strategies:

1. The error term for the method of order p can be found similarly to the Taylor method:

$$err_{n+1} = h^p \frac{F^{(p-1)}(Y_{n+1})}{p!}, \tag{32}$$

which is practically applicable since $F^{(p-1)}(Y_{n+1})$ is already computed.

2. The error term for the method of order p can be found from the previous order method. Let $p - 1$ be odd, then the $(p - 1)$ -order solution is

$$\tilde{Y}_{n+1} = Y_n + Q_{p-1}^{-1}(Y_n)h\Theta(Y_n),$$

and the p -order solution is

$$Y_{n+1} = Y_n + \left(Q_{p-1} - h^p \frac{F^{(p-1)}(Y_n)}{p!} \right)^{-1} (Y_n)h\Theta(Y_n),$$

since $\Theta(Y_n)$ is similar for orders p and $p - 1$, from where

$$err_{n+1} = Y_{n+1} - \tilde{Y}_{n+1}. \tag{33}$$

3. The error term for the method of order p can be found using the numerator alternation technique. Let the p -order solution be

$$Y_{n+1} = Y_n + Q^{-1}(Y_n)h\Theta(Y_n),$$

then the $p - 1$ -order solution is

$$\tilde{Y}_{n+1} = Y_n + Q^{-1}(Y_n) \left(h\Theta(Y_n) + ch^p F^{(p-1)}(Y_n) \right), \tag{34}$$

where c is a constant. We propose to use $c = \frac{1}{p!}$. The error is then found using the formula (33). The technique (34) is less computationally expensive since the matrix Q is similar for both \tilde{Y}_{n+1} and Y_{n+1} , which allows for performing its LU -decomposition and for speeding up the second matrix inverse.

Furthermore, we used the error estimator based on Equation (34), which provided slightly better results in our preliminary tests.

In order to provide smoother step size control, we use a step control algorithm developed by Söderlind [23]:

$$h_{n+1} = \gamma h_n \left(\frac{\xi \text{tol}}{\|err_{n+1}\|} \right)^{b_1} \left(\frac{\xi \text{tol}}{\|err_n\|} \right)^{b_2} \left(\frac{\xi \text{tol}}{\|err_{n-1}\|} \right)^{b_3} \left(\frac{h_n}{h_{n-1}} \right)^{-a_2} \left(\frac{h_{n-1}}{h_{n-2}} \right)^{-a_3},$$

where a and b are parameter vectors, $\gamma = 0.99, \xi = 0.9$ are safety factors, and tol is the tolerance chosen to satisfy the condition:

$$\text{tol} = \max(\text{reltol} \cdot \|Y_{n+1}\|, \text{abstol}),$$

where reltol is the relative tolerance and abstol is the absolute tolerance.

The best results were achieved using

$$b = \left(\frac{1}{4k} \quad \frac{1}{2k} \quad \frac{1}{4k} \right),$$

$$a = (1 \quad 0 \quad 0),$$

where k is chosen proportional to the method order. In our practical tests, we use $k = 4p$ for better damping.

5. Experimental Results

In this section, the implementation of the fourth-order rational approximation method with its third-order counterpart (34) for determining the error is considered. The resulting method is denoted as RA4(3). The order 2 RA method is identical to the widely known linearly implicit midpoint method, of which the properties have been investigated in [8]; therefore, we omit its analysis. The sixth-order method RA6(5) and the higher-order methods can be efficient only with extremely high tolerance settings, which require extended data types due to the complexity of the higher derivatives in matrix form (see Appendix A), so their properties are to be investigated in the future. Nevertheless, fourth-order methods, including the popular Runge–Kutta 4 method, are widely used in engineering and scientific computations and are among the most common methods [1].

We compare the RA4(3) methods with the Taylor 4(3) method and two Runge–Kutta methods. The first Runge–Kutta method under investigation is the explicit one-step 4(3) method described in [24]. Precisely, it is the classical Runge–Kutta 4 method with an embedded counterpart of order 3. This method is further denoted as ERK4(3). The second Runge–Kutta method is the implicit LobattoIIIC 4(3) method, which is L -stable and B -stable and was considered by some authors to be an excellent method for stiff problems [25]. Meanwhile, it is a fully implicit RK method and, therefore, a computationally expensive one. To make it more efficient in our study, we use the conventional technique of “Jacobian freezing” in the Newton method iterations; see [8].

To obtain a reference solution, we used the Dormand–Prince 8(7) method with high accuracy settings and extended data type [1].

We considered four stiff test problems in this study: the Van der Pol oscillator problem (VDPL), High Irradiance Responses (HIRES) of the photomorphogenesis problem [1], the Riccati four-dimensional equation problem (RICCATI) [26], and a modified FitzHugh–Nagumo–Rinzel neuron model (FNR) from [27]. Our first example shows how different methods deal with the step size and how numerical differentiation affects step size dynamics. The second example studies the performance of the methods.

5.1. Time Domain Analysis and Issues of Step Control

This subsection considers several test problems and how different methods behave with step-size controllers on them. The Van der Pol (VDPL) test problem is given by the following equations:

$$\begin{cases} \dot{y}_1 = y_2, \\ \dot{y}_2 = \mu(1 - y_1^2)y_2 - y_1, \end{cases} \quad (35)$$

where μ is the nonlinearity parameter: the higher its value, the stiffer the problem. In our experiments, we use the value $\mu = 1000$ and experiment time $t_{\max} = 2000$. We used the following settings: $h_{\min} = 10^{-10}$, $h_{\max} = 10$, $abstol = 10^{-11}$, $reltol = 10^{-8}$. Figure 2 depicts the obtained results in the time domain.

The time-domain solution is given on a logarithmic scale of its absolute value. The plot at the bottom shows that the largest steps are selected by the L -stable Lobatto IIIC method. The RA method generally follows it, while the step size strategies of the Taylor and ERK methods are quite similar.

Figure 3 shows the difference in step selection strategies between the method with element-wise division (8) and the method with matrix inverse. This example compromises the element-wise division applied to the RA method. The element-wise division in the Van der Pol oscillator leads to a notable accuracy loss (see upper plot) and reduces the efficiency of the method because it becomes unable to select large step sizes in the same way that a truly implicit method can. The Taylor method stepping strategy is given for comparison.

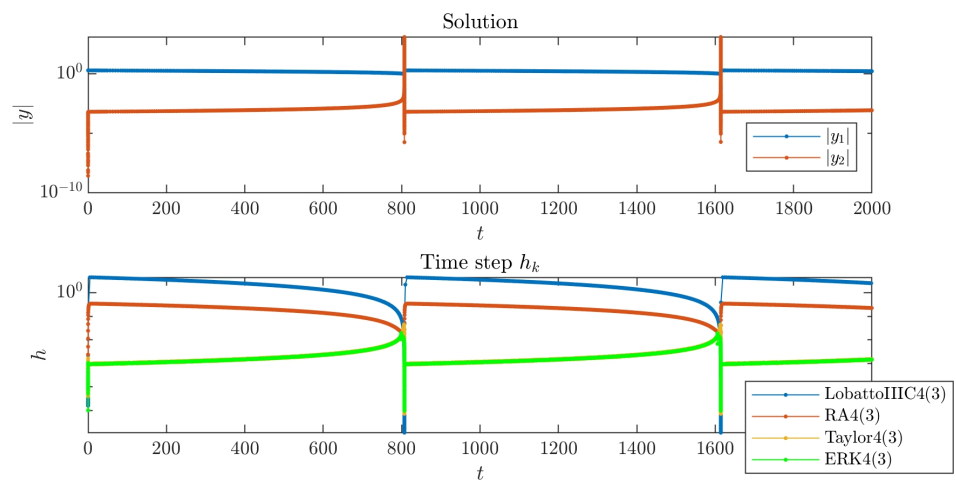


Figure 2. Solution to the Van der Pol system in the time domain (**top**) and step sizes for different solvers (**bottom**).

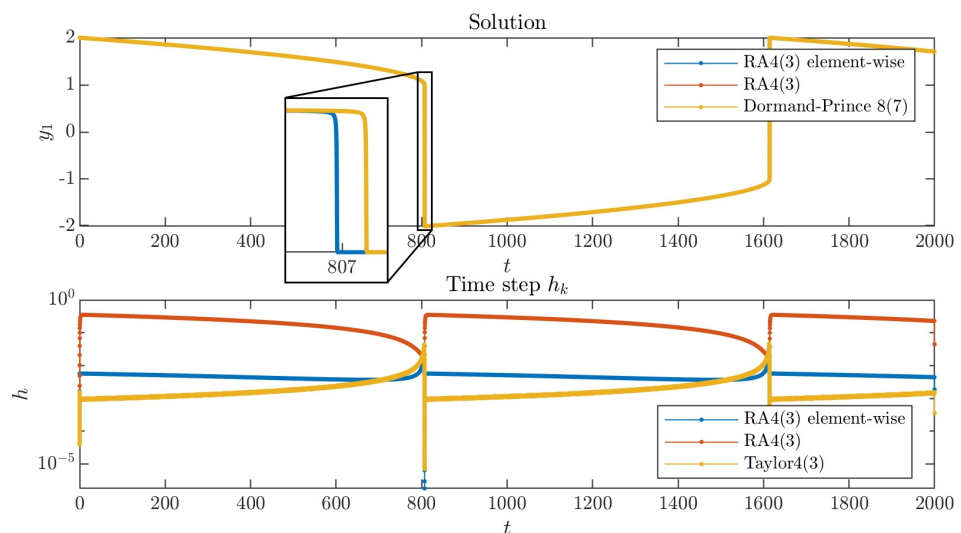


Figure 3. Comparison of element-wise and matrix operation in the rational approximation method. The solution to the Van der Pol system in the time domain (**top**), step sizes for different solvers (**bottom**).

The stiff Riccati equation test problem (RICCATI) is described by the following equations:

$$\begin{cases} \dot{y}_1 = 10,000 - y_1^2 - y_3y_2, \\ \dot{y}_2 = -y_2(y_1 + y_4), \\ \dot{y}_3 = -y_3(y_1 + y_4), \\ \dot{y}_4 = 10,000 - y_4^2 - y_3y_2, \end{cases} \quad (36)$$

with initial value $y(t_0) = (0 \ 0 \ 1 \ 0)^\top$ and simulation time $t_{\max} = 3$. The solver settings used for time-domain test were $h_{\min} = 10^{-10}$, $h_{\max} = 100$, $abstol = 10^{-10}$, $reltol = 10^{-5}$, Figure 4 depicts the obtained results in the time domain.

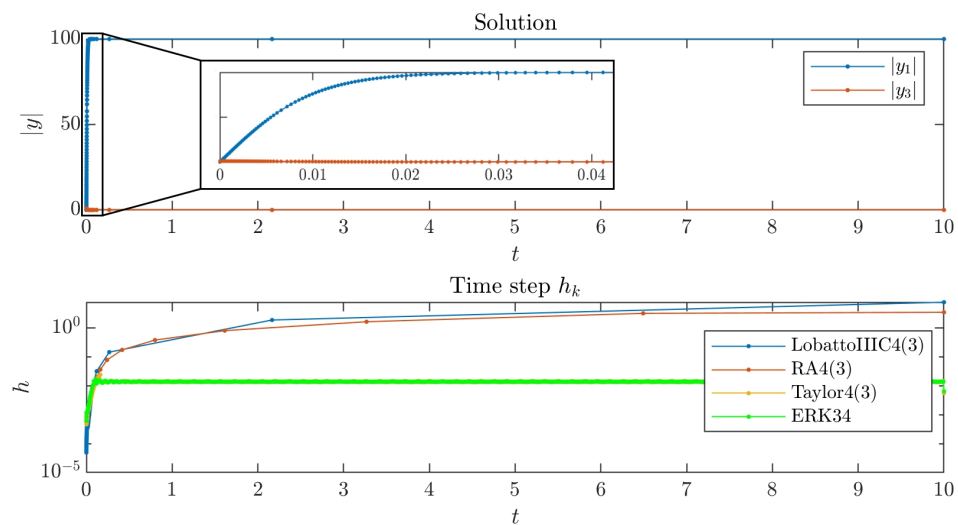


Figure 4. Solution to the Riccati system in the time domain (top) and step sizes for different solvers (bottom).

The High Irradiance Responses test problem (HIRES) is described by the following equations:

$$\begin{cases} \dot{y}_1 = -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007, \\ \dot{y}_2 = 1.71y_1 - 8.75y_2, \\ \dot{y}_3 = -10.03y_3 + 0.43y_4 + 0.035y_5, \\ \dot{y}_4 = 8.32y_2 + 1.71y_3 - 1.12y_4, \\ \dot{y}_5 = -1.745y_5 + 0.43y_6 + 0.43y_7, \\ \dot{y}_6 = -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7, \\ \dot{y}_7 = 280y_6y_8 - 1.81y_7, \\ \dot{y}_8 = -280y_6y_8 + 1.81y_7, \end{cases} \tag{37}$$

with initial value $y(t_0) = (1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.0057)^\top$ and simulation time $t_{\max} = 100$. The solver settings were $h_{\min} = 10^{-10}$, $h_{\max} = 100$, $abstol = 10^{-10}$, $reltol = 10^{-5}$, Figure 5 depicts the obtained results in the time domain.

The sine-excited FitzHugh–Nagumo–Rinzel neuron test problem (FHN) is described by the following equations:

$$\begin{cases} \dot{y}_1 = y_2 - y_1^3/3 - y_2 - y_3 + y_5, \\ \dot{y}_2 = a_1(a_2 + y_1 - a_3y_2), \\ \dot{y}_3 = a_4(a_5 - y_3 - y_1), \\ \dot{y}_4 = Ay_5, \\ \dot{y}_5 = -\omega^2y_4, \end{cases} \tag{38}$$

where the parameter vector is $a = (0.5 \ 0.7 \ 0.8 \ 0.002 \ -1)^\top$, the sine perturbation function amplitude is $A = 0.083$, the radial frequency is $\omega = 0.00003 \cdot 2\pi$, the initial value is $y(t_0) = (0 \ 0 \ 0 \ 0 \ 1)^\top$, and the simulation time is $t_{\max} = 70,000$.

The solver settings were $h_{\min} = 10^{-10}$, $h_{\max} = 100$, $abstol = 10^{-10}$, $reltol = 10^{-8}$, Figure 6 shows the behavior of the obtained solution.

In all examples, the LobattoIIC method can work with the largest step sizes, while the Taylor and ERK solvers select small step sizes, restricted by stability requirements.

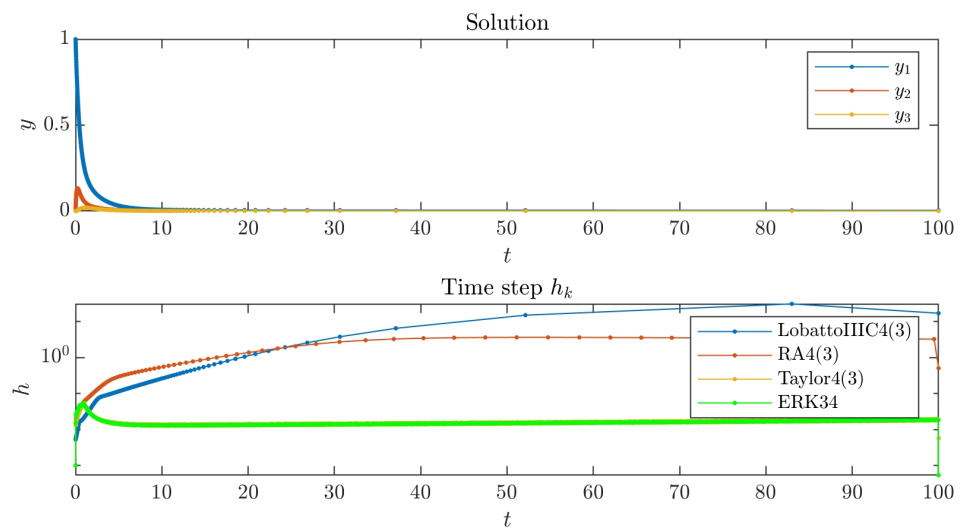


Figure 5. Solution to the HIRES system in the time domain (top) and step sizes for different solvers (bottom).

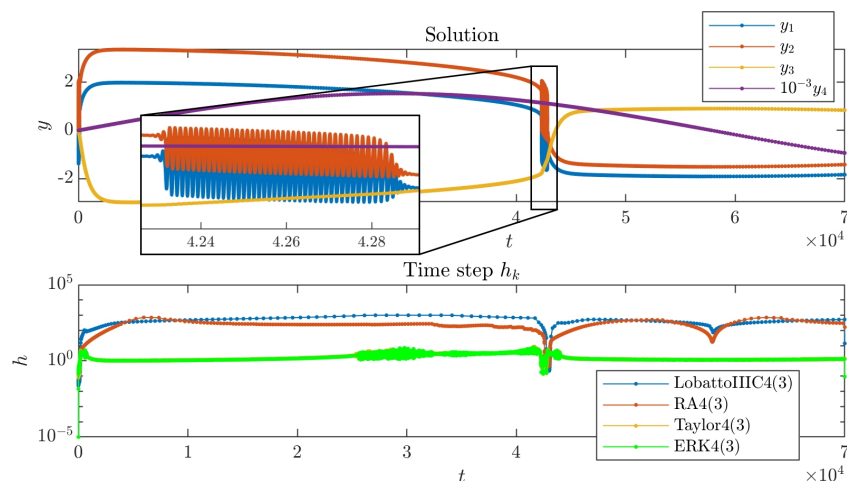


Figure 6. Solution to the FitzHugh–Nagumo–Rinzel system in the time domain (top) and step sizes for different solvers (bottom).

5.2. Performance Analysis

In this subsection, we study the performance of the methods on four described problems. The results are given in Figure 7. Four considered test problems highlighted the higher performance of the proposed RA method. Explicit methods are poorly suited for this set of problems, as seen from the plots: their computational time is almost independent of accuracy. Thus, their step controllers restrict step sizes on all investigated tolerances to maintain the stability.

All of the experiments were carried out in a MATLAB R2018b environment, on a computer with the 64-bit Win10 operational system, Intel Core i5-7300HQ 2.5 GHz CPU, and 16 GB RAM. The relative tolerance value was altered within the interval $[10^{-9}, 10^{-1}]$, and the absolute tolerance was chosen as $abstol = 10^{-5} \cdot reltol$. The double data type was used.

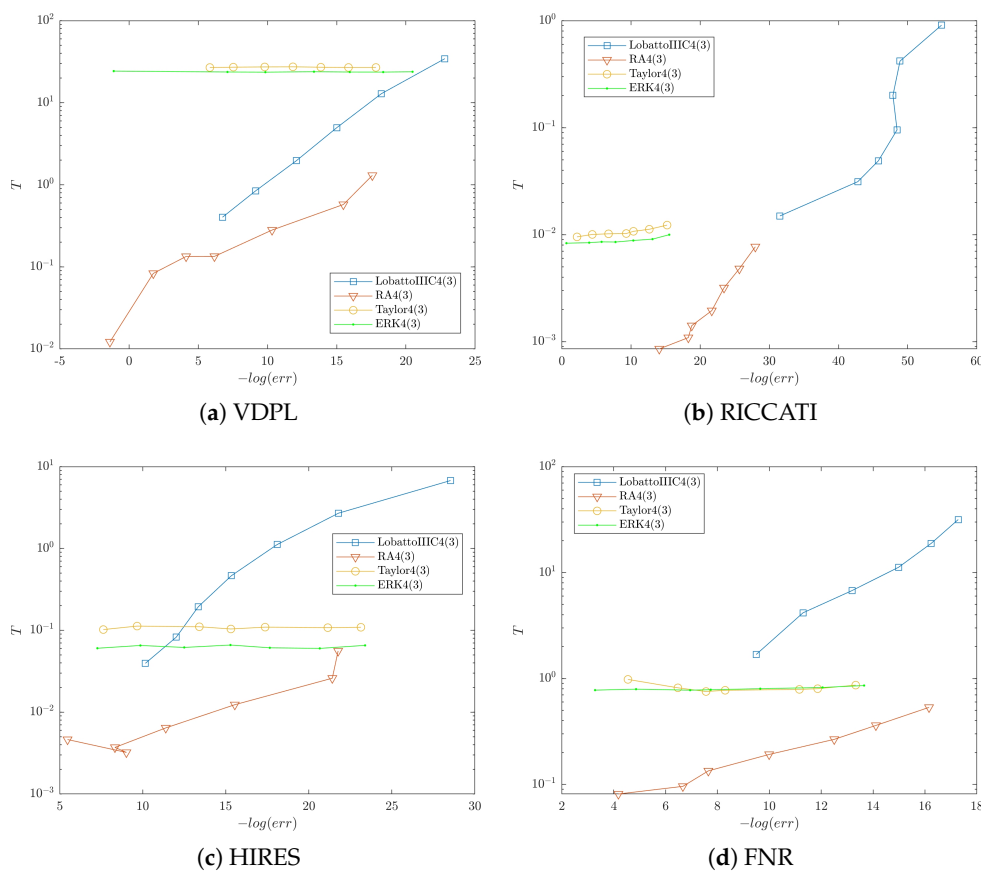


Figure 7. Performance plots of the 4(3) methods for the stiff test problems: LobattoIIC method, rational approximation method with analytical derivative (RA), Taylor series method, and explicit Runge–Kutta (ERK) method. The horizontal axis represents the negative logarithm of the achieved error in the last point; the vertical axis represents the computation time as a median value of 7 successive runs.

6. Conclusions and Discussion

In this paper, we proposed a prospective type of rational approximation method for solving stiff ODEs. This method has a strong relation to the Taylor expansion and is easy to derive and implement. The A - and $A(\alpha)$ -stability of this method is proven, and most notably, the method is A -stable in the case of orders 2 and 4. Compared with the Taylor method, and explicit and implicit Runge–Kutta methods, it has the best performance when solving stiff test problems, as our experiments with 4(3) order methods on double data type revealed.

Similar to the Taylor series methods, the proposed method requires algebraic differentiation to achieve the best performance. It is of great interest to test the performance of the proposed method on extended data types and high orders, which will be the topic of our future work.

The question arises of whether we found the “best” rational approximation for numerical integration or whether there exist other types of approximation that can achieve better performance. It is also of interest to derive a Runge–Kutta-like formula for this method, especially for the matrix denominator, which may considerably increase the method performance.

The proposed numerical method is also prospective for solving problems with singularities, especially poles in the denominator of the derivative function.

RA methods are potentially suitable for solving DAEs, which will be a subject of our future investigation.

Author Contributions: Conceptualization, A.K. and D.B.; data curation, E.G.N.; formal analysis, E.G.N.; investigation, A.K., D.B. and V.A.; methodology, D.B.; project administration, D.B.; resources, A.K. and V.A.; software, A.K. and V.A.; validation, A.K., D.B. and E.G.N.; visualization, A.K.; writing—original draft, A.K. and D.B.; writing—review and editing, V.A. and E.G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Appendix A. Higher-Order Derivatives in Matrix Form

Here, we present derivatives of orders 4 and 5, needed for RA and Taylor series methods of orders 5 and 6:

$$\begin{aligned}
 F^{(4)} = & \frac{\partial^3 J}{\partial Y^3} FFFF + \frac{\partial^2 J}{\partial Y^2} JFFF + 2 \frac{\partial^2 J}{\partial Y^2} FJFF + 3 \frac{\partial^2 J}{\partial Y^2} FFJF + \frac{\partial J}{\partial Y} \frac{\partial J}{\partial Y} FFF + \dots \\
 & + \frac{\partial J}{\partial Y} J J F F + 3 \frac{\partial J}{\partial Y} J F J F + 3 \frac{\partial J}{\partial Y} F \frac{\partial J}{\partial Y} F F + 3 \frac{\partial J}{\partial Y} F J J F + J \frac{\partial^2 J}{\partial Y^2} F F F + \dots \\
 & + J \frac{\partial J}{\partial Y} J F F + 2 J \frac{\partial J}{\partial Y} F J F + J J \frac{\partial J}{\partial Y} F F + J J J J F,
 \end{aligned}$$

$$\begin{aligned}
 F^{(5)} = & \frac{\partial^4 J}{\partial Y^4} FFFFF + \frac{\partial^3 J}{\partial Y^3} JFFFF + 2 \frac{\partial^3 J}{\partial Y^3} FJFFF + 3 \frac{\partial^3 J}{\partial Y^3} FFJFF + 4 \frac{\partial^3 J}{\partial Y^3} FFFJF + \dots \\
 & \frac{\partial^2 J}{\partial Y^2} \frac{\partial J}{\partial Y} FFFF + \frac{\partial^2 J}{\partial Y^2} J J F F F + 3 \frac{\partial^2 J}{\partial Y^2} J F J F F + 4 \frac{\partial^2 J}{\partial Y^2} J F F J F + 3 \frac{\partial^2 J}{\partial Y^2} F \frac{\partial J}{\partial Y} F F F + \dots \\
 & + 3 \frac{\partial^2 J}{\partial Y^2} F J J F F + 8 \frac{\partial^2 J}{\partial Y^2} F J F J F + 6 \frac{\partial^2 J}{\partial Y^2} F F \frac{\partial J}{\partial Y} F F + 6 \frac{\partial^2 J}{\partial Y^2} F F J J F + \frac{\partial J}{\partial Y} \frac{\partial^2 J}{\partial Y^2} F F F F + \dots \\
 & + \frac{\partial J}{\partial Y} \frac{\partial J}{\partial Y} J F F F + 2 \frac{\partial J}{\partial Y} \frac{\partial J}{\partial Y} F J F F + 4 \frac{\partial J}{\partial Y} \frac{\partial J}{\partial Y} F F J F + \frac{\partial J}{\partial Y} J \frac{\partial J}{\partial Y} F F F + \frac{\partial J}{\partial Y} J J J F F + \dots \\
 & + 4 \frac{\partial J}{\partial Y} J J F J F + 6 \frac{\partial J}{\partial Y} J F \frac{\partial J}{\partial Y} F F + 6 \frac{\partial J}{\partial Y} J F J J F + 4 \frac{\partial J}{\partial Y} F \frac{\partial^2 J}{\partial Y^2} F F F + 4 \frac{\partial J}{\partial Y} F \frac{\partial J}{\partial Y} J F F + \dots \\
 & + 8 \frac{\partial J}{\partial Y} F \frac{\partial J}{\partial Y} F J F + 4 \frac{\partial J}{\partial Y} F J \frac{\partial J}{\partial Y} F F + 4 \frac{\partial J}{\partial Y} F J J J F + J \frac{\partial^3 J}{\partial Y^3} F F F F + J \frac{\partial^2 J}{\partial Y^2} J F F F + \dots \\
 & + 2 J \frac{\partial^2 J}{\partial Y^2} F J F F + 3 J \frac{\partial^2 J}{\partial Y^2} F F J F + J \frac{\partial J}{\partial Y} \frac{\partial J}{\partial Y} F F F + J \frac{\partial J}{\partial Y} J J F F + 3 J \frac{\partial J}{\partial Y} J F J F + \dots \\
 & 3 J \frac{\partial J}{\partial Y} F \frac{\partial J}{\partial Y} F F + 3 J \frac{\partial J}{\partial Y} F J J F + J J \frac{\partial^2 J}{\partial Y^2} F F F + J J \frac{\partial J}{\partial Y} J F F + 2 J J \frac{\partial J}{\partial Y} F J F + \dots \\
 & + J J J \frac{\partial J}{\partial Y} F F + J J J J F.
 \end{aligned}$$

Due to their complexity, we do not recommend implementing methods utilizing these formulas using double data type. Instead, the extended data type should be used.

References

- Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*; Springer: Berlin/Heidelberg, Germany, 1987.
- Corliss, G.; Chang, Y. Solving ordinary differential equations using Taylor series. *ACM Trans. Math. Softw. (TOMS)* **1982**, *8*, 114–144. [[CrossRef](#)]
- Abad, A.; Barrio, R.; Blesa, F.; Rodríguez, M. Algorithm 924: TIDES, a Taylor series integrator for differential equations. *ACM Trans. Math. Softw. (TOMS)* **2012**, *39*, 1–28. [[CrossRef](#)]
- Miletics, E.; Molnárka, G. Taylor Series Method with Numerical Derivatives for numerical solution of ODE initial values problems. *Hung. Electron. J. Sci.* **2003**, 1–16.
- Berry, D.W.; Childs, A.M.; Cleve, R.; Kothari, R.; Somma, R.D. Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.* **2015**, *114*, 090502. [[CrossRef](#)] [[PubMed](#)]

6. Barrio, R. Sensitivity analysis of ODEs/DAEs using the Taylor series method. *SIAM J. Sci. Comput.* **2006**, *27*, 1929–1947. [[CrossRef](#)]
7. Brugnano, L.; Mazzia, F.; Trigiante, D. Fifty Years of Stiffness. In *Recent Advances in Computational and Applied Mathematics*; Springer: Dordrecht, The Netherlands, 2009. [[CrossRef](#)]
8. Hairer, E.; Wanner, G. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*; Springer: Berlin/Heidelberg, Germany, 1996.
9. Barrio, R. Performance of the Taylor series method for ODEs/DAEs. *Appl. Math. Comput.* **2005**, *163*, 525–545. [[CrossRef](#)]
10. Koskela, A.; Ostermann, A. Exponential Taylor methods: Analysis and implementation. *Comput. Math. Appl.* **2013**, *65*, 487–499. [[CrossRef](#)]
11. El-Zahar, E.R.; Tenreiro Machado, J.; Ebaid, A. A New Generalized Taylor-Like Explicit Method for Stiff Ordinary Differential Equations. *Mathematics* **2019**, *7*, 1154. [[CrossRef](#)]
12. ul Islam, S.; Aziz, I.; Haq, F. A comparative study of numerical integration based on Haar wavelets and hybrid functions. *Comput. Math. Appl.* **2010**, *59*, 2026–2036. [[CrossRef](#)]
13. Sabermahani, S.; Ordokhani, Y.; Yousefi, S.A. Fractional-order general Lagrange scaling functions and their applications. *BIT Numer. Math.* **2020**, *60*, 101–128. [[CrossRef](#)]
14. Wuytack, L. Numerical integration by using nonlinear techniques. *J. Comput. Appl. Math.* **1975**, *1*, 267–272. [[CrossRef](#)]
15. Alaybeyi, M.M. On the Relation between Numerical Integration and Padé Approximation. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
16. Dahlquist G.; Björck, Å. *Numerical Methods in Scientific Computing, Volume I*; SIAM: Philadelphia, PA, USA, 2008.
17. Burtsev, Y. High Precision Methods Based on Pade Approximation of Matrix Exponent for Numerical Analysis of Stiff-Oscillatory Electrical Circuits. In Proceedings of the 2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russia, 18–22 May 2020; pp. 1–7.
18. Gadella, M.; Lara, L. A Numerical method for solving ODE by rational approximation. *Appl. Math. Sci.* **2013**, *7*, 1119–1130. [[CrossRef](#)]
19. Graves-Morris, P.; Hopkins, T. Reliable rational interpolation. *Numer. Math.* **1980**, *36*, 111–128. [[CrossRef](#)]
20. Ramos, H.; Singh, G.; Kanwar, V.; Bhatia, S. An embedded 3(2) pair of nonlinear methods for solving first order initial-value ordinary differential systems. *Numer. Algorithms* **2017**, *75*, 509–529. [[CrossRef](#)]
21. Butusov, D.; Karimov, A.; Tutueva, A.; Kaplun, D.; Nepomuceno, E.G. The effects of Padé numerical integration in simulation of conservative chaotic systems. *Entropy* **2019**, *21*, 362. [[CrossRef](#)] [[PubMed](#)]
22. Wen, B.J.; Yun, N.; Wang, F.Z. An A-stable Explicit Numerical Integration Method Based on Padé Approximation. In Proceedings of the 2016 International Conference on Electrical Engineering and Automation (ICEEA 2016), Xiamen, China, 18–19 December 2016.
23. Söderlind, G. Digital filters in adaptive time-stepping. *ACM Trans. Math. Softw. (TOMS)* **2003**, *29*, 1–26. [[CrossRef](#)]
24. Balac, S.; Mahé, F. Embedded Runge–Kutta scheme for step-size control in the interaction picture method. *Comput. Phys. Commun.* **2013**, *184*, 1211–1219. [[CrossRef](#)]
25. Jay, L.O. Lobatto methods. *Chemistry* **1996**, *29*, 298–305.
26. Butusov, D.; Karimov, T.; Ostrovskii, V. Semi-implicit ODE solver for matrix Riccati equation. In Proceedings of the 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EConRusNW), Saint Petersburg, Russia, 2–3 February 2016; pp. 168–172.
27. Wojcik, J.; Shilnikov, A. Voltage interval mappings for an elliptic bursting model. In *Nonlinear Dynamics New Directions*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 195–213.