

Semi-Implicit Multistep Extrapolation ODE Solvers

Denis Butusov ^{1,*} , Aleksandra Tutueva ² , Petr Fedoseev ², Artem Terentev ² and Artur Karimov ¹ 

¹ Youth Research Institute, Saint Petersburg Electrotechnical University “LETI”,
197376 Saint Petersburg, Russia; aikarimov@etu.ru

² Department of Computer-Aided Design, Saint Petersburg Electrotechnical University “LETI”,
197376 Saint Petersburg, Russia; avtutueva@etu.ru (A.T.); psfedoseev@etu.ru (P.F.);
arter2011@gmail.com (A.T.)

* Correspondence: dnbutusov@etu.ru; Tel.: +7-950-008-7190

Received: 10 May 2020; Accepted: 2 June 2020; Published: 8 June 2020



Abstract: Multistep methods for the numerical solution of ordinary differential equations are an important class of applied mathematical techniques. This paper is motivated by recently reported advances in semi-implicit numerical integration methods, multistep and extrapolation solvers. Here we propose a novel type of multistep extrapolation method for solving ODEs based on the semi-implicit basic method of order 2. Considering several chaotic systems and van der Pol nonlinear oscillator as examples, we implemented a performance analysis of the proposed technique in comparison with well-known multistep methods: Adams–Bashforth, Adams–Moulton and the backward differentiation formula. We explicitly show that the multistep semi-implicit methods can outperform the classical linear multistep methods, providing more precision in the solutions for nonlinear differential equations. The analysis of stability regions reveals that the proposed methods are more stable than explicit linear multistep methods. The possible applications of the developed ODE solver are the long-term simulations of chaotic systems and processes, solving moderately stiff differential equations and advanced modeling systems.

Keywords: numerical integration; multistep methods; semi-implicit methods; chaotic systems; extrapolation; ODE solver

1. Introduction

In recent years, tools for computer simulation have become increasingly important in various fields of science. Computational experiments allow one to obtain new knowledge about the properties and dynamics of the object under investigation without the significant economic and time costs and risks that attend a full-scale experiment. To study the continuous dynamical systems described by ordinary differential equations (ODE), numerical integration methods are traditionally used. While the scale and complexity of the systems under investigation increase, one needs more sophisticated and efficient integration techniques [1]. Therefore, the advanced numerical methods used to simulate dynamical systems should possess great stability, precision and low computational costs. In engineering applications, especially while solving high-dimensional initial value problems (IVP), multistep integration is widespread [2].

Being compared to their single-step counterparts, multistep methods employ one call to the right-hand-side function per each integration step. To obtain the numerical solution with the desired accuracy, such algorithms reuse values calculated at previous steps. Therefore, multistep algorithms significantly exceed single-step integration in computational speed. However, the well-known problem of such integration is a decrease in numerical stability with an increase in the accuracy order of the chosen scheme. This property restricts the application of high-order multistep methods for solving

stiff ODE systems excluding implicit methods, such as backward differentiation formulas (BDF) [3]. However, the application of implicit methods involves computationally expensive approximation of the numerical solution at the next point using Newton's method or some other iterative technique. This fact eliminates one of the main advantages of multistep algorithms—their computational simplicity and high efficiency.

Several approaches that circumvent the so-called Dahlquist order barrier, including the second derivative multistep method [4], hybrid linear multistep methods [5–7], and extended backward differentiation formulas [8], were recently proposed. Being applied to the stiff ODEs, such methods have shown high numerical efficiency.

Recent studies also revealed the possibility to construct efficient implicit–explicit linear multistep methods for stiff problems [5,6,9]. One of the prospective techniques combining the advantages of implicit and explicit integration for the numerical solution of differential equations is semi-implicit integration. High-performance compositional [10] ODE solvers based on semi-implicit single-step calculations have been proposed. Efficient techniques for non-Hamiltonian systems [11] simulation were developed based on Aitken–Neville extrapolation and the semi-implicit symmetric basic method [12]. It has been shown that, compared with other integration algorithms, semi-implicit methods demonstrate good stability and low computational costs, which allows outperforming both explicit and implicit techniques [13]. Thus, semi-implicit integration no longer remains a specialized tool for Hamiltonian systems and can be efficiently applied to the general class of initial value problems (IVP).

One of the most efficient single-step numerical integration techniques is extrapolation solvers [4]. The well-known *ODEx* solver, based on the Gragg–Bulirsch–Stoer method, is one of the most efficient explicit integration methods [14]. Being implemented with adaptive stepsize it can be even more numerically efficient than embedded Runge–Kutta methods [15]. However, although the first studies of extrapolation multistep schemes have been known since the 1970s [12], there are no multistep extrapolation solvers presented in the modern simulation software. Thus, it is of interest to develop semi-implicit multistep methods based on numerical extrapolation techniques. In this paper, we suggest closing this gap by introducing a novel multistep ODE solver combining the advantages of the extrapolation technique and the semi-implicit symmetric basic method.

The rest of the paper is organized as follows. In Section 2, a novel type of multistep integration methods is described and tables of coefficients are given for solvers of order 3–6 in “full” and “short” form. Section 3 represents the set of computational experiments. We introduce several nonlinear test systems and estimate the computational efficiency of the compared methods through performance plots. In Section 4, we investigate the stability and convergence of ESIMM methods. Finally, in Section 5, some conclusions and discussion are given.

2. Materials and Methods

Consider a basic symmetric integration method for solving IVP of order p . A sequence of local error terms after stepping from the point x_n to x_{n+1} with a stepsize h reads

$$x(t_{n+1}) = x_{n+1} + h^{p+1}e_{p+1} + h^{p+2}e_{p+2} + \dots \quad (1)$$

A sequence of local error terms after stepping from the previous point x_{n-1} to x_{n+1} with a stepsize $2h$ reads

$$x(t_{n+1}) = x_{n+1} + 2^{p+1}h^{p+1}e_{p+1} + 2^{p+2}h^{p+2}e_{p+2} + \dots \quad (2)$$

Following the idea of extrapolation methods [4,12], let us combine the solution at the point $n + 1$ with the stepsize h (denote it as T_1) and the solution at the same point with the stepsize $2h$ (denote it as T_2) as follows:

$$x_{n+1} = k_1 T_1 + k_2 T_2 \quad (3)$$

The dataflow representation of this combination is depicted in Figure 1. Keeping in mind Aitken-Neville interpolation rule, let us select the coefficients k_1 and k_2 so as to satisfy the following conditions:

$$\begin{cases} k_1 + k_2 = 1 \\ k_1 + 2^{p+1}k_2 = 0 \end{cases} \quad (4)$$

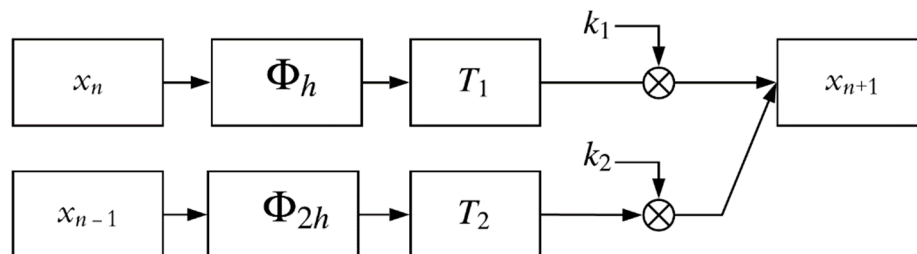


Figure 1. Graphical interpretation of the proposed multistep algorithm: solution T_1 obtained with stepsize h from the point x_n and solution T_2 obtained with stepsize $2h$ from the point x_{n-1} are combined to obtain a higher-order solution. Basic integration methods are denoted as Φ_h and Φ_{2h} with respect to the stepsize.

The system of Equation (4) can be analytically solved as

$$\begin{cases} k_1 = \frac{2^{p+1}}{2^{p+1}-1} \\ k_2 = \frac{-1}{2^{p+1}-1} \end{cases}$$

For example, for the method of order 3 with $p = 2$ we have:

$$k_1 = \frac{8}{7}, k_2 = \frac{-1}{7}$$

This allows eliminating the term by the error e_{p+1} and increasing the order by one.

Let us organize the computational process in a cascade manner as shown in Figure 2. The main term T_{ij} represents a component in the extrapolation table on i -th stage on the j -th line. Indices by coefficients k_{ij}^m are the following: m denotes whether it is the coefficient by the first or the second term, with the sum giving the term T_{ij} .

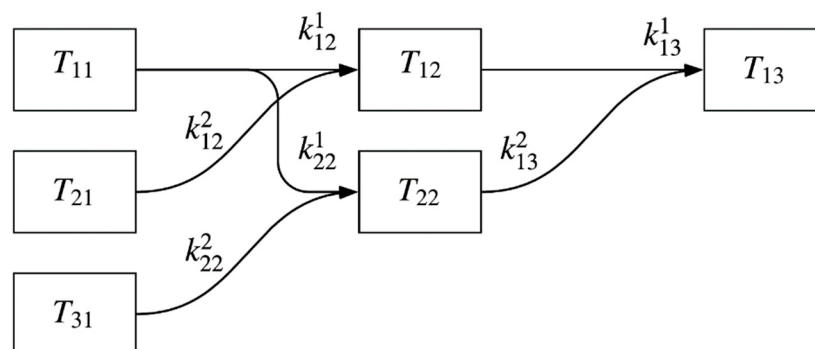


Figure 2. High-order extrapolation table.

A general form for increasing an order can be obtained if we consider how the operation (3) affects the error terms in (1) and (2). Introducing the matrix formalism, for the coefficients on the first stage one can obtain

$$k_{12} = \begin{pmatrix} k_{12}^1 \\ k_{12}^2 \end{pmatrix}; \begin{pmatrix} 1 & 1 \\ 1 & 2^{p+1} \end{pmatrix} \begin{pmatrix} k_{12}^1 \\ k_{12}^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (5)$$

After the first stage of extrapolation, equations for two approximations of $x(t_{n+1})$ denoted as T_{12} and T_{22} , which are then multiplied by k_{13}^1 and k_{13}^2 respectively, are as follows:

$$T_{12} = x_{n+1} + (k_{12}^1 h^{p+2} + k_{12}^2 (2h)^{p+2}) e_{p+2} + \dots$$

$$T_{22} = x_{n+1} + (k_{22}^1 h^{p+2} + k_{22}^2 (3h)^{p+2}) e_{p+2} + \dots$$

Let us rewrite them using a vector form:

$$T_{12} = x_{n+1} + \begin{pmatrix} k_{12}^1 & k_{12}^2 \end{pmatrix} \begin{pmatrix} 1 \\ 2^{p+2} \end{pmatrix} h^{p+2} e_{p+2} + \dots$$

$$T_{22} = x_{n+1} + \begin{pmatrix} k_{22}^1 & k_{22}^2 \end{pmatrix} \begin{pmatrix} 1 \\ 3^{p+2} \end{pmatrix} h^{p+2} e_{p+2} + \dots$$

Similar to the Equation (5), the final equation for k_{13} reads:

$$k_{13} = \begin{pmatrix} k_{13}^1 \\ k_{13}^2 \end{pmatrix}; \left(k_{12}^T \begin{pmatrix} 1 \\ 2^{p+2} \end{pmatrix} \quad k_{13}^T \begin{pmatrix} 1 \\ 3^{p+2} \end{pmatrix} \right) \begin{pmatrix} k_{13}^1 \\ k_{13}^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (6)$$

Continuing such a sequence of operations, obtain an equation for k_{14} :

$$\left(k_{13} \begin{pmatrix} k_{12}^T \begin{pmatrix} 1 \\ 2^{p+3} \end{pmatrix} \\ k_{13}^T \begin{pmatrix} 1 \\ 3^{p+3} \end{pmatrix} \end{pmatrix} \quad k_{23} \begin{pmatrix} k_{12}^T \begin{pmatrix} 1 \\ 2^{p+3} \end{pmatrix} \\ k_{14}^T \begin{pmatrix} 1 \\ 4^{p+3} \end{pmatrix} \end{pmatrix} \right) \begin{pmatrix} k_{14}^1 \\ k_{14}^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (7)$$

Continuing this process, we can derive the coefficients for the method of an arbitrary accuracy order. For the basic symmetric method of order 2, the coefficients for the 6-order multistep extrapolation method are given in Table 1.

Table 1. Coefficients of the 6th order multistep extrapolation method with 2nd order basic method.

Order	Coefficient	Value
3	k_{12}	$(8/7; -1/7)^T$
	k_{22}	$(27/26; -1/26)^T$
	k_{32}	$(64/63; -1/63)^T$
	k_{42}	$(125/124; -1/124)^T$
4	k_{13}	$(189/85; -104/85)^T$
	k_{23}	$(8/5; -3/5)^T$
	k_{33}	$(875/627; -248/627)^T$
5	k_{14}	$(272/83; -189/83)^T$
	k_{24}	$(10625/4982; -5643/4982)^T$
6	k_{15}	$(51875/12019; -39856/12019)^T$

Let us call the method illustrated by Figure 2 and Table 1, a “full” ESIMM method. However, one can use a simplified (“short”) version of the same method:

$$x_{n+1} = x_n + \sum_{i=1}^s k_i T_{i1}.$$

In order to obtain coefficients k_i for the “short” version of the method, consider the Taylor expansion for $k_i x(t_{n+1})$ through T_{i1} up to the desired order of the error term $s + p$:

$$\begin{cases} k_1 x(t_{n+1}) = k_1 T_{11} + k_1 h^{p+1} e_{p+1} + k_1 h^{p+2} e_{p+2} + \dots + k_1 h^{p+s} e_{p+s}, \\ k_2 x(t_{n+1}) = k_2 T_{21} + k_2 (2h)^{p+1} e_{p+1} + k_1 (2h)^{p+2} e_{p+2} + \dots + k_1 (2h)^{p+s} e_{p+s}, \\ k_s x(t_{n+1}) = k_s T_{s1} + k_s (sh)^{p+1} e_{p+1} + k_s (sh)^{p+2} e_{p+2} + \dots + k_s (sh)^{p+s} e_{p+s} \end{cases}$$

One can see, that the sum of k_i equals 1, while the sum of any other terms by any power of h equals zero, which can be written as a system of linear algebraic equations:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2^{p+1} & 3^{p+1} & \dots & s^{p+1} \\ 1 & 2^{p+2} & 3^{p+2} & \dots & s^{p+2} \\ & & \vdots & & \\ 1 & 2^{p+s} & 3^{p+s} & \dots & s^{p+s} \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ \vdots \\ k_s \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

from where k_i can be easily found. Table 2 represents the coefficients for the “short” method version.

Table 2. Coefficients for “short” ESIMM methods of order 3–6

Order 2	1	-	-	-	-
Order 3	$\frac{8}{7}$	$-\frac{1}{7}$	-	-	-
Order 4	$\frac{108}{85}$	$-\frac{27}{85}$	$\frac{4}{85}$	-	-
Order 5	$\frac{576}{415}$	$-\frac{216}{415}$	$\frac{64}{415}$	$-\frac{9}{415}$	-
Order 6	$\frac{18000}{12019}$	$-\frac{9000}{12019}$	$\frac{4000}{12019}$	$-\frac{1125}{12019}$	$\frac{144}{12019}$

One can obtain coefficients for methods of higher order following the general logic of this approach. It should be noted that this technique is efficient only when a self-adjoint symmetric basic method is chosen. In our study, we used the semi-implicit CD [11,12] method of order 2, but the implicit midpoint method can be used as a basic method as well.

3. Results

In this section, we investigate the performance of two versions of the extrapolation semi-implicit multistep (ESIMM) method. Let us recall, that the first version is labeled as “full” and obtains a solution using a direct calculation scheme with the full set of coefficients per each solution step. The second version, named “short”, uses pre-calculated coefficients values for each stage and sums the weighted results, which results in an increased calculation speed at the cost of a possibly insignificant decrease in accuracy compared to the “full” version due to the round-off error. All multistep methods under investigation were used with the Dormand–Prince 8 (DOPRI8) method as a starting solver and reference solution. All the experiments were performed in an NI LabVIEW 2018 environment with double floating-point precision.

We have chosen several well-known nonlinear dynamical systems as a test set for the numerical method investigation.

3.1. Rössler Attractor

The system was proposed by Otto E. Rossler [16], and can be described as follows:

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases} \quad (8)$$

where a , b and c are system parameters. The semi-implicit basic method of order 2 for the system (8) can be formulated as [11–13]:

$$\begin{aligned}y^{n+0.5} &= y^n + \frac{h}{2} * (x^n + a * y^n) \\z^{n+0.5} &= z^n + \frac{h}{2} * (b + z^n * x^n - c * z^n) \\x^{n+1} &= x^n + h(-y^{n+0.5} - z^{n+0.5}) \\z^{n+1} &= \frac{z^{n+0.5} + \frac{h}{2} * b}{1 - \frac{h}{2} * x^{n+1} + c * \frac{h}{2}} \\y^{n+1} &= \frac{y^{n+0.5} + \frac{h}{2} * x^{n+1}}{1 - a * \frac{h}{2}}\end{aligned}$$

We carried out experiments with the following parameter values: $a = 0.2$, $b = 0.2$, $c = 5.7$. Other simulation parameters, including simulation time, stepsize values, initial conditions and number of measurements are given in Table 3.

Table 3. Parameters used with the experiments with the Rossler system.

Order of Accuracy	3	4	5	6
Step values, sec.	0.02		0.02	0.008
	0.015		0.015	0.0075
	0.01		0.01	0.006
	0.008		0.008	0.005
	0.006		0.006	0.004
	0.003		0.003	0.003
	0.002		0.002	0.0025
	0.001		0.0015	0.002
Initial conditions	(1; 1; 1)			
Simulation time, sec.	40			
Number of code runs	50			

One can see from Figure 3 that the proposed semi-implicit extrapolation multistep method (ESIMM) possesses better performance than well-known linear multistep methods. The “short” version of ESIMM provides a great balance between computational cost and accuracy, while the “full” version is slightly more precise but requires more time to perform calculations. Classical explicit multistep methods such as Adams–Bashforth tend to have closer performance in higher-order schemes but are still significantly less accurate compared to the proposed method.

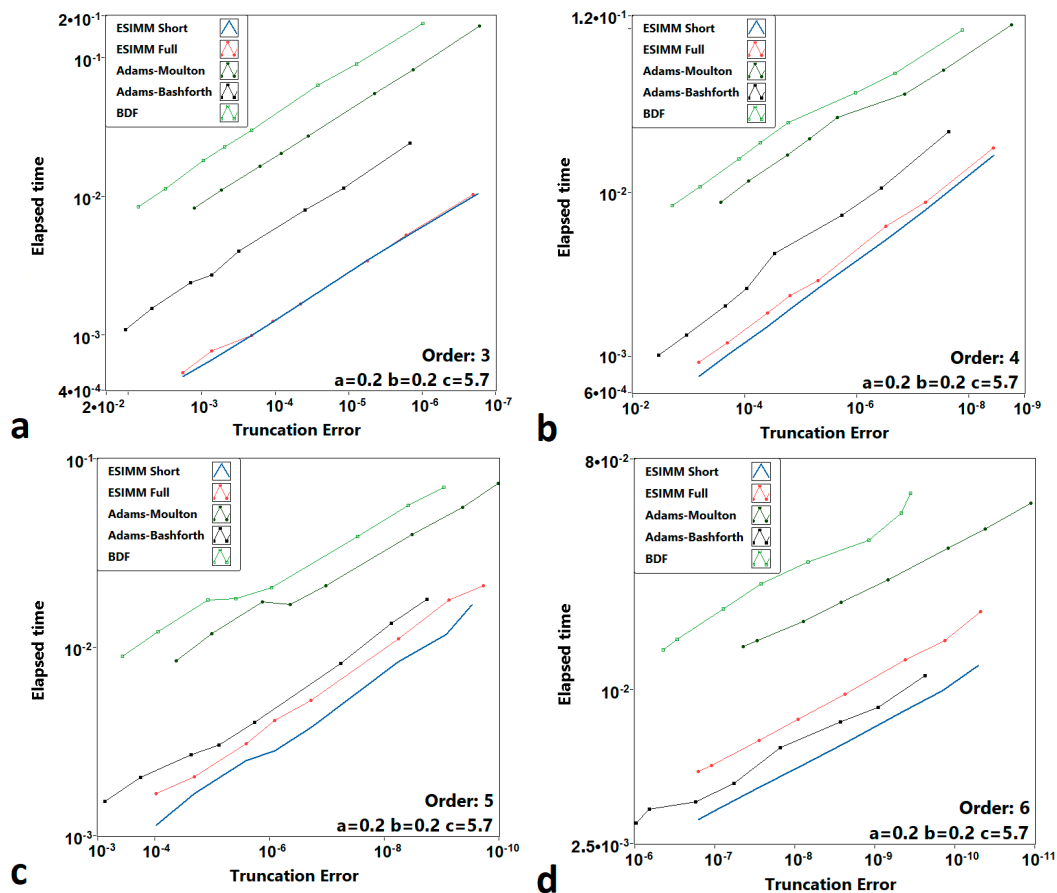


Figure 3. Performance plots obtained while simulating Rossler systems by various multistep methods with the accuracy (a) order 3, (b) order 4, (c) order 5 and (d) order 6.

3.2. Sprott System Case A and Case E

These two systems were proposed by J.C. Sprott in his famous paper [17] and can be written as following initial value problems:

$$\begin{cases} \frac{dx}{dt} = ay \\ \frac{dy}{dt} = -x + yz \\ \frac{dz}{dt} = b - y^2 \end{cases} \quad (9)$$

$$\begin{cases} \frac{dx}{dt} = yz \\ \frac{dy}{dt} = x^2 - y \\ \frac{dz}{dt} = d - 4x \end{cases} \quad (10)$$

Sprott Case A is a conservative chaotic system and exhibits chaotic behavior with parameter values $a = 1$, $b = 1$. The basic semi-implicit symmetric algorithm for the Sprott Case A system is as follows:

$$\begin{aligned} x^{n+0.5} &= x^n + \frac{h}{2} * (a * y^n) \\ y^{n+0.5} &= y^n + \frac{h}{2} * (y^n * z^n - x^{n+0.5}) \\ z^{n+1} &= z^n + h(b - y^{n+0.5} * y^{n+0.5}) \\ y^{n+1} &= \frac{y^{n+0.5} - x^{n+0.5} * \frac{h}{2}}{1 - \frac{h}{2} * z^{n+1}} \end{aligned}$$

$$x^{n+1} = x^{n+0.5} + \frac{h}{2} * (a * y^{n+1})$$

The semi-implicit basic method for the Sprott Case E system can be written as:

$$x^{n+0.5} = x^n + \frac{h}{2} * (y^n * z^n)$$

$$y^{n+0.5} = y^n + \frac{h}{2} * (x^{n+0.5} * x^{n+0.5} - y^n)$$

$$z^{n+1} = z^n + h(d - 4 * x^{n+0.5})$$

$$y^{n+1} = \frac{y^{n+0.5} + \frac{h}{2} * (x^{n+0.5} * x^{n+0.5})}{1 + \frac{h}{2}}$$

$$x^{n+1} = x^{n+0.5} + \frac{h}{2} * (y^{n+1} * z^{n+1})$$

Parameter value $d = 11$ corresponds to the chaotic regime of oscillations. Figure 4 represents the performance analysis for all investigated methods including the proposed ESIMM algorithms.

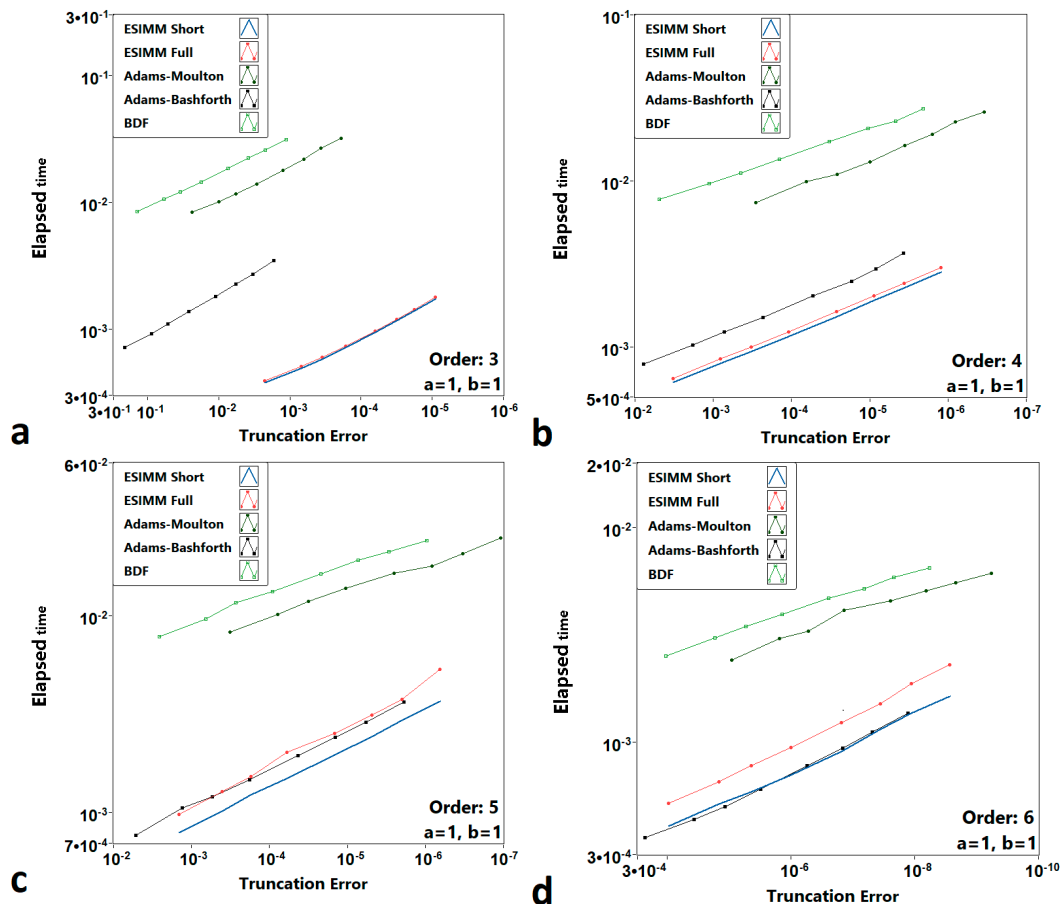


Figure 4. Performance plots for simulating Sprott Case A system by various multistep methods with the accuracy of order 3, (b) order 4, (c) order 5 and (d) order 6.

The rest of the parameters used to perform the numerical experiments with the system (9) are given in Table 4. One can see from Figure 4 that the performance of the ESIMM methods in the case of the conservative chaotic system Sprott Case A was unexpectedly lower despite the symmetry of the underlying basic semi-implicit method.

Table 4. Parameters used in experiments with the Sprott Case A system.

Order	3	4	5	6
Step values, sec.			0.04	
			0.03	
			0.025	
			0.02	
			0.015	
			0.012	
			0.01	
			0.008	
Initial conditions	(1; 1; 1)			
Number of code runs	55			

Thus, the numerical efficiency of the ESIMM algorithm decreases with the increase in accuracy order due to the one extra calculation of RHS function per order, unlike classical multistep methods.

The simulation of the Sprott Case E system confirmed the preliminary results. One can see from Figure 5 that the performance gain of ESIMM methods over the Adams–Bashforth method decreases as the order of the scheme increases. However, the overall results are inspiring due to the known fact that the application of Adams–Bashforth method with orders of accuracy 6 and higher is rare due to the stability issues. Table 5 gives some information on the simulation parameters.

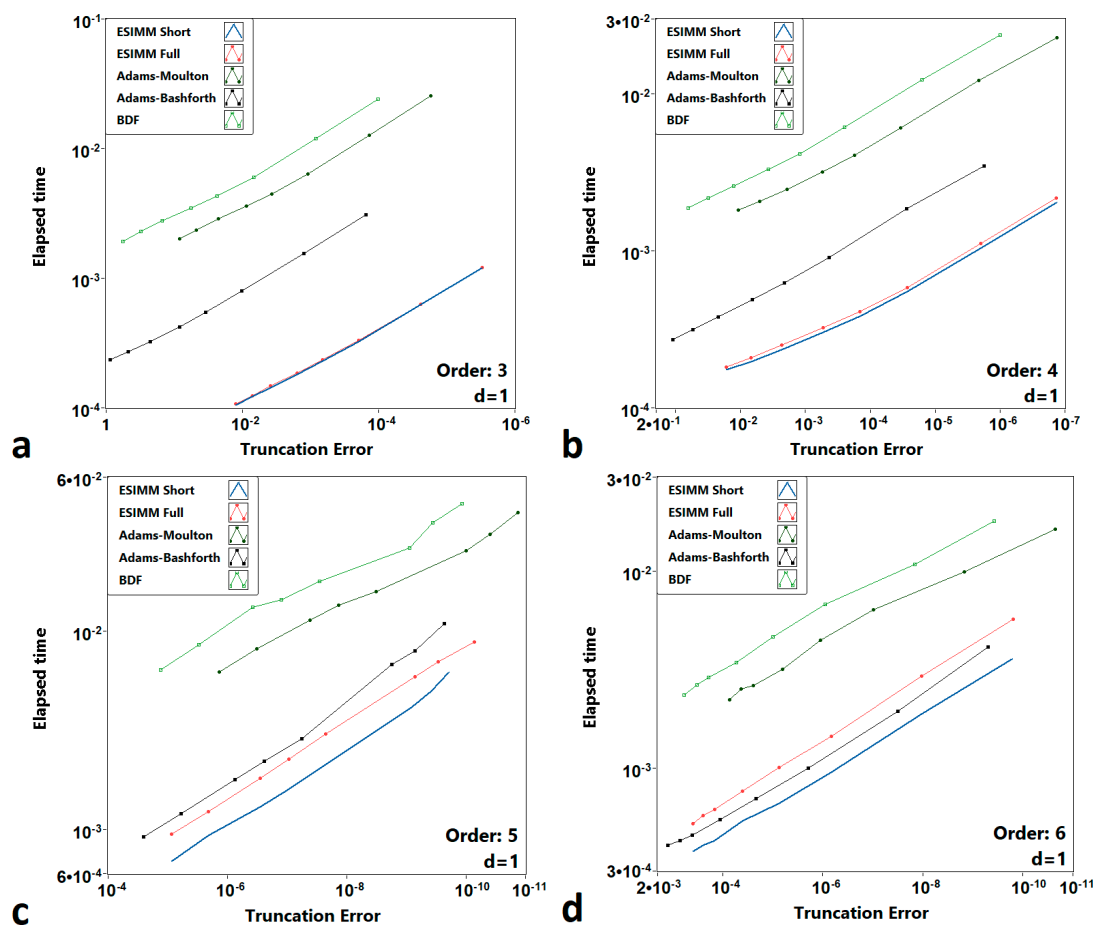


Figure 5. Performance plots for simulating Sprott Case E system by various multistep methods with the accuracy (a) order 3, (b) order 4, (c) order 5 and (d) order 6.

Table 5. Parameters used in experiments with the Sprott Case E system.

Order	3	4	5	6
Step values, sec.	0.085		0.02	0.06
	0.07		0.015	0.055
	0.055		0.01	0.05
	0.04		0.008	0.04
	0.03		0.006	0.03
	0.02		0.003	0.02
	0.01		0.0025	0.01
	0.005		0.002	0.005
Parameter d value	1			
Initial conditions	(1; 0; −2)			
Simulation time, sec.	30			
Number of code runs	40			

By analyzing results shown in Figures 4 and 5, one can conclude that the semi-implicit ESIMM methods perform well with both conservative (Sprott case A) and dissipative (Sprott case E) systems. They are significantly faster than all other listed methods and give us a rather accurate representation of systems behavior. However, implicit multistep methods such as Adams–Moulton can be more accurate than ESIMM in higher-order schemes, but the proposed method proposes the best trade-off between precision and simulation time, which is extremely important in long-term simulations. To investigate the impact of stiffness on the investigated methods, we additionally considered one more system that is well known as a testbench for numerical integration methods.

3.3. Van der Pol System

This classical nonlinear oscillator was proposed by Balthasar van der Pol [18] and can be written as the following IVP:

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = m(1 - x^2)y - x \end{cases} \quad (11)$$

This system is important for our study because the value of parameter m determines the stiffness of the system. The algorithm of the basic symmetric CD method for the system (11) is as follows:

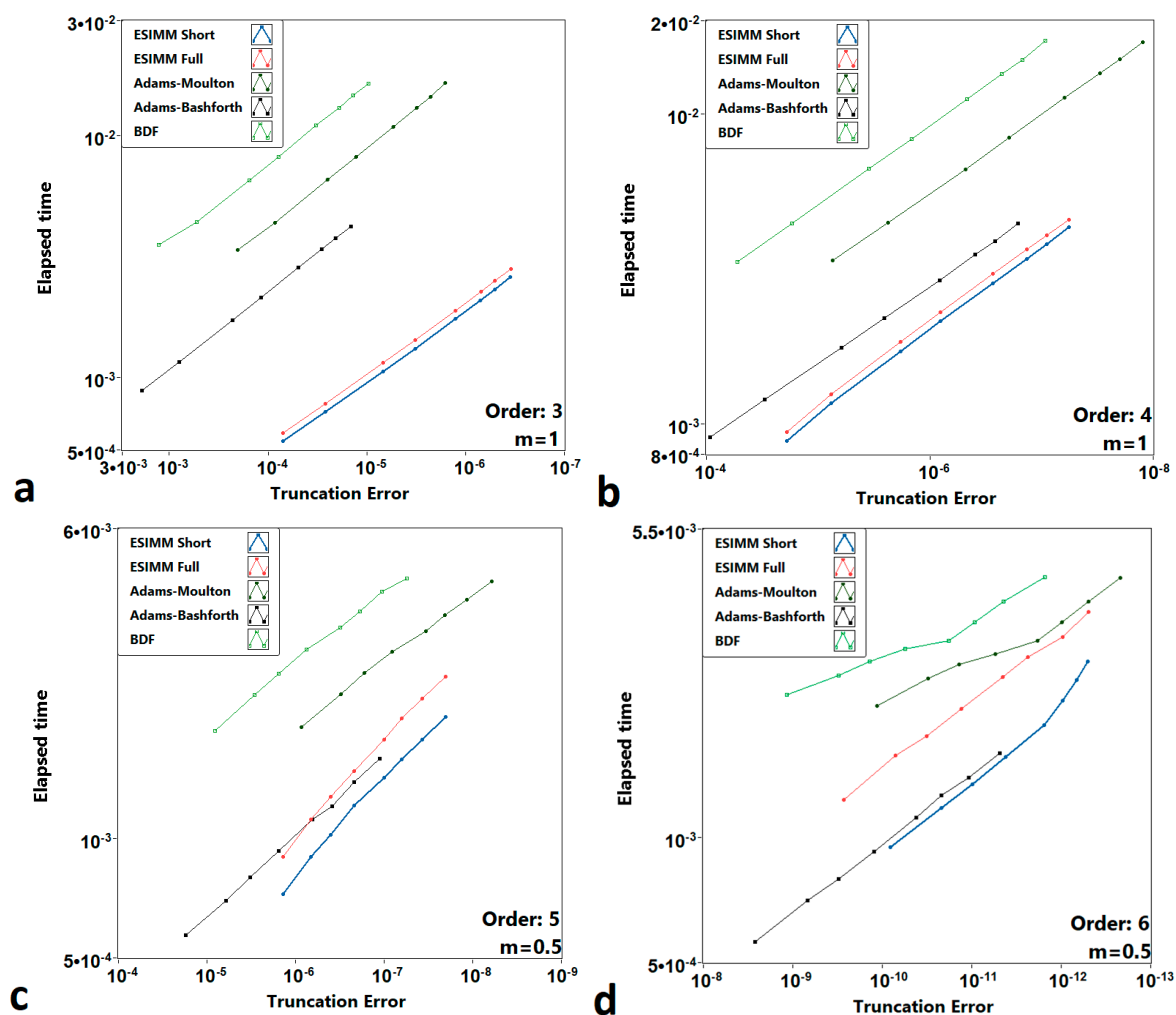
$$\begin{aligned} y^{n+0.5} &= y^n + \frac{h}{2} * (m * (1 - x^n * x^n) * y^n - x^n) \\ x^{n+1} &= x^n + h * y^{n+0.5} \\ y^{n+1} &= \frac{y^{n+0.5} - \frac{h}{2} * x^{n+1}}{1 - \frac{h}{2} * m * (1 - x^{n+1} * x^{n+1})} \end{aligned}$$

The simulation parameters are given in Table 6. The parameter m initially was set to 1, but for higher-order schemes we reduced it to 0.5 to avoid the loss of stability of explicit methods.

Table 6. The parameters used for experiments with the van der Pol system.

Order	3	4	5	6
Step values	0.02		0.01	0.01
	0.015		0.008	0.008
	0.01		0.007	0.007
	0.008		0.006	0.006
	0.006		0.005	0.0055
	0.005		0.0045	0.005
	0.0045		0.004	0.0045
	0.004		0.0035	0.004
Parameter value	1		0.5	
Initial conditions	(1; 0)		(5; 3)	(2; 1)
Simulation time, sec.	30			
Number of code runs	40			

We varied the initial conditions during the experiments to ensure that they do not significantly affect the obtained results. The performance plots for the van der Pol system are shown in Figure 6.

**Figure 6.** The performance plots of van der Pol system obtained for investigated multistep methods with the accuracy (a) order 3, (b) order 4, (c) order 5 and (d) order 6.

One can see from Figure 6 that implicit linear multistep methods are slightly more accurate than other considered algorithms. Explicit methods such as Adams–Bashforth are, on the other hand, as fast as semi-implicit ones but they lose their accuracy due to less numerical stability. Further increase in m leads to total stability loss for Adams–Bashforth methods. We illustrated this in the final experiment with this system, the results of which are shown in Figure 7. One can see the higher stability of semi-implicit multistep methods in comparison with their well-known explicit counterparts. However, the Adams–Moulton method appears to be the most precise among the investigated algorithms, but the “short” version of the ESIMM solver generally outperforms it due to lower computational costs.

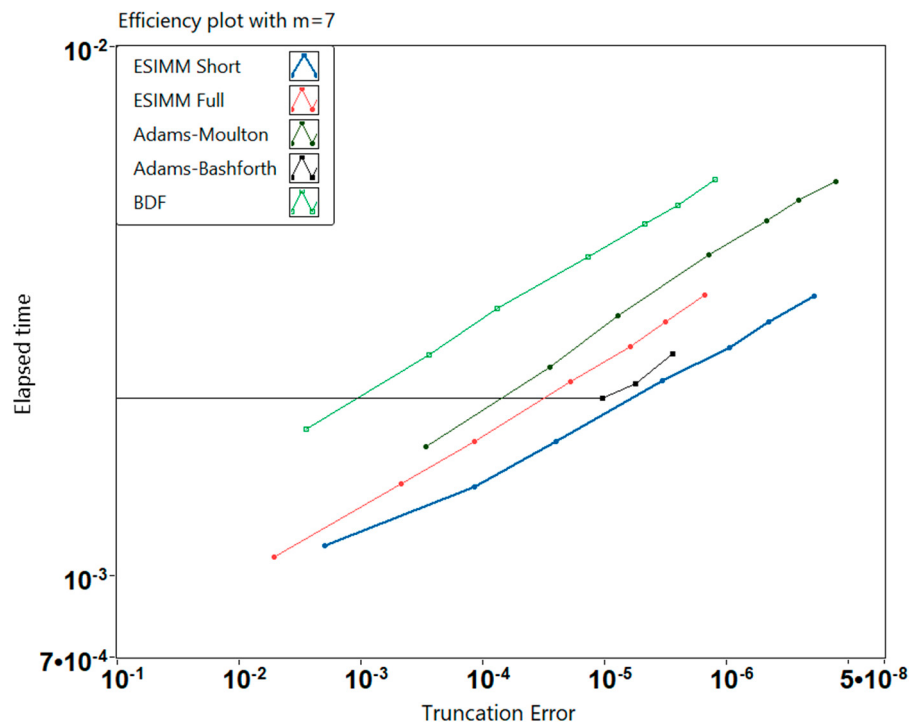


Figure 7. The performance plots for the van der Pol system with $m = 7$ obtained for investigated multistep integration methods of order 6. The stability loss on larger stepsizes can be observed for the explicit Adams–Bashforth method.

4. Stability Analysis

4.1. Order Plots and Convergence Test

To experimentally evaluate the convergence of the methods under investigation, we applied the approach known as order plots. This type of analysis involves the definition of global truncation error for a numerical method of order N , which can be generally described as $E = O(h)^N$, where O is a system-dependent constant and h is integration stepsize [19]. In our experiment, the truncation error was estimated numerically through the simulation of the Rossler nonlinear oscillator by all methods under investigation with stepsize h and $h/2$. Then, order plots can be constructed as $E(h)/E(h/2)$ vs. stepsize h . An order plot for an “ideal” numerical method of accuracy order 4 should be a horizontal line equal to 16, or for order 5 equal to 32, etc. Thus, the investigated finite-difference scheme can be considered more stable if its order plot is situated higher on the Y-axis and converges faster to the “ideal” line while stepsize decreases. One should note that this method evaluates the convergence and order preservation only empirically and for a certain test problem. We used the 2-norm of maximum truncation errors of all state variables of the system to estimate $E(h)$ and $E(h/2)$.

Figures 8 and 9 show the order plots for all investigated solvers of order 4 and 5, respectively. We omitted the plot for the “full” ESIMM method because it possesses minimal differences from the “short” version in this experiment. One can see from Figure 8 that the proposed method outperforms

both explicit and implicit linear multistep methods on the Rossler problem, being closer to the ideal method of order 4 ($E(h)/E(h/2) = 16$). Figure 9 represents real order evaluation for multistep methods of order 5. The left-hand part of the order plot can be used to roughly evaluate the convergence empirically. One can see that the ESIMM method quickly converges to the “ideal” order 5 ($E(h)/E(h/2) = 32$) while stepsize decreases. Other methods under investigation show similar convergence, but the explicit Adams–Bashforth method loses stability at a certain stepsize value. Thus, we can conclude that the proposed semi-implicit extrapolation methods can be more stable than explicit linear multistep methods, and possess higher practical order of accuracy for a given set of problems. This confirms the theoretically predicted properties of such methods. However, the analytical investigation of numerical stability is still needed, including the plotting of stability regions.

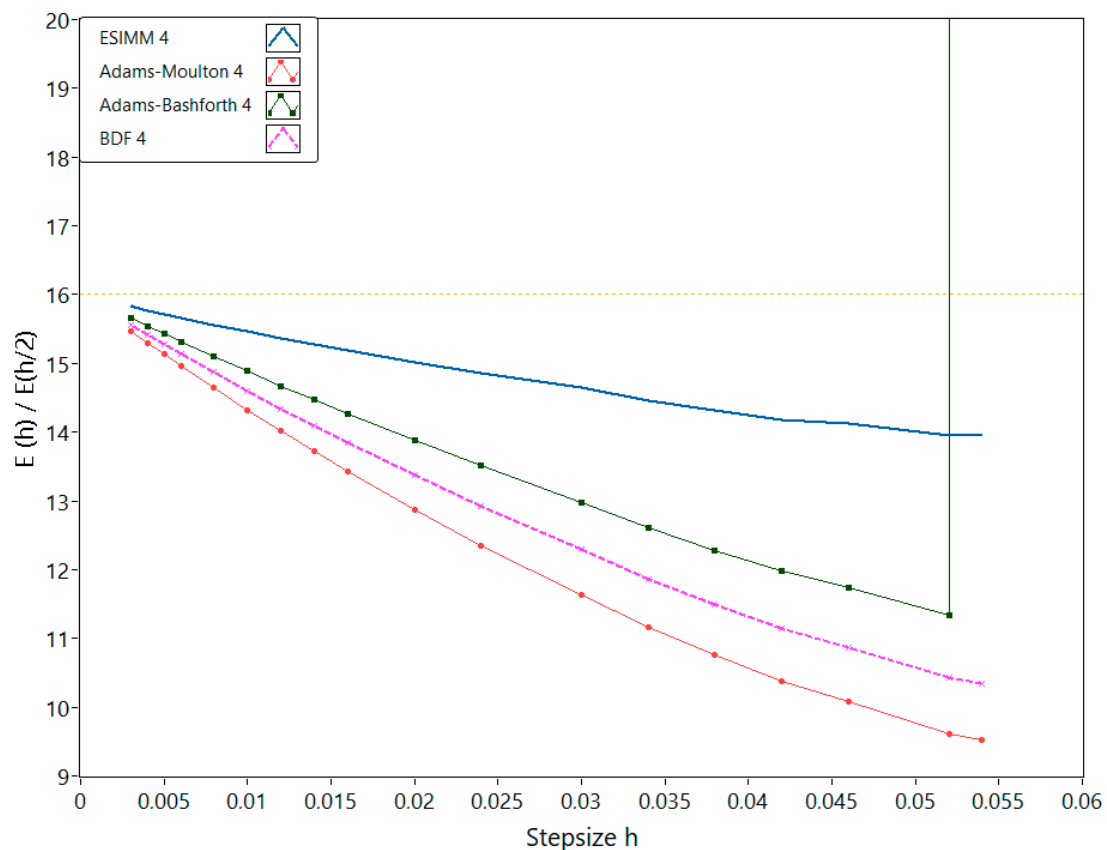


Figure 8. The order plots for investigated methods of order 4 while solving the Rossler problem. The Adams–Bashforth method experiences instability on large stepsizes.

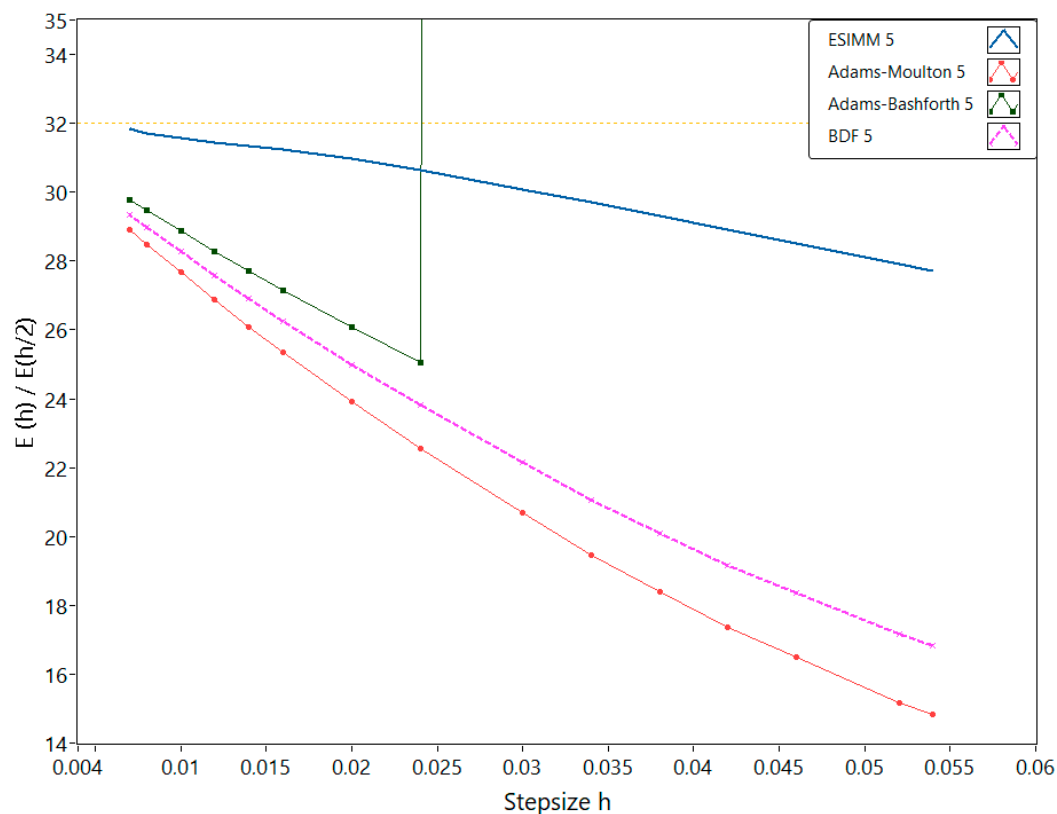


Figure 9. The order plots for investigated methods of order 5 while solving the Rossler problem. Note that the superiority of ESIMM methods over implicit solvers increases with the order of accuracy. Adams–Bashforth 5 method experiences stability problems on large stepsizes.

4.2. Stability Regions

To analytically evaluate the stability of ESIMM methods we will use a technique based on the known approach with 2-dimensional test problems [11,20]. Let us recall that the semi-implicit methods exist only for the initial value problems of dimension 2 or greater, and one needs a special technique here. Consider a 2-dimensional autonomous test problem

$$\dot{x} = Ax, \quad A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Let eigenvalues of the matrix A be

$$\lambda_{12} = \sigma \pm j\omega.$$

Then, the contents of the matrix can be determined by two free parameters, denote them $r \geq 0$ (eccentricity coefficient) and $k \geq 0$ (asymmetry coefficient):

$$\begin{aligned} a &= kd, \\ b &= rc, \\ c &= -\sqrt{\frac{1}{r}(\lambda_1^2 - (1+k)d\lambda_1 + kd^2)}, \\ d &= \frac{2\sigma}{1+k}. \end{aligned}$$

The shape of the stability region depends on k , reaching its maximal size when $k = 1$, which corresponds to the Jordan normal form of the matrix, and minimal size when $k = 0$ or $k = \infty$, which corresponds to the Frobenius normal form of the matrix. The stability region for the

2-dimensional problem is the area in the complex plane where the absolute values of complex conjugate eigenvalues of the matrix $R(Ah)$ are not greater than 1.

Let stability region be $R_h(Ah)$ for stepsize h , $R_{2h}(Ah)$ for stepsize $2h$, and so on. Then, for the two-dimensional test problem, the s -stage multistep semi-implicit method can be written in terms of a shift operator $z : x_{n+s} = z^s x_n$ as follows:

$$z^s x_0 = k_1 R_h(Ah) z^{s-1} x_0 + k_2 R_{2h}(Ah) z^{s-2} x_0 + \dots + k_s R_{sh}(Ah) x_0. \quad (12)$$

From the recurrence equation (12), implying that $x_0 \neq (0 \ 0)^T$, and denoting the identity matrix as I , one can obtain the characteristic equation of the method:

$$\det(Iz^s - k_1 R_h(Ah) z^{s-1} - k_2 R_{2h}(Ah) z^{s-2} - \dots - k_s R_{sh}(Ah)) = 0.$$

Then, the stability region of the ESIMM method is a region where the maximal absolute value of the root z_{max} of the characteristic polynomial $\det(Iz^s - k_1 R_h(Ah) z^{s-1} - \dots - k_s R_{sh}(Ah))$ satisfies the condition $|z_{max}| \leq 1$.

Stability regions of “full” and “short” ESIMM methods are almost the same and are given in Figures 10 and 11. For comparison, we provide stability regions of explicit Adams-Bashforth and implicit Adams-Moulton linear multistep methods as Figures 12 and 13, respectively.

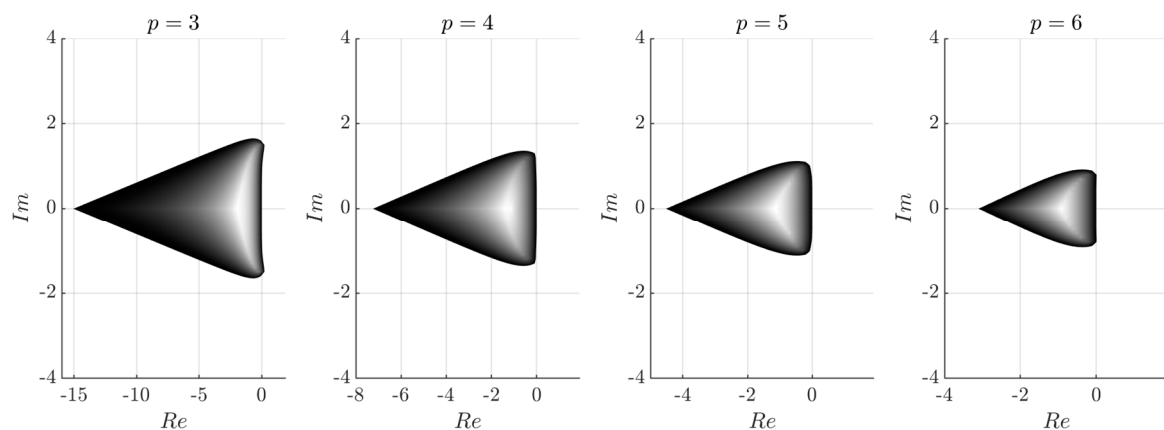


Figure 10. Stability regions of ESIMM methods of accuracy order p for symmetric test problem matrix ($k = 1$).

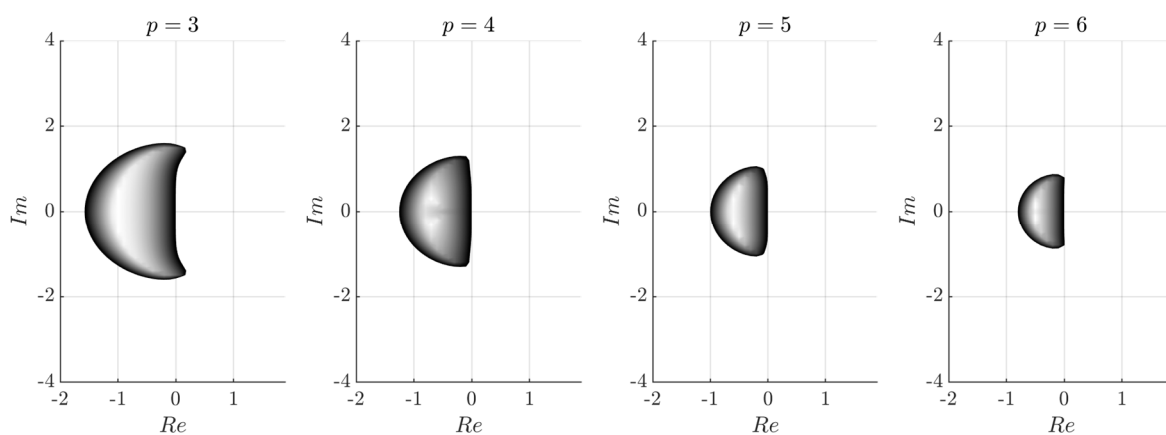


Figure 11. Stability regions of ESIMM methods of accuracy order p for asymmetric test problem matrix ($k = 0$).

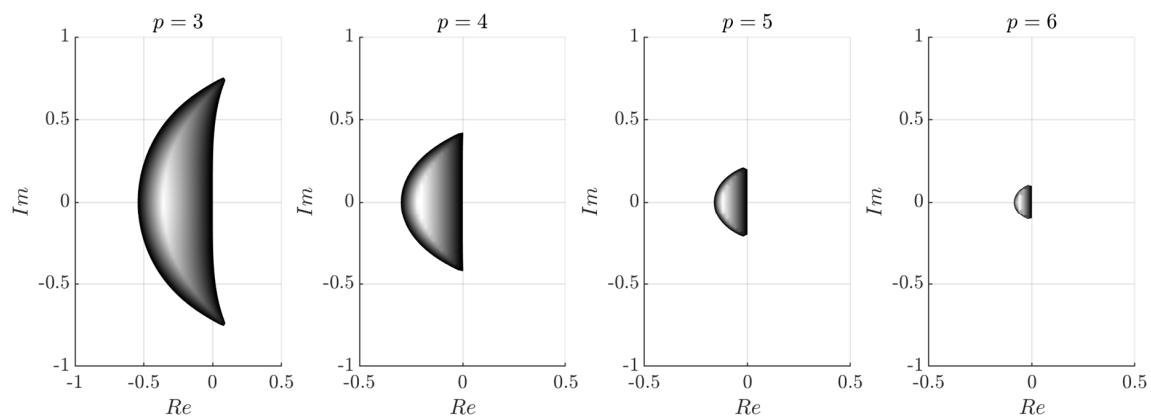


Figure 12. Stability regions of Adams-Bashforth methods of order p .

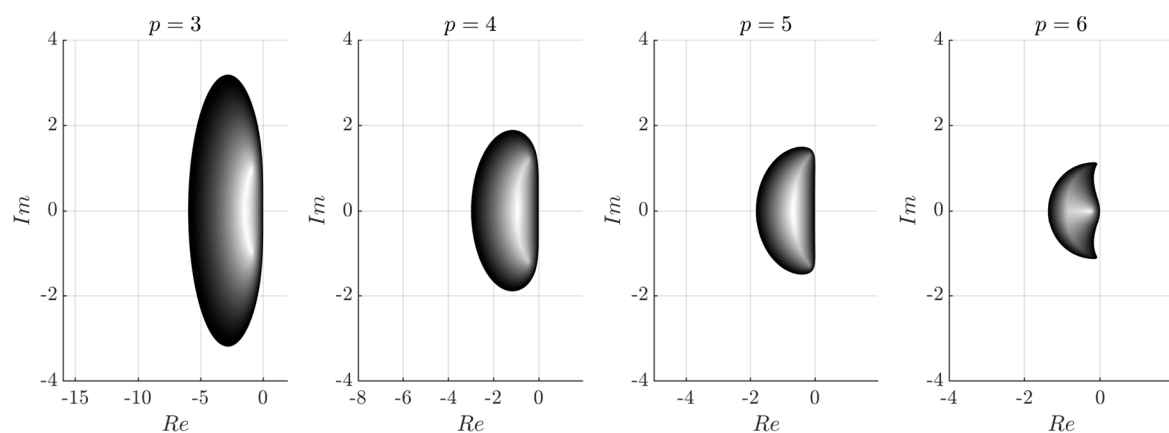


Figure 13. Stability regions of Adams-Moulton methods of order p .

Observing the results of stability analysis, one can see that theoretically predicted properties of the ESIMM methods and the results of the order plot analysis were successfully confirmed. ESIMM methods possess numerical stability that is comparable with implicit Adams-Moulton methods and is far superior to the stability of explicit Adams-Bashforth methods. Despite the fact that the numerical stability of semi-implicit methods strongly depends on the symmetry of the investigated system, we can declare that even in a worst case (full asymmetry of the simulated system) their stability is better than the stability of explicit linear multistep methods which allows implementing these methods with larger stepsizes.

5. Conclusions and Discussion

In this paper, we proposed novel multistep extrapolation methods based on semi-implicit symmetric integration. We explicitly show that the proposed ESIMM integration technique in many cases of nonlinear systems possesses greater performance than well-known multistep ODE solvers. We investigated the order and convergence of the investigated methods experimentally through the order plots using Rossler chaotic system as a testbench. One of the important research questions was the analytical estimation of the numerical stability of ESIMM methods. The investigation of the numerical stability of semi-implicit methods is rather complicated due to their non-trivial call to the right-hand-side function. We used a technique based on second-order linear test problems [11,20] which allowed plotting stability regions for ESIMM and comparing them with stability regions of linear multistep methods.

We explicitly show that the proposed ESIMM solvers are more numerically stable than explicit Adams-Bashforth methods, and, in some cases, implicit Adams-Moulton methods. Moreover, the stepsize range where the order of ESIMM methods preserves, theoretically allows constructing

highly efficient multistep extrapolation solvers with adaptive stepsize. Being based on numerical extrapolation, ESIMM methods possess the same capacity to construct built-in estimators of local truncation error as single-step extrapolation solvers. Besides, the ESIMM ODE solvers do not need the concatenation procedure for output data obtained during the starting method and main solution, which also simplifies and fastens the algorithm. The computational cost of the proposed methods was estimated experimentally and is visualized through the performance plots. The program implementation of the proposed methods is significantly simpler than for implicit multistep solvers. Speaking generally, the ESIMM methods are almost as simple as explicit multistep methods, being almost as precise as implicit multistep methods. One of the main drawbacks of the ESIMM method is the necessity to increase the number of basic method calculations by 1 per each increase in the accuracy order. Therefore, high-order solvers, based on the proposed technique, can be less efficient.

In our further studies we plan to implement ESIMM solvers with adaptive stepsize, which requires the development of special step control and coefficient recalculation techniques. ESIMM methods based on other symmetric integrators, e.g., implicit midpoint method or composition solvers [10], are also to be studied.

Author Contributions: Conceptualization, D.B. and A.K.; data curation, A.T. (Artem Terentev); formal analysis, P.F. and A.K.; funding acquisition, D.B.; investigation, D.B., P.F. and A.T. (Artem Terentev); methodology, D.B., A.T. (Aleksandra Tutueva) and A.K.; project administration, D.B.; resources, A.T. (Aleksandra Tutueva); software, D.B., A.T. (Aleksandra Tutueva), P.F. and A.T. (Artem Terentev); supervision, D.B.; validation, A.T. (Aleksandra Tutueva) and A.T. (Artem Terentev); Visualization, P.F.; writing—original draft, D.B., P.F. and A.K.; writing—review and editing, A.T. (Aleksandra Tutueva). All authors have read and agreed to the published version of the manuscript.

Funding: Denis N. Butusov was supported by a grant from the Russian Science Foundation (RSF), project 19-71-00087.

Acknowledgments: We thank the anonymous reviewers for their careful reading of our manuscript and insightful comments. The authors are very grateful to Professor Higinio Ramos for his helpful suggestions during the elaboration of this paper.

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

References

1. Chen, C.; Zhang, X.; Liu, Z.; Zhang, Y. A new high-order compact finite difference scheme based on precise integration method for the numerical simulation of parabolic equations. *Adv. Differ. Equ.* **2020**, *2020*, 1–28. [\[CrossRef\]](#)
2. Papanikolaou, T.; Tsoulis, D. Assessment of numerical integration methods in the context of low Earth orbits and inter-satellite observation analysis. *Acta Geod. Geophys.* **2016**, *51*, 619–641. [\[CrossRef\]](#)
3. Ascher, U.M.; Petzold, L.R. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1998. [\[CrossRef\]](#)
4. Hairer, E.; Wanner, G. *Solving Ordinary Differential Equations II. Stiff and Differential—Algebraic Problems*; Springer: Berlin/Heidelberg, Germany, 2010; Volume II.
5. Okuonghae, R.I.; Ikhile, M.N.O. On the construction of high order $A(\alpha)$ -stable hybrid linear multistep methods for stiff IVPs and ODEs. *J. Numer. Anal. Appl.* **2012**, *15*, 231–241. [\[CrossRef\]](#)
6. Okuonghae, R.I.; Ikhile, M.N.O. A class of hybrid linear multistep methods with $A(\alpha)$ -stability properties for stiff IVPs in ODEs. *J. Numer. Math.* **2013**, *21*, 157–172. [\[CrossRef\]](#)
7. Olatunji, P.O.; Ikhile, M.N.O. Variable order nested hybrid multistep methods for stiff ODEs. *J. Math. Comput. Sci.* **2020**, *10*, 78–94. [\[CrossRef\]](#)
8. Cash, J.R. Second Derivative Extended Backward Differentiation Formulas for the Numerical Integration of Stiff Systems. *SIAM J. Numer. Anal.* **1981**, *18*, 21–36. [\[CrossRef\]](#)
9. Yousefzadeh, N.; Hojjati, G.; Abdi, A. Construction of Implicit–Explicit Second-Derivative BDF Methods. *Bull. Iran. Math. Soc.* **2018**, *44*, 991–1006. [\[CrossRef\]](#)
10. Casas, F.; Escorihuela-Tomás, A. Composition Methods for Dynamical Systems Separable into Three Parts. *Mathematics* **2020**, *8*, 533. [\[CrossRef\]](#)
11. Butusov, D.; Ostrovskii, V.Y.; Karimov, A.I.; Andreev, V.S. Semi-Explicit Composition Methods in Memcapacitor Circuit Simulation. *Int. J. Embed. Real Time Commun. Syst.* **2019**, *10*, 37–52. [\[CrossRef\]](#)

12. Butusov, D.; Karimov, A.I.; Tutueva, A.V. Hardware-targeted semi-implicit extrapolation ODE solvers. In Proceedings of the 2016 International Siberian Conference on Control and Communications (SIBCON), (IEEE), Moscow, Russia, 12–14 May 2016; pp. 1–6.
13. Butusov, D.; Tutueva, A.V.; Homitskaya, E.S. Extrapolation Semi-implicit ODE solvers with adaptive timestep. In Proceedings of the 2016 XIX IEEE International Conference on Soft Computing and Measurements (SCM), (IEEE), Saint-Petersburg, Russia, 25–27 May 2016; pp. 137–140.
14. Deuflhard, P. A study of extrapolation methods based on multistep schemes without parasitic solutions. *Z. Für Angew. Math. Phys.* **1979**, *30*, 177–189. [[CrossRef](#)]
15. Dormand, J.; Prince, P. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26. [[CrossRef](#)]
16. Rossler, O. An equation for continuous chaos. *Phys. Lett. A* **1976**, *57*, 397–398. [[CrossRef](#)]
17. Sprott, J.C. Some simple chaotic flows. *Phys. Rev. E* **1994**, *50*, R647–R650. [[CrossRef](#)] [[PubMed](#)]
18. Cartwright, M.L. Balthazar van der Pol. *J. Lond. Math. Soc.* **1960**, *35*, 367–376. [[CrossRef](#)]
19. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I*; Springer Series in Computational Mathematics; Berlin, Germany, 1993; Volume 8.
20. Hairer, E.; Hochbruck, M.; Iserles, A.; Lubich, C. Geometric Numerical Integration. *Oberwolfach Rep.* **2006**, 805–882. [[CrossRef](#)]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).