

Article

Adaptive Stepsize Control for Extrapolation Semi-Implicit Multistep ODE Solvers

Denis Butusov 

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

Abstract: Developing new and efficient numerical integration techniques is of great importance in applied mathematics and computer science. Among the variety of available methods, multistep ODE solvers are broadly used in simulation software. Recently, semi-implicit integration proved to be an efficient compromise between implicit and explicit ODE solvers, and multiple high-performance semi-implicit methods were proposed. However, the computational efficiency of any ODE solver can be significantly increased through the introduction of an adaptive integration stepsize, but it requires the estimation of local truncation error. It is known that recently proposed extrapolation semi-implicit multistep methods (ESIMM) cannot operate with existing local truncation error (LTE) estimators, e.g., embedded methods approach, due to their specific right-hand side calculation algorithm. In this paper, we propose two different techniques for local truncation error estimation and study the performance of ESIMM methods with adaptive stepsize control. The first considered approach is based on two parallel semi-implicit solutions with different commutation orders. The second estimator, called the “double extrapolation” method, is a modification of the embedded method approach. The introduction of the double extrapolation LTE estimator allowed us to additionally increase the precision of the ESIMM solver. Using several known nonlinear systems, including stiff van der Pol oscillator, as the testbench, we explicitly show that ESIMM solvers can outperform both implicit and explicit linear multistep methods when implemented with an adaptive stepsize.

Keywords: differential equations; semi-implicit methods; multistep method; extrapolation; adaptive stepsize



Citation: Butusov, D. Adaptive Stepsize Control for Extrapolation Semi-Implicit Multistep ODE Solvers. *Mathematics* **2021**, *9*, 950. <https://doi.org/10.3390/math9090950>

Academic Editor: Fabio Caraffini

Received: 7 April 2021
Accepted: 21 April 2021
Published: 23 April 2021

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



Copyright: © 2021 by the author. 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

Many real-world processes can be mathematically described by systems of ordinary differential equations. One of the common ways to implement the solution of ODEs in discrete computers is numerical integration. The increasing complexity and scale of simulated systems requires a corresponding increase in the efficiency of ODE solvers. Thus, the development of novel numerical integration methods is of great interest in applied mathematics. In the last decade, many highly efficient solvers were proposed, including Falkner-type block methods [1,2], symmetric multistep methods [3], Störmer–Cowell methods [4], and single-step composition schemes [5]. Great attention is usually paid to the convergence and stability of designed methods, especially in the case of partial differential equations solvers [6]. Long-term simulations are also of great interest in the field, and many highly efficient and stable multistep schemes were developed recently [7,8]. The latest studies show that semi-implicit methods provide a reasonable compromise between highly stable implicit solvers and computationally efficient explicit solvers [9]. Moreover, being implemented in composition [5] and splitting [10] schemes, semi-implicit methods possess some properties of geometric integrators [11], e.g., time-reversibility and symmetry [9]. It is also possible to construct highly efficient single-step ODE solvers using the Aitken–Neville extrapolation [12]. In paper [13], the authors proposed a novel extrapolation semi-implicit multistep method (ESIMM), which combines the strengths of multistep schemes

with the benefits of the extrapolation solvers, using the semi-implicit method as a basic integrator. It has been shown that ESIMM methods outperform classical multistep solvers, such as Adams–Bashforth (AB), Adams–Moulton (AM) and backward differentiation formula (BDF) being implemented with a fixed integration stepsize. However, the most advanced approaches for solving ODEs usually involve adaptive stepsize techniques [1,4]. Conventional local truncation error (LTE) estimators, such as the embedded methods approach [14], appeared to be barely suitable for ESIMM due to its specific method of right-hand side (RHS) calculation. Each step of the ESIMM solver in the variable–stepsize version requires recalculating all extrapolation coefficients due to their explicit dependency on the stepsize values used in previous iterations. Therefore, a new efficient LTE estimator and step control algorithm are needed to perform the ESIMM with an adaptive stepsize efficiently. In this paper, we propose two new techniques of adaptive stepsize control for ESIMM solvers. We develop a formula for recalculating the method’s coefficients for an arbitrary sequence of stepsizes and propose two algorithms for local truncation error estimation. The first LTE estimator is based on the difference between two parallel solutions with different commutation orders [15] and the second, more efficient approach uses the double–extrapolation technique. Simulating chaotic systems is of great interest in applied mathematics due to the complex behavior and high sensitivity of their finite-difference models to the truncation error accumulation. Thus, we have chosen Rössler [16], Dadras-Momeni [17], Nose-Hoover [18–20] and van der Pol nonlinear systems [21] as test ODEs. In several computational experiments, we have shown that the ESIMM methods with adaptive stepsizes outperform both explicit and implicit linear multistep solvers, while simulating dissipative and conservative chaotic systems, as well as stiff nonlinear equations. The rest of the paper is organized as follows: in Section 2, the basic architecture of ESIMM methods is recalled, and a formula to recalculate the coefficients of the method is given. Section 3 introduces the novel LTE estimators and step control technique for ESIMM methods and provides the experimental estimation of the computational efficiency of the proposed method using a test set of dynamical systems. Finally, in Section 4, some conclusions are given.

2. Materials and Methods

Let us revisit the general idea of ESIMM solvers. These multistep methods use an extrapolation of the values obtained on previous solution steps together with newly calculated values, combining the multistep and extrapolation approaches to increase the accuracy order of the resulting scheme.

Let H_1 be a stepsize used to calculate the solution for IVP of order p at the point $x_{(n+1)}$ from the point $x_{(n)}$. Then, the sequence of local error terms e is as follows:

$$x(t_{(n+1)}) = x_{(n+1)} + H_1^{p+1}e_{p+1} + H_1^{p+2}e_{p+2} + \dots \tag{1}$$

In order to obtain the solution at the point $x_{(n+1)}$ by calculating the increment from the point $x_{(n-1)}$ we will use the step value $H_2 = H_1 + H_n$, where H_n is a step value which steps from point $x_{(n-1)}$ to $x_{(n)}$ (see Figure 1).

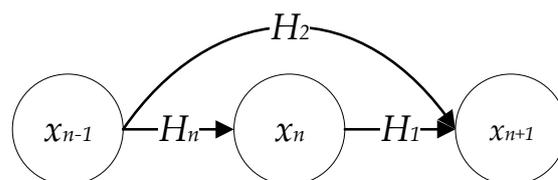


Figure 1. The basic principle of choosing a stepsize in the ESIMM method.

Now the sequence of local error terms can be calculated as follows:

$$x(t_{(n+1)}) = x_{(n+1)} + H_2^{p+1}e_{p+1} + H_2^{p+2}e_{p+2} + \dots \tag{2}$$

The further algorithm is described in [13] and generally follows the idea of the extrapolation methods [11,22] in multistep form.

The main idea of the ESIMM method is to use the solution with the stepsize H_1 (denote it as T_1) and the solution at the same point with the stepsize H_2 (denote it as T_2) in the following way:

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

Coefficients k_1 and k_2 are chosen in accordance with general extrapolation principles, taking into account the ratio of truncation errors in adjacent stages so that the following conditions are satisfied:

$$\begin{cases} k_1 + k_2 = 1 \\ k_1 + \frac{H_2^{p+1}}{H_1^{p+1}} k_2 = 0 \end{cases} \tag{4}$$

where H_1 and H_2 are the step values used to calculate the solutions at point $x_{(n+1)}$, $x_{(n)}$ and $x_{(n-1)}$, respectively. Figure 2 illustrates the general scheme of the ESIMM method of order 3.

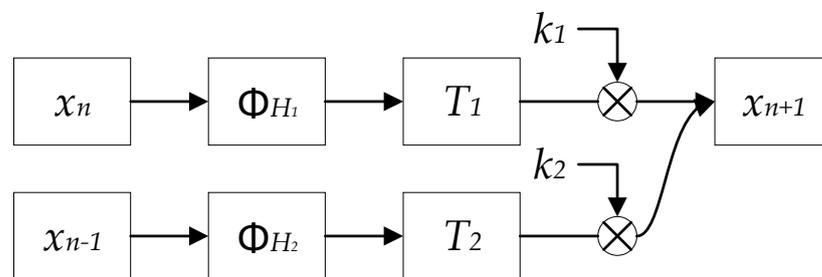


Figure 2. Graphical interpretation of 3rd order ESIMM method. Two parallel stages are combined to obtain a third-order solution. Φ denotes a basic CD self-adjoint integration method.

Following the same logic, one can describe the computational process for ESIMM of order 4 (Figure 3), where T_{ij} represents a component of the extrapolation table on the i -th stage on the j -th line, and k_{ij}^u is a coefficient by the first or the second term, respective to the value of u , which can be either 1 or 2.

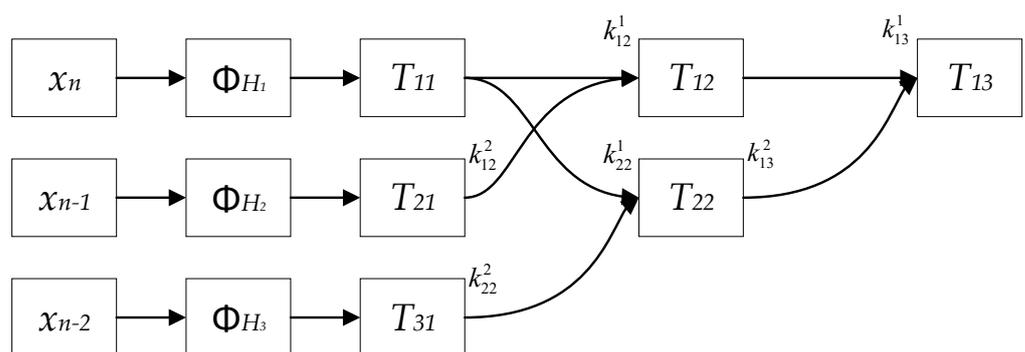


Figure 3. Graphical interpretation of 4th order ESIMM method.

By using matrix formalism, one can obtain coefficients for the first stage of extrapolation according to Equation (3). Because we have to eliminate the error term e_{p+1} in both (1) and (2), the coefficients are as follows:

$$\begin{aligned} k_{12} &= \begin{pmatrix} k_{12}^1 \\ k_{12}^2 \end{pmatrix}; \begin{pmatrix} 1 & 1 \\ H_1^{p+1} & H_2^{p+1} \end{pmatrix} k_{12} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \\ k_{22} &= \begin{pmatrix} k_{22}^1 \\ k_{22}^2 \end{pmatrix}; \begin{pmatrix} 1 & 1 \\ H_1^{p+1} & H_3^{p+1} \end{pmatrix} k_{22} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned} \tag{5}$$

After completing the first extrapolation step, the equations for two of our approximations for the point $x(t_{n+1})$, which we will call T_{12} and T_{22} , are as follows:

$$\begin{aligned} T_{12} &= x_{n+1} + (k_{12}^1 H_1^{p+2} + k_{12}^2 H_2^{p+2})e_{p+2} + \dots; \\ T_{22} &= x_{n+1} + (k_{22}^1 H_1^{p+2} + k_{22}^2 H_3^{p+2})e_{p+2} + \dots \end{aligned} \tag{6}$$

The vector form can be used to rewrite those expressions as follows:

$$\begin{aligned} T_{12} &= x_{n+1} + (k_{12}^1, k_{12}^2) \begin{pmatrix} H_1^{p+2} \\ H_2^{p+2} \end{pmatrix} e_{p+2} + \dots; \\ T_{22} &= x_{n+1} + (k_{22}^1, k_{22}^2) \begin{pmatrix} H_1^{p+2} \\ H_3^{p+2} \end{pmatrix} e_{p+2} + \dots \end{aligned} \tag{7}$$

Using a similar approach, one can obtain the final equations for k_{12} and k_{14} :

$$\begin{aligned} k_{12} &= \begin{pmatrix} k_{13}^1 \\ k_{13}^2 \end{pmatrix}; \left(k_{12}^T \begin{pmatrix} 1 \\ H_1^{p+2} \\ H_2^{p+2} \end{pmatrix} \quad k_{22}^T \begin{pmatrix} 1 \\ H_1^{p+2} \\ H_3^{p+2} \end{pmatrix} \right) k_{13} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \\ k_{14} &= \begin{pmatrix} k_{22}^1 \\ k_{22}^2 \end{pmatrix}; \left(k_{13}^T \begin{pmatrix} 1 \\ k_{12}^T \begin{pmatrix} H_1^{p+3} \\ H_2^{p+3} \end{pmatrix} \\ k_{22}^T \begin{pmatrix} H_1^{p+3} \\ H_3^{p+3} \end{pmatrix} \end{pmatrix} \quad k_{23}^T \begin{pmatrix} 1 \\ k_{12}^T \begin{pmatrix} H_1^{p+3} \\ H_2^{p+3} \end{pmatrix} \\ k_{32}^T \begin{pmatrix} H_1^{p+3} \\ H_4^{p+3} \end{pmatrix} \end{pmatrix} \right) k_{14} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned} \tag{8}$$

Using this technique, one can obtain extrapolation coefficients for a “full” variation of the ESIMM method, which requires solving the matrix equation at every step. It is possible to significantly reduce the number of calculations using the so-called “short” version of the same method, previously described in [13]. We will use the “short” ESIMM method throughout this paper.

Let us modify a short ESIMM method to make it suitable for implementation with a variable stepsize by interpreting our step values as the following system of linear algebraic equations:

$$x_{n+1} = \sum_{i=1}^s k_i T_i. \tag{9}$$

Considering the Taylor expansion for $k_i \cdot x(t_{n+1})$ through T_i where T_i represents the first components of the extrapolation table, we obtain coefficients for the short version of the ESIMM method of the desired order:

$$\begin{cases} k_1 x(t_{n+1}) = k_1 T_1 + k_1 H_1^{p+1} e_{p+1} + k_1 H_1^{p+2} e_{p+2} + \dots + k_1 H_1^{p+s} e_{p+s} \\ k_2 x(t_{n+1}) = k_2 T_2 + k_2 H_2^{p+1} e_{p+1} + k_2 H_2^{p+2} e_{p+2} + \dots + k_2 H_2^{p+s} e_{p+s} \\ \dots \\ k_s x(t_{n+1}) = k_s T_s + k_s H_s^{p+1} e_{p+1} + k_s H_s^{p+2} e_{p+2} + \dots + k_s H_s^{p+s} e_{p+s} \end{cases} \tag{10}$$

Keeping in mind that that the sum of coefficients k_i should equal 1, while the sum of all the other terms by any power of H equals zero, one can write the following system of algebraic equations:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ H_1^{p+1} & H_2^{p+1} & H_3^{p+1} & \dots & H_s^{p+1} \\ H_1^{p+2} & H_2^{p+2} & H_3^{p+2} & \dots & H_s^{p+2} \\ \dots & \dots & \dots & \dots & \dots \\ H_1^{p+s} & H_2^{p+s} & H_3^{p+s} & \dots & H_s^{p+s} \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ \dots \\ k_s \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix} \tag{11}$$

This approach can be used for obtaining coefficients of the ESIMM method of an arbitrary accuracy order and proved to be efficient while using a symmetric basic method. In the current study, we use the semi-implicit self-adjoint CD method [9,22] of order 2 as a basic method for the ESIMM solver.

The important note here is that one needs to recalculate all extrapolation coefficients on each step of ESIMM while implementing it with variable stepsizes because these coefficients are directly influenced by the step values used in previous iterations. Thus, we recommend using the “short” version of the ESIMM method in practical applications because it is more computationally efficient, provides fewer overhead calculations, and maintains the same high accuracy as the “full” version.

3. Results

In this section, we propose two algorithms for estimating the local truncation error while controlling the stepsize of the ESIMM solver. Then, we analyze the performance of the extrapolation semi-implicit multistep (ESIMM) method with a variable stepsize in comparison with classical multistep methods, including Adams–Bashforth, Adams–Moulton and Backward Differentiation Formula. In our study, we use the Dormand–Prince 8 (DOPRI8) method [23] to obtain the reference solution. All experiments were conducted using NI LabVIEW 2020 environment with double floating-point data precision.

3.1. Rössler Attractor

This dissipative chaotic system was proposed by Otto E. Rössler [16] and can be described as follows:

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases}, \tag{12}$$

where a, b and c are system parameters. The basic semi-implicit CD method for the system is as follows:

$$\begin{aligned} y^{n+0.5} &= y^n + \frac{H}{2}(x^n + ay^n); \\ z^{n+0.5} &= z^n + \frac{H}{2}(b + z^n x^n - cz^n); \\ x^{n+1} &= x^n + H(-y^{n+0.5} - z^{n+0.5}); \\ z^{n+1} &= \frac{z^{n+0.5} + \frac{H}{2}b}{1 - \frac{H}{2}x^{n+1} + c\frac{H}{2}}; \\ y^{n+1} &= \frac{y^{n+0.5} + \frac{H}{2}x^{n+1}}{1 - a\frac{H}{2}}. \end{aligned} \tag{13}$$

We carried out experiments with the following parameter values: $a = 0.2, b = 0.2, c = 5.7$. The timing parameters and initial conditions are given in Table 1.

Table 1. Initial conditions and simulation parameters for experiments with the Rössler system.

Order of Accuracy	3	4	5
Initial Conditions	(0.95; 0; −1.5)	(3; 0; −0.3)	(0.35; 0; −2)
Simulation Time (s.)	15		
Step Values (s.)			
Minimum	0.001		
Starting	0.005		
Maximum	1		

To implement the adaptive stepsize control, one needs to estimate the local truncation error (LTE) of the method. The first LTE estimator for the ESIMM method considered in this study is based on the difference between two parallel solutions with different commutation orders, generally following the idea used in [15] to create new metrics for quantifying chaotic dynamics.

The term “commutations order” basically means changing the order of the state variables calculation, as shown in Equation (14), where a second possible commutation of CD method for the Rössler system is given:

$$\begin{aligned}
 z^{n+0.5} &= z^n + \frac{H}{2}(b + z^n x^n - cz^n); \\
 y^{n+0.5} &= y^n + \frac{H}{2}(x^n + ay^n); \\
 x^{n+1} &= x^n + H(-y^{n+0.5} - z^{n+0.5}); \\
 y^{n+1} &= \frac{y^{n+0.5} + \frac{H}{2}x^{n+1}}{1 - a\frac{H}{2}}; \\
 z^{n+1} &= \frac{z^{n+0.5} + \frac{H}{2}b}{1 - \frac{H}{2}x^{n+1} + c\frac{H}{2}}.
 \end{aligned}
 \tag{14}$$

Figure 4 shows the scheme of the LTE evaluation using two different commutations of the basic CD method executed in parallel.

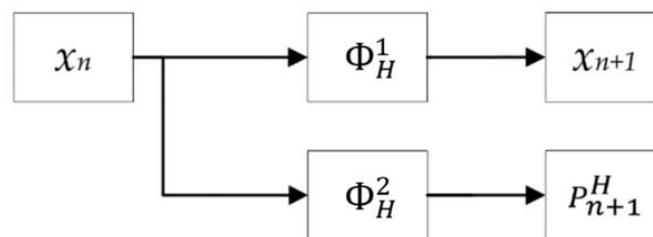


Figure 4. Block diagram of the commutations-based LTE estimator. Φ_H^1 and Φ_H^2 denote two different commutations of a basic CD method.

The local truncation error is then estimated as the difference between ESIMM solutions obtained with basic methods (13) and (14). The step control law is as follows [14]:

$$H_{n+1} = H_n \left| \frac{Tol}{d(x_n)} \right|^{\frac{1}{m}} + O(H_n),
 \tag{15}$$

where m is the accuracy order of the scheme, Tol is the desired tolerance value and $d(x_n)$ is a local error between two solutions.

Figure 5 illustrates the behavior of the stepsize in the ESIMM scheme with the commutation-based LTE estimator being compared with the explicit Adams–Bashforth linear multistep method with a standard stepsize control algorithm [14].

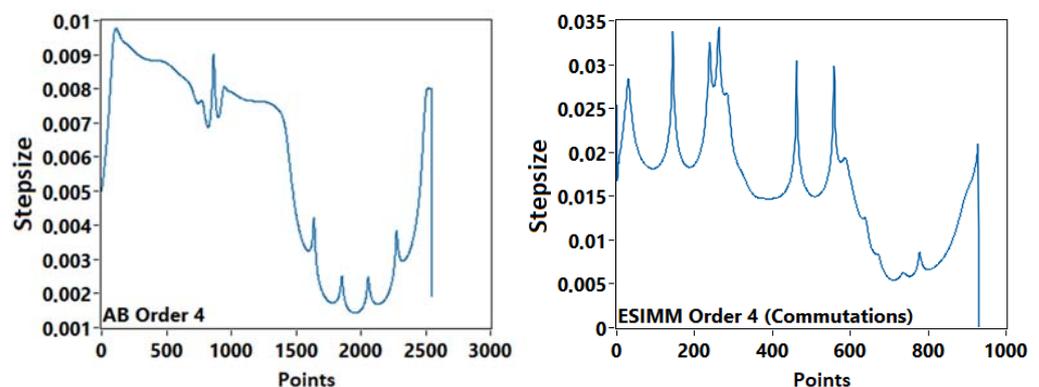


Figure 5. Step behavior for Adams–Bashforth and ESIMM methods of order 4 while solving the Rössler system. A commutation-based LTE estimator is used for controlling the stepsize in the ESIMM scheme.

One can see from Figure 5 that the ESIMM solver demonstrates a sharper curve of the stepsize dynamics graph, but at the same time, it tends to use higher step values, resulting in fewer RHS function calculations. It follows from the fewer number of steps required

for obtaining a solution with ESIMM. Although these results were already promising, we additionally developed an alternative step control approach to further increase the ESIMM solver efficiency.

The second proposed technique for local truncation error estimation derives from the embedded method approach, which is broadly used in single-step extrapolation solvers and embedded Runge–Kutta methods [14]. To estimate the local truncation error, a difference between two basic methods with the same commutation orders but with different stepsizes was used. One solution was obtained using the step value H and the other contains a composition of two consequent steps with $H/2$. Taking into account that the basic method is of order 2, the local truncation error demonstrates a quadratic dependence from the chosen integration step. Both solutions are extrapolated to increase the overall accuracy as follows:

$$x_{n+1} = \frac{4P_{n+1}^{H/2} - P_{n+1}^H}{3}, \tag{16}$$

where P_{n+1}^H and $P_{n+1}^{H/2}$ are two solutions found with step sizes of H and $H/2$, respectively. We called this approach “double extrapolation” because the ESIMM method already includes one extrapolation procedure in the computation algorithm. After acquiring two ESIMM solutions, one can use the difference between them as the estimation of LTE and calculate the new stepsize value using Equation (15) with $d(x_n) = \sqrt{\left(|x_{(n+1)} - P_{n+1}^H|^2\right)}$. Figure 6 illustrates the modification of the basic method in the ESIMM scheme, incorporating a double extrapolation estimator. Figure 7 illustrates the benefits of the double extrapolation approach in the form of performance plots obtained during the Rössler system simulation. The performance plots show the dependence between the achieved precision and the real program execution time needed to obtain the solution with the desired tolerance. The execution time was averaged from 100 simulation runs. The simulation parameters were set in accordance with Table 1.

As is shown in Figure 7, using the double-extrapolation technique results in a significant increase in the calculation speed on higher tolerance values and a slight improvement in the accuracy of the solution on low tolerances. Note that the stepsize curve in Figure 7b is smoother than that in Figure 7a. Figure 8 represents the performance plots for all methods under investigation of accuracy order 3 and 5 while solving the Rössler system.

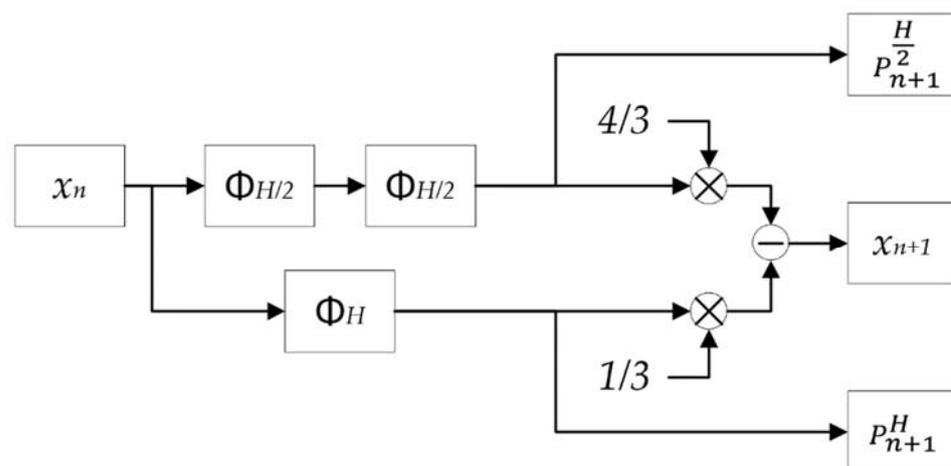


Figure 6. Block diagram of double extrapolation LTE estimator. Φ denotes a basic CD self-adjoint integration method.

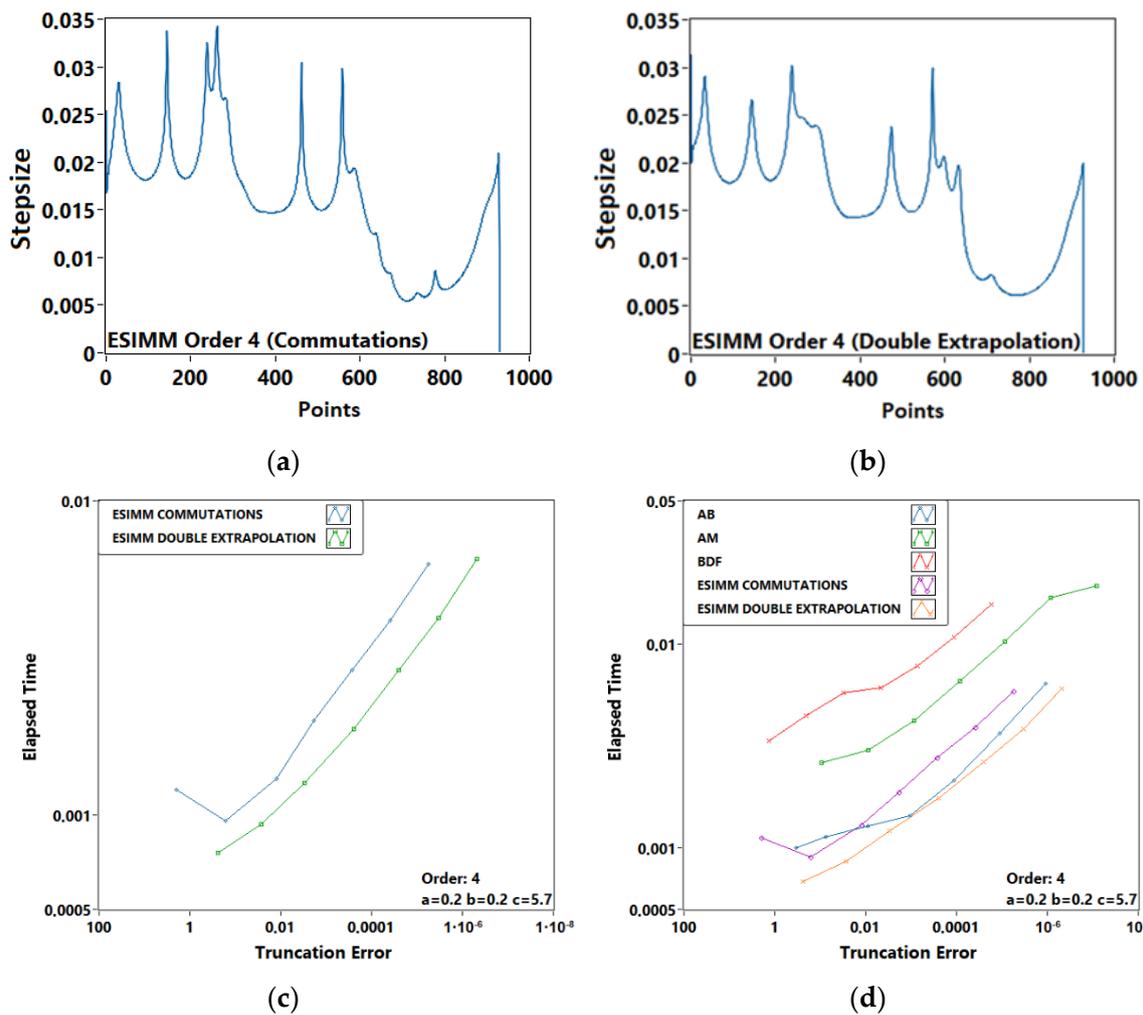


Figure 7. (a) Step behavior using commutation-based LTE estimator; (b) step behavior with double extrapolation estimator. (c) The comparison of ESIMM solver with commutation-based LTE estimator vs. ESIMM with the double-extrapolation LTE estimator. The precision gain can be clearly observed; (d) comparison of AB, AM and BDF methods with two proposed versions of the adaptive ESIMM algorithm.

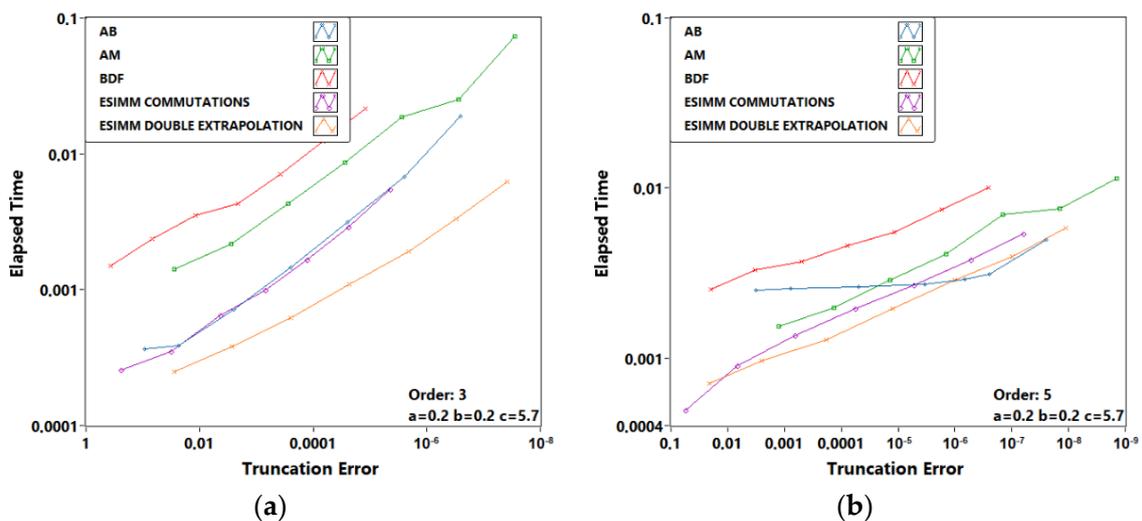


Figure 8. Performance plots obtained for investigated multistep methods while simulating the Rössler system with the accuracy order 3 (a) and 5 (b).

One can see from Figure 8 that ESIMM methods with adaptive stepsizes appear to be more computationally efficient than other multistep methods under investigation, namely, Adams–Bashforth, Adams–Moulton and Backward Differentiation Formula. The observed differences in performance with explicit methods, such as Adams–Bashforth solver, become less noticeable with the increase in the accuracy order, but are still significant in terms of accuracy and calculation speed on lower tolerance values. This fact can be explained by the extra RHS calculations introduced to the ESIMM scheme by both proposed LTE estimation algorithms. However, the higher stability of ESIMM methods still provides some superiority over Adams–Bashforth solver on lower tolerance values. Let us consider some other test dynamical systems to verify the obtained results in a series of computational experiments.

3.2. Dadras–Momeni Attractor

This chaotic system was described by S. Dadras and H.R. Momeni [11] and can be written as the following initial value problem:

$$\begin{cases} \frac{dx}{dt} = y - ax + byz \\ \frac{dy}{dt} = cy - xz + z \\ \frac{dz}{dt} = dxy - mz \end{cases}, \tag{17}$$

where a, b, c, d and m are system parameters. The basic semi-implicit CD solver for the system is formulated as follows:

$$\begin{aligned} x^{n+0.5} &= \frac{x^n + \frac{H}{2}(y^n + by^n z^n)}{1 + a\frac{H}{2}}; \\ z^{n+0.5} &= \frac{z^n + \frac{H}{2}dx^{n+0.5}y^n}{1 + \frac{H}{2}m}; \\ y^{n+0.5} &= \frac{y^n + \frac{H}{2}(-x^{n+0.5}z^{n+0.5} + z^{n+0.5})}{1 - \frac{H}{2}c}; \\ y^{n+1} &= y^{n+0.5} + \frac{H}{2}(cy^{n+0.5} - x^{n+0.5}z^{n+0.5} + z^{n+0.5}); \\ z^{n+1} &= z^{n+0.5} + \frac{H}{2}(dx^{n+0.5}y^{n+1} - mz^{n+0.5}); \\ x^{n+1} &= x^{n+0.5} + \frac{H}{2}(y^{n+1} - ax^{n+0.5} + by^{n+1}z^{n+1}). \end{aligned} \tag{18}$$

We carried out experiments with the Dadras–Momeni system using the following parameter values: $a = 3, b = 2.7, c = 4.7, d = 2, m = 9$. Other simulation parameters are given in Table 2.

Table 2. The initial conditions and simulation parameters for Dadras–Momeni system.

Order of Accuracy	3	4	5
Initial Conditions	(1; 0; −1)		
Simulation Time (s.)	10		
Step Values (s.)			
Minimum	10 ^{−10}		
Starting	0.0001		
Maximum	1		

One can see from Figure 9 that the efficiency of the ESIMM methods decreases with the increase in accuracy order due to the necessity of calculating additional RHS functions per order. Nevertheless, the ESIMM solver with the double-extrapolation LTE estimator provides the best computational efficiency among the investigated methods. It is of certain interest to estimate the performance of the proposed multistep methods while simulating a conservative nonlinear system.

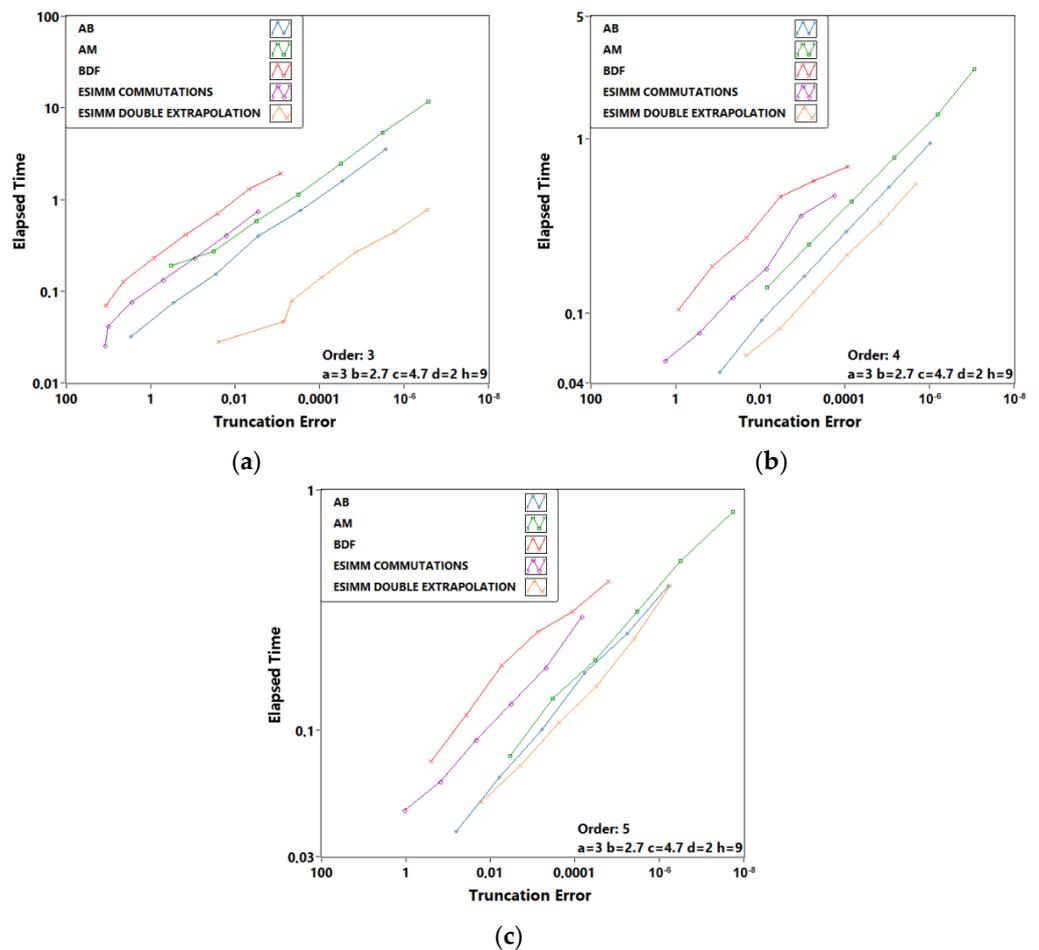


Figure 9. Performance plots obtained while simulating Dadras–Momeni system by investigated multistep methods of accuracy order 3 (a), 4 (b) and 5 (c).

3.3. Nose–Hoover Attractor

This conservative chaotic system was proposed by S. Nose and W.G. Hoover [18–20] and can be formulated as follows:

$$\begin{cases} \frac{dx}{dt} = ay \\ \frac{dy}{dt} = -x + yz \\ \frac{dz}{dt} = d - y^2 \end{cases}, \tag{19}$$

where a and d are system parameters. The basic semi-implicit algorithm for the system is as follows:

$$\begin{aligned} y^{n+0.5} &= \frac{y^n - x^n \frac{H}{2}}{1 - \frac{H}{2} z^n}; \\ z^{n+0.5} &= z^n + \frac{H}{2} (d - y^{n+0.5} y^{n+0.5}); \\ x^{n+1} &= x^n + H a y^{n+0.5}; \\ z^{n+1} &= z^{n+0.5} + \frac{H}{2} (d - y^{n+0.5} y^{n+0.5}); \\ y^n &= y^{n+0.5} + \frac{H}{2} (y^{n+0.5} z^{n+1} - x^{n+1}). \end{aligned} \tag{20}$$

We carried out the experiments with the Nose–Hoover system using the following set of parameter values: $a = 1, d = 1$. The rest of the simulation parameters are given in Table 3.

Table 3. The initial conditions and simulation parameters for Nose–Hoover system.

Order of Accuracy	3	4	5
Initial Conditions	(0.1; 0; −0.1)	(0.1; 0; −0.1)	(0.1; 0; −0.1)
Simulation Time (s.)	15		25
Step Values (s.)			
Minimum	10 ^{−5}		
Starting	0.001		
Maximum	1		

From Figure 10, one can see that while simulating a conservative system, ESIMM methods maintain good performance even in the case of higher-order schemes. This fact can possibly be explained by the geometric properties of the basic CD method, which provides extra precision while solving conservative ODEs. However, the proposed step control algorithms do not possess energy-preserving properties, and this phenomenon still is to be investigated. Note that other linear multistep methods under investigation appear to be much less accurate compared with results obtained for dissipative systems, while ESIMM gives promising results in both cases.

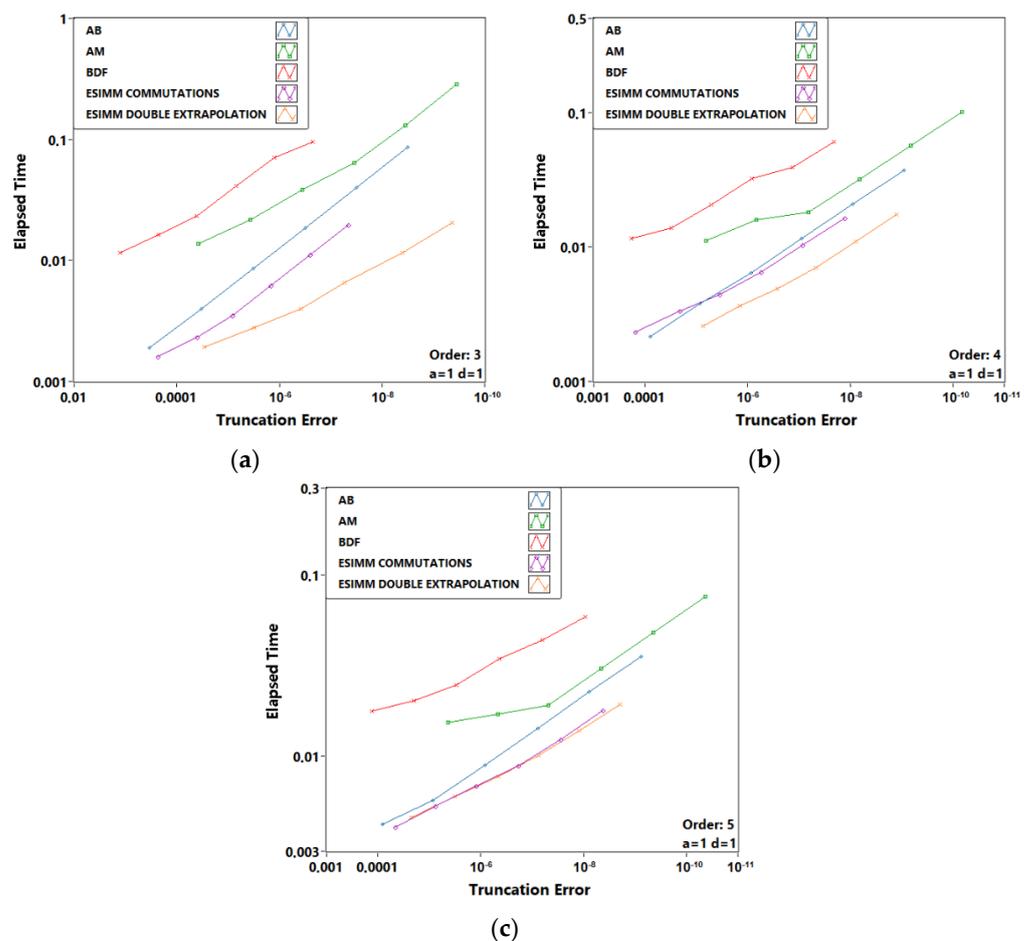


Figure 10. Performance plots obtained while simulating the conservative Nose–Hoover system by investigated multistep methods of accuracy order 3 (a), 4 (b) and 5 (c).

3.4. Van der Pol System

Considering the possibility that the abovementioned differences in the methods’ performance can originate from the varied stiffness of the previously studied systems, we

considered a well-known stiff dynamical system as a test ODE. This nonlinear oscillator is a traditional testbench for numerical integration methods and was proposed by Balthasar van der Pol [21]:

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = m(1 - x^2)y - x \end{cases} \quad (21)$$

where m is a parameter that defines the stiffness of the system. The basic semi-implicit CD algorithm for the van der Pol oscillator is as follows:

$$\begin{aligned} y^{n+0.5} &= y^n + \frac{H}{2}(m(1 - x^n x^n)y^n - x^n); \\ x^{n+1} &= x^n + Hy^{n+0.5}; \\ y^{n+1} &= \frac{y^{n+0.5} - \frac{H}{2}x^{n+1}}{1 - \frac{H}{2}m(1 - x^{n+1}x^{n+1})}. \end{aligned} \quad (22)$$

Experiments were carried out with $m = 55$ which corresponds to the medium-stiff scenario. The rest of the simulation parameters are presented in Table 4.

Table 4. The initial conditions and simulation parameters for van der Pol system.

Order of Accuracy	3	4	5
Initial Conditions	(0.1; 0)	(0.9; -0.2)	(0.9; -0.2)
Simulation Time (s.)	15	10	
Step Values (s.)			
Minimum		10^{-10}	
Starting		0.001	
Maximum	1	0.35	0.4

One can see from Figure 11 that the proposed adaptive ESIMM methods perform better than the other investigated multistep methods while simulating the stiff nonlinear system. The excellent numerical stability of the ESIMM scheme, analytically determined in [13], resulted in a better overall performance in the stiff system case. Moreover, it is possible to further increase the stability of the ESIMM solver by changing the complexity of the basic method and, therefore, achieving a perfect balance between the performance and stability of the solver.

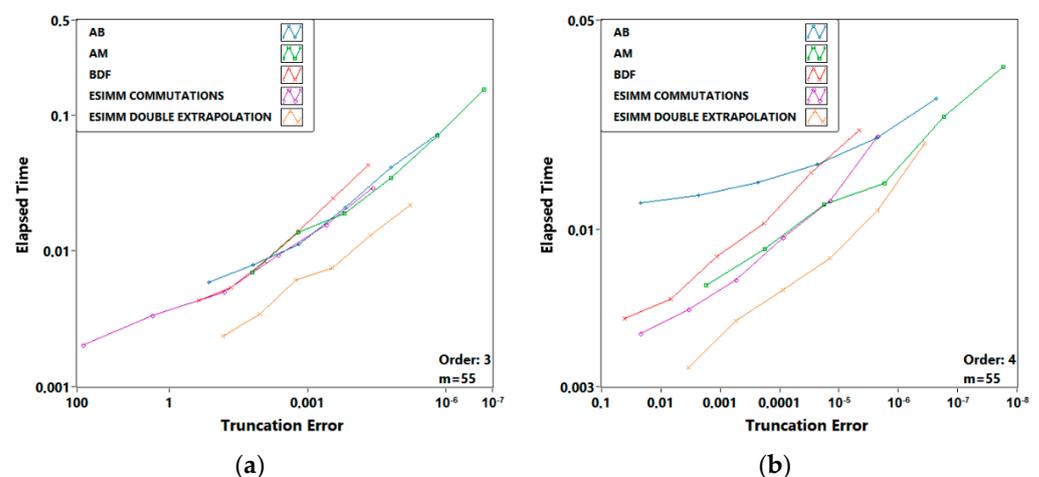


Figure 11. Cont.

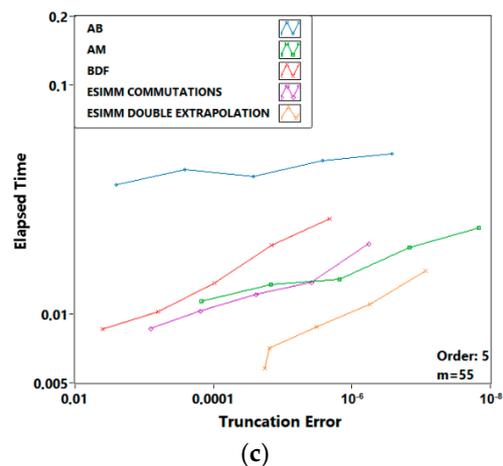


Figure 11. Performance plots obtained while simulating the stiff van der Pol system by investigated multistep methods of accuracy order 3 (a), 4 (b) and 5 (c).

4. Conclusions and Discussion

This study considered two possible techniques to implement extrapolation semi-implicit multistep ODE solvers with adaptive stepsizes. The first method, which uses the idea of estimating the local truncation error as a difference between two parallel solutions with different string commutations, proven to be a reliable and computationally cheap step control algorithm but, in some cases, it failed to outperform explicit multistep methods. The second approach, named the double extrapolation estimator, is based on a more traditional technique of LTE evaluation, which is similar to the embedded Runge–Kutta methods or embedded LTE estimators in extrapolation solvers. A double extrapolation ESIMM solver assumes the parallel solution consisting of the composition of two basic methods performed with a half-step. This approach allowed us to introduce an extra extrapolation procedure per stage to the ESIMM method, thereby increasing its accuracy. The double extrapolation technique not only allowed estimating the local error more precisely but also additionally increased the overall performance of the scheme. The numerical experiments included the simulation of four non-linear ODEs: the dissipative Rössler and Dadrás–Momeni systems, the conservative Nose–Hoover system and the classical stiff van der Pol stiff oscillator. ESIMM methods with the proposed adaptive step control techniques appeared to be the most efficient solvers among the tested integration schemes. Further studies will be devoted to the investigation of ESIMM methods with different basic methods, e.g., semi-implicit SED solver, semi-implicit midpoint methods and symmetrized explicit midpoint method. We will also consider the possibility of applying the proposed step control techniques to some other numerical methods, including PDE solvers.

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

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Ramos, H.; Singh, G.; Kanwar, V.; Bhatia, S. An efficient variable step-size rational Falkner-type method for solving the special second-order IVP. *Appl. Math. Comput.* **2016**, *291*, 39–51. [[CrossRef](#)]
2. Ramos, H.; Rufai, M.A. Third derivative modification of k-step block Falkner methods for the numerical solution of second order initial-value problems. *Appl. Math. Comput.* **2018**, *333*, 231–245. [[CrossRef](#)]
3. Hairer, E.; Lubich, C. Symmetric multistep methods for charged-particle dynamics. *SMAI J. Comput. Math.* **2017**, *3*, 205–218. [[CrossRef](#)]
4. Ramos, H.; Vigo-Aguiar, J. A note on step-size selection in the Störmer–Cowell methods. *J. Comput. Appl. Math.* **2005**, *175*, 149–159. [[CrossRef](#)]

5. Blanes, S.; Casas, F.; Murua, A. Splitting and composition methods in the numerical integration of differential equations. *Bol. Soc. Esp. Mat. Appl.* **2008**, *45*, 89–145.
6. Gottlieb, S.; Wang, C. Stability and Convergence Analysis of Fully Discrete Fourier Collocation Spectral Method for 3-D Viscous Burgers' Equation. *J. Sci. Comput.* **2012**, *53*, 102–128. [[CrossRef](#)]
7. Cheng, K.; Wang, C. Long Time Stability of High Order Multistep Numerical Schemes for Two-Dimensional Incompressible Navier-Stokes Equations. *SIAM J. Numer. Anal.* **2016**, *54*, 3123–3144. [[CrossRef](#)]
8. Yan, Y.; Chen, W.; Wang, C.; Wise, S.M. A Second-Order Energy Stable BDF Numerical Scheme for the Cahn-Hilliard Equation. *Commun. Comput. Phys.* **2018**, *23*, 572–602. [[CrossRef](#)]
9. Butusov, D.; Ostrovskii, V.Y.; Karimov, A.I.; Andreev, V.S. Semi-implicit Composition Methods in Memcapacitor Circuit Simulation. *Int. J. Embed. Real Time Commun. Syst.* **2019**, *10*, 37–52. [[CrossRef](#)]
10. Blanes, S.; Casas, F. *A Concise Introduction to Geometric Numerical Integration*; CRC Press: Boca Raton, FL, USA, 2016.
11. Hairer, E.; Wanner, G. *Solving Ordinary Differential Equations II. Stiff and Differential—Algebraic Problems*; Springer: Berlin/Heidelberg, Germany, 2010; Volume II.
12. Butusov, D.N.; 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), St. Petersburg, Russia, 25–27 May 2016.
13. Butusov, D.; Tutueva, A.; Fedoseev, P.; Terentev, A.; Karimov, A. Semi-Implicit Multistep Extrapolation ODE Solvers. *Mathematics* **2020**, *8*, 943. [[CrossRef](#)]
14. Hairer, E.; Wanner, G.; Nørsett, S.P. *Solving Ordinary Differential Equations I. Nonstiff Problems*; Springer: Berlin/Heidelberg, Germany, 1987.
15. Butusov, D.N.; Pesterev, D.O.; Tutueva, A.V.; Kaplun, D.I.; Nepomuceno, E.G. New technique to quantify chaotic dynamics based on differences between semi-implicit integration schemes. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *92*, 105467. [[CrossRef](#)]
16. Rössler, O. An equation for continuous chaos. *Phys. Lett. A* **1976**, *57*, 397–398. [[CrossRef](#)]
17. Dadras, S.; Momeni, H.R. A novel three-dimensional autonomous chaotic system generating two, three and four-scroll attractors. *Phys. Lett. A* **2009**, *373*, 3637–3642. [[CrossRef](#)]
18. Nosé, S. A molecular dynamics method for simulations in the canonical ensemble. *Mol. Phys.* **1984**, *52*, 255–268. [[CrossRef](#)]
19. Hoover, W.G. Nonlinear conductivity and entropy in the two-body Boltzmann gas. *J. Stat. Phys.* **1986**, *42*, 587–600. [[CrossRef](#)]
20. Posch, H.A.; Hoover, W.; Vesely, F.J. Canonical dynamics of the Nosé oscillator: Stability, order, and chaos. *Phys. Rev. A* **1986**, *33*, 4253–4265. [[CrossRef](#)] [[PubMed](#)]
21. Van der Pol, B. On relaxation-oscillations. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1927**, *2*, 978–992. [[CrossRef](#)]
22. 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), Moscow, Russia, 12–14 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
23. Dormand, J.R.; Prince, P.J. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26. [[CrossRef](#)]