

Article

New Step Size Control Algorithm for Semi-Implicit Composition ODE Solvers

Petr Fedoseev ¹, Dmitriy Pesterev ¹, Artur Karimov ² and Denis Butusov ^{1,*}¹ Youth Research Institute, Saint Petersburg Electrotechnical University “LETI”, 197376 Saint Petersburg, Russia² Department of Computer-Aided Design, Saint Petersburg Electrotechnical University “LETI”, 197376 Saint Petersburg, Russia

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

Abstract: Composition is a powerful and simple approach for obtaining numerical integration methods of high accuracy order while preserving the geometric properties of a basic integrator. Adaptive step size control allows one to significantly increase the performance of numerical integration methods. However, there is a lack of efficient step size control algorithms for composition solvers due to some known difficulties in constructing a low-cost embedded local error estimator. In this paper, we propose a novel local error estimator based on a difference between the semi-implicit CD method and semi-explicit midpoint methods within a common composition scheme. We evaluate the performance of adaptive composition schemes with the proposed local error estimator, comparing it with the other state-of-the-art approaches. We show that composition ODE solvers with the proposed step size control algorithm possess higher numerical efficiency than known methods, by using a comprehensive set of nonlinear test problems.

Keywords: ODE; numerical integration; composition method; semi-implicit method; embedded integration; step size control



Citation: Fedoseev, P.; Pesterev, D.; Karimov, A.; Butusov, D. New Step Size Control Algorithm for Semi-Implicit Composition ODE Solvers. *Algorithms* **2022**, *15*, 275. <https://doi.org/10.3390/a15080275>

Academic Editor: Devendra Kumar

Received: 16 July 2022

Accepted: 1 August 2022

Published: 4 August 2022

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



Copyright: © 2022 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

One of the promising areas in numerical integration methods is composition and splitting methods. These solvers are known for their ability to preserve geometric properties of the basic method [1,2] and construct higher-order schemes for solving problems described by Hamiltonian ordinary and partial differential equations [3,4]. H. Yoshida reported some coefficients for constructing higher-order symplectic integrators by composition in [5]. M. Suzuki developed this approach further in [6], providing a generalized formula for so-called fractal composition methods. In [7], A. Murua and J. M. San-Serna developed an alternative technique for finding order conditions of composition schemes to improve the overall solver's performance. In [8], the authors show that semi-implicit basic methods allow one to efficiently solve non-Hamiltonian systems via composition schemes, and that the computational efficiency of semi-implicit and semi-explicit composition solvers can be close to the efficiency of the broadly used explicit Runge–Kutta methods.

It is known that adaptive step size control algorithms greatly increase the efficiency of ODE solvers due to the possibility to achieve better accuracy with fewer calculation steps. Applying variable-step integration techniques to composition methods is one of the key problems in the field. The main reason that prevents researchers from using the variable-step control in composition schemes is the lack of computationally cheap local truncation error estimators. Known approaches based on parallel solutions obtained via different methods are barely applicable due to significant overheads.

Recently, Blanes et al. showed the possibility to construct embedded error estimators for splitting and composition schemes [9] and proved the efficiency of the proposed approach when simulating Hamiltonian systems.

In this paper, we present a novel step size control algorithm for composition solvers that are based on simultaneous usage of semi-explicit and semi-implicit basic methods in a single composition scheme. The proposed approach can be used with an arbitrary composition scheme and requires only a single additional evaluation of the right-hand side (RHS) function per the method's stage, which makes it computationally efficient. In our study, we compare the developed adaptive step size technique with several known methods of step control using a representative set of test problems.

The rest of the paper is as follows. In Section 2, semi-implicit CD and semi-explicit midpoint methods are considered, and a new error estimator based on the difference between their solutions is proposed. In Section 3, several composition solvers with different error estimators are tested on three nonlinear test problems. Performance and step size behavior plots are used to evaluate the efficiency of the studied step control algorithms.

2. Materials and Methods

2.1. Using Variable-Step Integration with Composition Schemes

The main idea of the composition scheme [5,6] is to achieve an approximation of higher order by combining solutions obtained by a basic method Φ_h as follows.

Let us choose numbers $\gamma_1, \dots, \gamma_s$ satisfying the consistency condition

$$\gamma_1 + \gamma_2 + \dots + \gamma_s = 1. \quad (1)$$

Using Equation (1), we can obtain a composition method with s stages:

$$\Psi_h = \Phi_{\gamma_s h} \circ \dots \circ \Phi_{\gamma_2 h} \circ \Phi_{\gamma_1 h}. \quad (2)$$

Coefficients $\gamma_1, \dots, \gamma_s$ are chosen to increase the order of the overall method Ψ_h over the order of the basic method. They can be found using various techniques. First, consider a symmetric sequence proposed by H. Yoshida in 1990 [5] which has proven to be simple and efficient, and is widely known among scholars in the field. For a three-stage composition method, the coefficients are

$$\gamma_1 = \gamma_3 = \left(2 - 2^{\frac{1}{p+1}}\right)^{-1}, \gamma_2 = -2^{\frac{1}{p+1}} \left(2 - 2^{\frac{1}{p+1}}\right)^{-1}, \quad (3)$$

where p is the accuracy order of the basic method Φ_h .

Following the idea of [8], one can conclude that choosing the symmetric basic method of order 2 for the composition method with coefficients Equation (3) results in a three-stage method of order 4.

The other several considered sequences of coefficients are based on Kahan's [10] interpretation of Yoshida's scheme and are presented in Table 1. The notation $sN\text{ord}M$ is used following the original work and corresponds to a scheme with N stages which gives an approximation of accuracy order M .

Table 1. Coefficients for composition schemes of different orders of accuracy [10].

s1ord2	
γ_1	1
s5ord4	
$\gamma_1 = \gamma_2 = \gamma_4 = \gamma_5$	0.414490771794375737142354063
γ_3	−0.65796308717750294856941625
s7ord6	

Table 1. Cont.

$\gamma_1 = \gamma_7$	0.78451361047755726382
$\gamma_2 = \gamma_6$	0.23557321335935813368
$\gamma_3 = \gamma_5$	−1.1776799841788710069
γ_4	1.3151863206839112189
s17ord8	
$\gamma_1 = \gamma_{17}$	0.13020248308889008088
$\gamma_2 = \gamma_{16}$	0.56116298177510838456
$\gamma_3 = \gamma_{15}$	−0.38947496264484728641
$\gamma_4 = \gamma_{14}$	0.15884190655515560090
$\gamma_5 = \gamma_{13}$	−0.39590389413323757734
$\gamma_6 = \gamma_{12}$	0.18453964097831570709
$\gamma_7 = \gamma_{11}$	0.25837438768632204729
$\gamma_8 = \gamma_{10}$	0.29501172360931029887
γ_9	−0.60550853383003451170

Implementing the ODE solver with adaptive step size requires a formula for calculating a new step size value based on the previous one. The general formula for a new step size in a step control algorithm is as follows:

$$h_{n+1} = h_n R,$$

where R is a coefficient which contains information about computation error and can be obtained using different algorithms. In our study, we used the conventional step size control algorithm presented by E. Hairer et al. in [11], which states that after choosing an initial step size h_n , two approximations of the solution are to be found using two integration methods with different accuracy orders to evaluate the local truncation error. Let us refer to these approximations as $\mathbf{x}_n$ and $\tilde{\mathbf{x}}_n$. Using those approximations, one can find a new step size value h_{n+1} as follows:

$$h_{n+1} = h_n \left(\frac{Tol}{\mathbf{x}_n - \tilde{\mathbf{x}}_n} \right)^{\frac{1}{m+1}}, \quad (4)$$

where m corresponds to the lowest accuracy order of the schemes, and Tol is the desired error tolerance.

2.2. Semi-Implicit CD Method

The semi-implicit CD method [12–14], which is a generalization of the Störmer–Verlet method for non-Hamiltonian systems, was chosen as a basic symmetric method for this study. We will show how its semi-explicit and semi-implicit variants can be used to construct an efficient error estimator. The CD method is of accuracy order 2 and exists for ODE systems of dimension 2 and higher. Let us consider the following initial value problem (IVP):

$$\begin{cases} \frac{dx}{dt} = f(x, y, t) \\ \frac{dy}{dt} = g(x, y, t) \end{cases},$$

with initial conditions $x(t_0) = x_0$, $y(t_0) = y_0$. We denote the vector of state variables as $(x \ y)^\top$. Then, a symmetric semi-implicit second-order method [12] can be described as a composition of two adjoint methods:

$$\Phi(\mathbf{x}_{n+1}, h) = \Phi_1\left(\mathbf{x}_n, \frac{h}{2}\right) \circ \Phi_2\left(\mathbf{x}_{n+\frac{1}{2}}, \frac{h}{2}\right), \Phi(\mathbf{x}_n, h) = \Phi_2\left(\Phi_1\left(\mathbf{x}_n, \frac{h}{2}\right), \frac{h}{2}\right),$$

where

$$\Phi_1\left(\mathbf{x}_n, \frac{h}{2}\right) : \begin{cases} x_{n+0.5} = x_n + \frac{h}{2} f(x_n, y_n, t_n) \\ y_{n+0.5} = y_n + \frac{h}{2} g(x_{n+0.5}, y_n, t_n) \end{cases}, \quad (5)$$

$$\Phi_2\left(\mathbf{x}_{n+0.5}, \frac{h}{2}\right) : \begin{cases} y_{n+1} = y_{n+0.5} + \frac{h}{2}g(x_{n+0.5}, y_{n+1}, t_{n+1}) \\ x_{n+1} = x_{n+0.5} + \frac{h}{2}f(x_{n+1}, y_{n+1}, t_{n+1}) \end{cases}. \quad (6)$$

The obtained methods Equations (5) and (6) are adjoint, with Φ_1 being semi-explicit and Φ_2 semi-implicit. The implicitness can be resolved through the Newton iterations, simple iterations, or analytically (where possible).

2.3. Semi-Explicit Modification of Midpoint Method

Following the idea of the explicit midpoint method [14], we propose its semi-explicit modification for constructing an embedded error estimator. Considering the previously described system of ODE, one can obtain a solution at time t_{n+1} using the following formula, which is a second-order combination of explicit and semi-explicit methods of first accuracy order.

$$\tilde{\Phi}(\mathbf{x}_n, h) = \Phi_3\left(\mathbf{x}_n, \Phi_1\left(\mathbf{x}_n, \frac{h}{2}\right), h\right),$$

where

$$\Phi_1\left(\mathbf{x}_n, \frac{h}{2}\right) : \begin{cases} x_{n+0.5} = x_n + \frac{h}{2}f(x_n, y_n, t_n) \\ y_{n+0.5} = y_n + \frac{h}{2}g(x_{n+0.5}, y_n, t_n) \end{cases}, \quad (7)$$

$$\Phi_3(\mathbf{x}_n, \mathbf{x}_{n+0.5}, h) : \begin{cases} x_{n+1} = x_n + hf(x_{n+0.5}, y_{n+0.5}, t_{n+0.5}) \\ y_{n+1} = y_n + hg(x_{n+0.5}, y_{n+0.5}, t_{n+0.5}) \end{cases}, \quad (8)$$

which is a modification of the classic explicit midpoint scheme up to changing the first stage of the method from explicit to semi-explicit Euler integrator. Note that Equations (5) and (7) are similar; therefore, one does not need to compute $\mathbf{x}_{n+0.5}$ twice. Obtaining $\mathbf{x}_{n+1}$ and $\tilde{\mathbf{x}}_{n+1}$ allows us to use Equation (4) for step control.

2.4. Truncation Error Estimators for Composition ODE Solvers

In order to evaluate a new error estimation scheme, we compare it with several known adaptive step size algorithms. The first one estimates local error using the difference between two numerical solutions obtained by the same composition of different basic integration methods, which we denote as OCDM (*One Composition for Different Methods*). Figure 1 depicts the general scheme of the OCDM error estimator for an arbitrary composition solver.

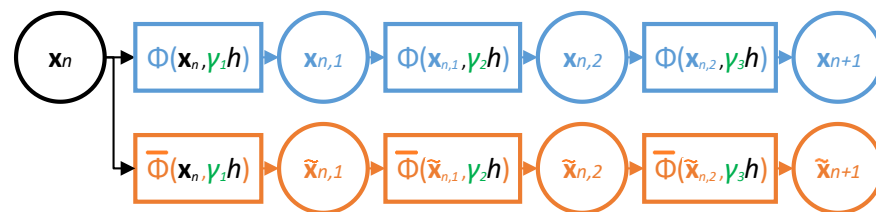


Figure 1. Scheme of OCDM algorithm. $\mathbf{x}_{n+1}$ is the main solution and $\tilde{\mathbf{x}}_{n+1}$ is the additional solution introduced for error estimation.

In this study, we will use two different *commutations* of the basic CD method. The term “commutation” means the order in which state variables are calculated. This way provides two different solutions as if we used different basic methods of the same accuracy order.

$$\Phi(\mathbf{x}_n, h) : \begin{cases} x_{n+0.5} = x_n + \frac{h}{2}f(x_n, y_n, t_n) \\ y_{n+0.5} = y_n + \frac{h}{2}g(x_{n+0.5}, y_n, t_n) \\ y_{n+1} = y_{n+0.5} + \frac{h}{2}g(x_{n+0.5}, y_{n+1}, t_{n+1}) \\ x_{n+1} = x_{n+0.5} + \frac{h}{2}f(x_{n+1}, y_{n+1}, t_{n+1}) \end{cases}$$

$$\bar{\Phi}(\mathbf{x}_n, h) : \begin{cases} y_{n+0.5} = y_n + \frac{h}{2}g(x_n, y_n, t_n) \\ x_{n+0.5} = x_n + \frac{h}{2}f(x_n, y_{n+0.5}, t_n) \\ x_{n+1} = x_{n+0.5} + \frac{h}{2}f(x_{n+1}, y_{n+0.5}, t_{n+1}) \\ y_{n+1} = y_{n+0.5} + \frac{h}{2}g(x_{n+1}, y_{n+1}, t_{n+1}) \end{cases}.$$

The second considered method of error estimation uses two solutions found using the same basic method in different composition solvers, which we call DCOM (*Different Compositions for One Method*). The DCOM scheme is shown in Figure 2.

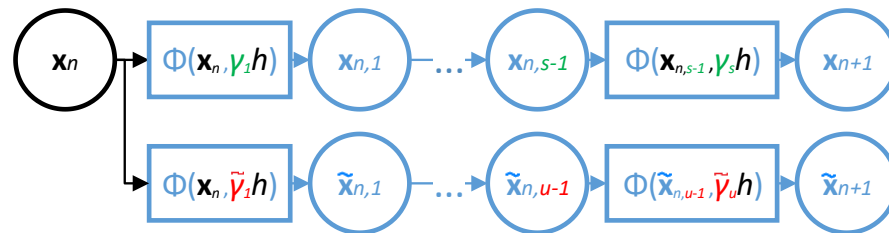


Figure 2. Scheme of DCOM algorithm. $\gamma_k, k = 1, 2, \dots, s$ and $\tilde{\gamma}_k, k = 1, 2, \dots, u$ are coefficients for different composition schemes with s and u stages, respectively.

2.5. Embedded Error Estimators in Composition Schemes

One way to evaluate a local truncation error is using pair of embedded methods, e.g., as in a well-known family of embedded Runge–Kutta schemes. Following the idea first described by Blanes et al. [9], one can use intermediate outputs obtained during the computation process of numerical approximation $x_{n+1} = \Psi_h(x_n) \approx x(t_{n+1}) = x(t_n + h)$ with Equation (2). Using a composition scheme with s stages, one obtains $s - 1$ intermediate outputs:

$$x_{n,0} = x_n, \\ x_{n,k} = \Psi_h(x_n) = \Phi_{\gamma_k h} \circ \dots \circ \Phi_{\gamma_1 h}, \quad k = 1, \dots, s - 1.$$

Approximation $\tilde{x}_{n+1}$ can be obtained as a linear combination of intermediate values $x_{n,k}$ and further used for error estimation:

$$\tilde{x}_{n+1} = \sum_{k=0}^{s-1} w_k x_{n,k}.$$

Coefficients w_k were found by solving a system of linear equations, which represent order conditions [9]. The number of order conditions to be satisfied increases according to the desired order of $\tilde{x}_{n+1}$ solution:

$$\begin{aligned} r_0 &\equiv w_0 + \sum_{k=1}^{s-1} w_k = 1, \\ r_1 &\equiv \sum_{k=1}^{s-1} w_k \sum_{j=1}^k \gamma_j = 1, \\ r_2 &\equiv \sum_{k=1}^{s-1} w_k \left(\sum_{j=1}^k \gamma_j \right)^2 = 1, \\ r_{3,1} &\equiv \sum_{k=1}^{s-1} w_k \left(\sum_{j=1}^k \gamma_j \right)^3 = 1, \\ r_{3,2} &\equiv \sum_{k=1}^{s-1} w_k \sum_{j=1}^k \gamma_j^3 = 1, \\ r_{4,1} &\equiv \sum_{k=1}^{s-1} w_k \left(\sum_{j=1}^k \gamma_j \right)^4 = 1, \\ r_{4,2} &\equiv \sum_{k=1}^{s-1} w_k \left(\sum_{j=1}^k \gamma_j^4 + 2 \sum_{j=1}^{k-1} \gamma_j^3 \sum_{l=j+1}^k \gamma_l \right) = 0, \\ r_{4,3} &\equiv \sum_{k=1}^{s-1} w_k \left(\sum_{j=1}^k \gamma_j^4 + 2 \sum_{j=2}^k \gamma_j^3 \sum_{l=1}^{j-1} \gamma_l \right) = 0. \end{aligned} \tag{9}$$

To obtain a solution of accuracy order 2 from the fourth-order scheme, one must solve five equations from (9). The lowest number of stages required to satisfy order conditions in

this case equals 5, and a unique solution for the particular scheme of fourth order by Kahan et al. [7] from Table 1 (s5ord4) will be as follows:

$$\tilde{x}_{n+1} = -x_n + w_1(x_{n,1} + x_{n,4}) + w_2(x_{n,2} + x_{n,3}),$$

with $w_1 = -1.40482876783862909$, $w_2 = 2.40482876783863197$.

To obtain a linear combination of order 4, one must solve the complete set of equations from (9). For the composition scheme with seven stages, a unique solution is as follows:

$$\tilde{x}_{n+1} = x_n + w_1(x_{n,1} - x_{n,6}) + w_2(x_{n,2} - x_{n,5}) + w_3(x_{n,3} - x_{n,4}).$$

Following the same idea, one can find the required w_i , $i = 1, 2, 3$ values for one of Kahan's schemes [10] with composition coefficients from Table 1 (s7ord6):

$$w_1 = -0.909832330075625028,$$

$$w_2 = 2.16331188722936796,$$

$$w_3 = 0.556955803872050015.$$

The proposed scheme can also be used with estimators of higher orders. In particular, one must solve a system of 13 linear equations to obtain an estimator of order 5 for a symmetric composition scheme with 17 stages of order 8 (s17ord8). The estimator in this case reads

$$\tilde{x}_{n+1} = -x_n + \sum_{i=1}^8 w_i(x_{n,i} - x_{n,17-i}),$$

where

$$w_1 = -2.77811433347582461058,$$

$$w_2 = 1.43336350604816157334,$$

$$w_3 = -2.35490307436226712937,$$

$$w_4 = 0.27249477875971647996,$$

$$w_5 = 3.09204406313073660493,$$

$$w_6 = 1.33511505989947708172,$$

$$w_7 = 0,$$

$$w_8 = 0.$$

Further, we will address this technique as Blanes' Error Estimation (BEE).

2.6. New Technique to Estimate Local Truncation Error

The main idea of the proposed embedded composition scheme implies the separation of the solution into two parts at every stage, as shown in Figure 3.

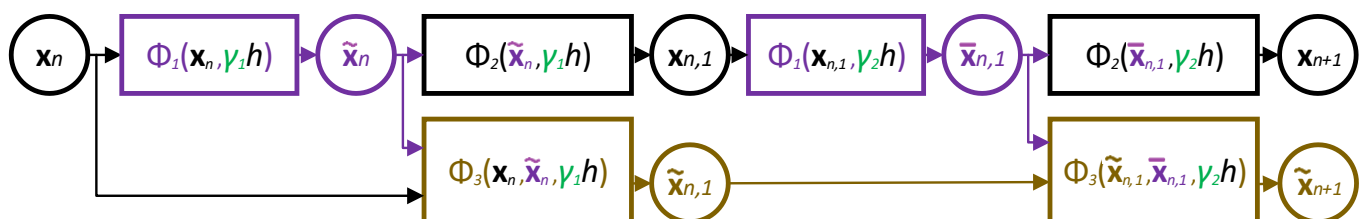


Figure 3. General scheme of proposed error estimation algorithm in 2-stage composition scheme.

In Figure 3, $x_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$, $x_{n,i} = \begin{pmatrix} x_{n,i} \\ y_{n,i} \end{pmatrix}$; Φ_1 is a semi-explicit D method; Φ_2 is a semi-implicit D method, which in composition with Φ_1 forms a semi-implicit CD method. To obtain a semi-explicit middle point method, one requires the composition of Φ_1 and Φ_3 . We will denote this scheme as ECDM (*Embedded Composition for Different Methods*).

Using formalization to describe the proposed technique, one can obtain the following:

$$\begin{aligned}\tilde{\mathbf{x}}_n &= \Phi_1(\mathbf{x}_n, \gamma_1 h), \\ \mathbf{x}_{n,1} &= \Phi_2(\tilde{\mathbf{x}}_n, \gamma_1 h), \\ \tilde{\mathbf{x}}_{n,1} &= \Phi_3(\mathbf{x}_n, \tilde{\mathbf{x}}_n, \gamma_1 h), \\ \bar{\mathbf{x}}_{n,1} &= \Phi_1(\mathbf{x}_{n,1}, \gamma_2 h), \\ \mathbf{x}_{n+1} &= \Phi_2(\bar{\mathbf{x}}_{n,1}, \gamma_2 h), \\ \tilde{\mathbf{x}}_{n+1} &= \Phi_3(\tilde{\mathbf{x}}_{n,1}, \bar{\mathbf{x}}_{n,1}, \gamma_2 h).\end{aligned}$$

One can see that the proposed approach is based mainly on the idea of using two different basic methods—the semi-implicit CD method and semi-explicit midpoint method, which have the same semi-implicit Euler as a starting substep. For every integration step, we take the solution at point $\mathbf{x}_n$ and use it as a starting value for two different compositions, which we address as $\mathbf{x}_n$ (main solution, semi-implicit CD) and $\tilde{\mathbf{x}}_n$ (additional solution, semi-explicit midpoint). Further analyzing the presented scheme, one can see that intermediate result $\mathbf{x}_{n,1}$ from the first part of the semi-implicit solution is being used as a part of explicit $\tilde{\mathbf{x}}_{n,2}$, and so on.

To further clarify the proposed solution, one can consider its pseudocode as Algorithm 1.

Algorithm 1. Pseudocode for the proposed step size control algorithm

```

Begin Algorithm
1  Set variable values  $\mathbf{x}_n, h$ ;
2  Set vector of composition coefficients  $\gamma$ ;
3  Set  $s$  as number of composition stages;
4  Begin DO-WHILE cycle until  $tol \geq ||\mathbf{x}_{n+1} - \tilde{\mathbf{x}}_{n+1}||$ 
5      Begin FOR  $i = 1, \dots, s$ 
6          Calculate a new value of  $h$  using formula  $h = h\gamma[i]$ ;
7          Calculate  $\mathbf{x}_{n+0.5}$  value starting from  $\mathbf{x}_n$  using semi-explicit  $D$  method with
              step size  $h/2$ ;
8          Calculate  $\tilde{\mathbf{x}}_{n+1}$  value starting from  $\mathbf{x}_n$  using explicit Euler method
              with step size  $h$  using previously obtained value  $\mathbf{x}_{n+0.5}$ ;
9          Calculate  $\mathbf{x}_{n+1}$  value starting from  $\mathbf{x}_n$  using semi-implicit  $D$  method with
              step size  $h/2$  using previously obtained value  $\mathbf{x}_{n+0.5}$ ;
10         End FOR cycle;
16         Choose new step value  $h$ ;
18     End DO-WHILE cycle;
End Algorithm

```

3. Results

In this section, we analyze the performance of the proposed approach in comparison with existing error estimators, by using a set of test ODE systems.

In each experiment, the following parameters were used: *Order*, which defines the composition scheme applied in the solver; initial conditions; simulation time T ; $h_{start}, h_{min}, h_{max}$ —initial, minimum, and maximum step size respectively; *Tol*—tolerance.

For plotting the performance plot, the tolerance value was changed in a specified interval (see Tables 2–4). In order to obtain an estimated solution error, we compared the result of the calculation with the reference solution taken by the extrapolation method of order 10 based on the linearly implicit midpoint method [15] over the same step sequences obtained from the evaluated solvers. Then, a maximum error over the simulation interval was taken for performance analysis. To obtain a reference solution, we used the same step sequences generated by the evaluated adaptive solvers but applying them to a high-order integration method.

Table 2. Parameter values used in simulation of the Rössler system.

Order	Initial Conditions	T	Step Size			Tolerance
			Min	Start	Max	
4	(1.6; 0; −0.1)	15	1×10^{-5}	5×10^{-3}	1	$1 \times 10^{-5} - 1 \times 10^{-9}$
6	(1.6; 0; −0.1)	15	1×10^{-5}	5×10^{-3}	1	$1 \times 10^{-7} - 1 \times 10^{-11}$
8	(1.6; 0; −0.1)	15	1×10^{-5}	5×10^{-3}	1	$1 \times 10^{-7} - 1 \times 10^{-11}$

Table 3. Dependence between the total number of steps and tolerance while solving Rössler problem.

Order	Tolerance	Total Number of Steps			
		OCDM	DCOM	BEE	ECDM
4	1×10^{-5}	86	662	186	71
	1×10^{-6}	134	1423	328	100
	1×10^{-7}	210	3063	581	151
	1×10^{-8}	330	6598	1031	231
	1×10^{-9}	522	14,213	1831	359
6	1×10^{-7}	146	191	307	138
	1×10^{-8}	199	301	484	177
	1×10^{-9}	273	475	766	219
	1×10^{-10}	377	751	1212	296
	1×10^{-11}	524	1188	1920	405
8	1×10^{-7}	61	118	116	55
	1×10^{-8}	68	158	168	67
	1×10^{-9}	84	217	244	83
	1×10^{-10}	107	299	357	104
	1×10^{-11}	137	414	522	132

Table 4. Parameter values used in simulation of Van der Pol oscillator.

Order	Initial Conditions	T	Step Size			Tolerance
			Min	Start	Max	
4	(1.15; 0)	15	1×10^{-5}	1×10^{-4}	1	$1 \times 10^{-5} - 1 \times 10^{-10}$
6	(1.15; 0)	15	1×10^{-5}	1×10^{-4}	1	$1 \times 10^{-5} - 1 \times 10^{-10}$
8	(1.52; 0)	15	1×10^{-5}	1×10^{-4}	1	$1 \times 10^{-4} - 1 \times 10^{-9}$

3.1. Problem 1: Rössler System

The Rössler system is well-known as being one of the simplest ODE systems with chaotic behavior and a fine test problem for numerical ODE solvers [16]. In this paragraph, all simulations were carried out with parameters $a = 0.2$, $b = 0.2$, $c = 5.7$, which correspond to a chaotic regime.

$$\begin{aligned}\dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c)\end{aligned}$$

A semi-implicit CD algorithm (7) for the Rössler problem can be formulated as follows:

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

A semi-explicit midpoint algorithm (8) for the Rössler system is as follows:

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

Table 2 presents parameter values used in our experiments. Table 3 provides data on total step evaluations for every considered algorithm. We tested three methods, of orders 4, 6, and 8. A double-precision floating-point data type was used. To compare the computational cost versus obtained error ratio for investigated step size control algorithms, including OCDM, DCOM, Blanes' embedded estimator (BEE), and the proposed technique, we present performance plots for them in Figure 4. Figure 5 illustrates the step size behavior for various local truncation error estimators. As one can see, while increasing the accuracy order of the desired solution, we simultaneously increase the number of composition stages, which leads to a significant decrease in calculation speed per step, but at the same time, it increases the overall performance due to larger step sizes.

One can see from Figures 4 and 5 that the proposed step size control technique shows a noticeable increase in speed and accuracy for all investigated schemes. One can see that OCDM and the proposed scheme result in almost similar step size curve amplitude while DCOM and BEE generate sufficiently smaller steps. For this problem, ease of implementation, and larger step sizes, provide the most efficient step behavior.

3.2. Problem 2: Van der Pol Oscillator

The Van der Pol oscillator is a system with non-linear damping controlled by parameter m and corresponding to the stiffness of the system [17]. The value $m = 55$ corresponds to moderately stiff behavior.

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= m(1 - x^2)y - x \end{aligned}$$

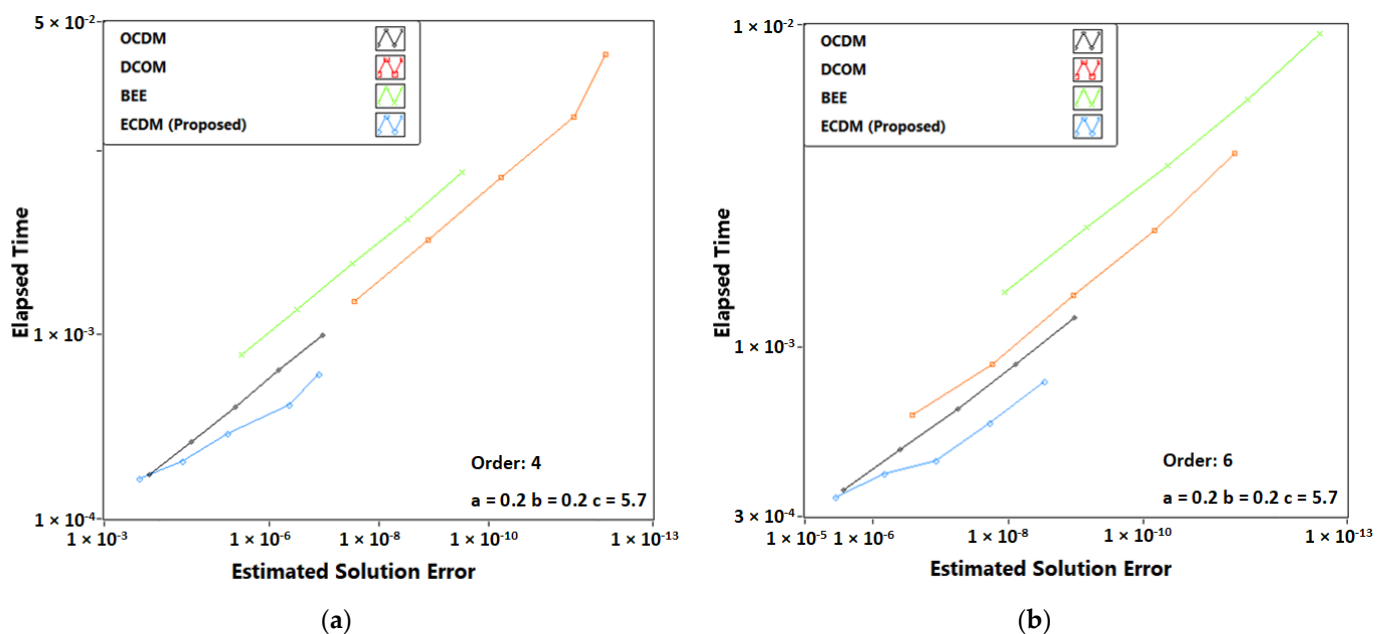


Figure 4. Cont.

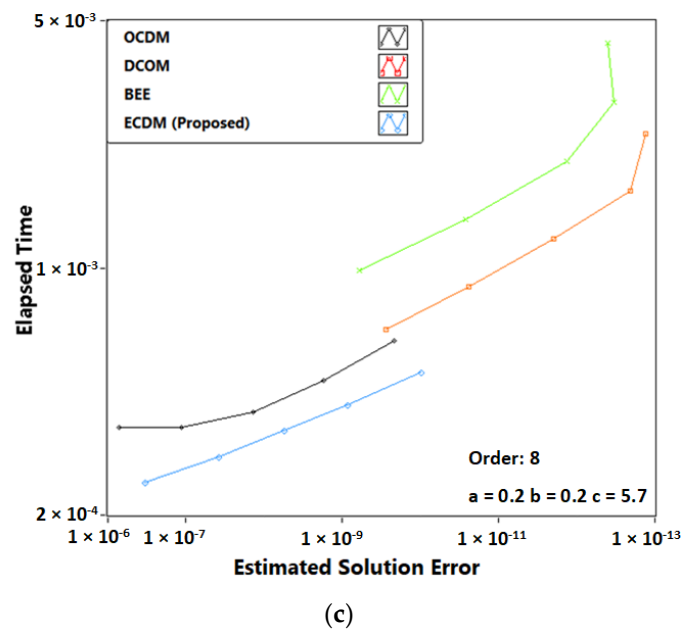


Figure 4. Performance plots for Rössler system simulation using composition solvers of orders 4 (a), 6 (b), and 8 (c).

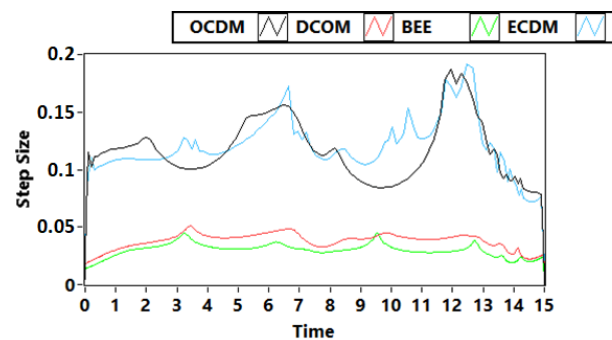


Figure 5. Step size behavior for different controllers under investigation while simulating Rössler system.

A semi-implicit CD algorithm for the Van der Pol oscillator can be written as follows:

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

A semi-explicit midpoint algorithm for the Van der Pol equation is as follows:

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

Figure 6 shows the performance plots for solving the Van der Pol equation through different adaptive step size composition solvers. Figure 7 illustrates the step size behavior for various local truncation error estimators. Parameter values used in experiments can be found in Table 4. Gathered data on total number of steps for every considered algorithm is presented in Table 5.

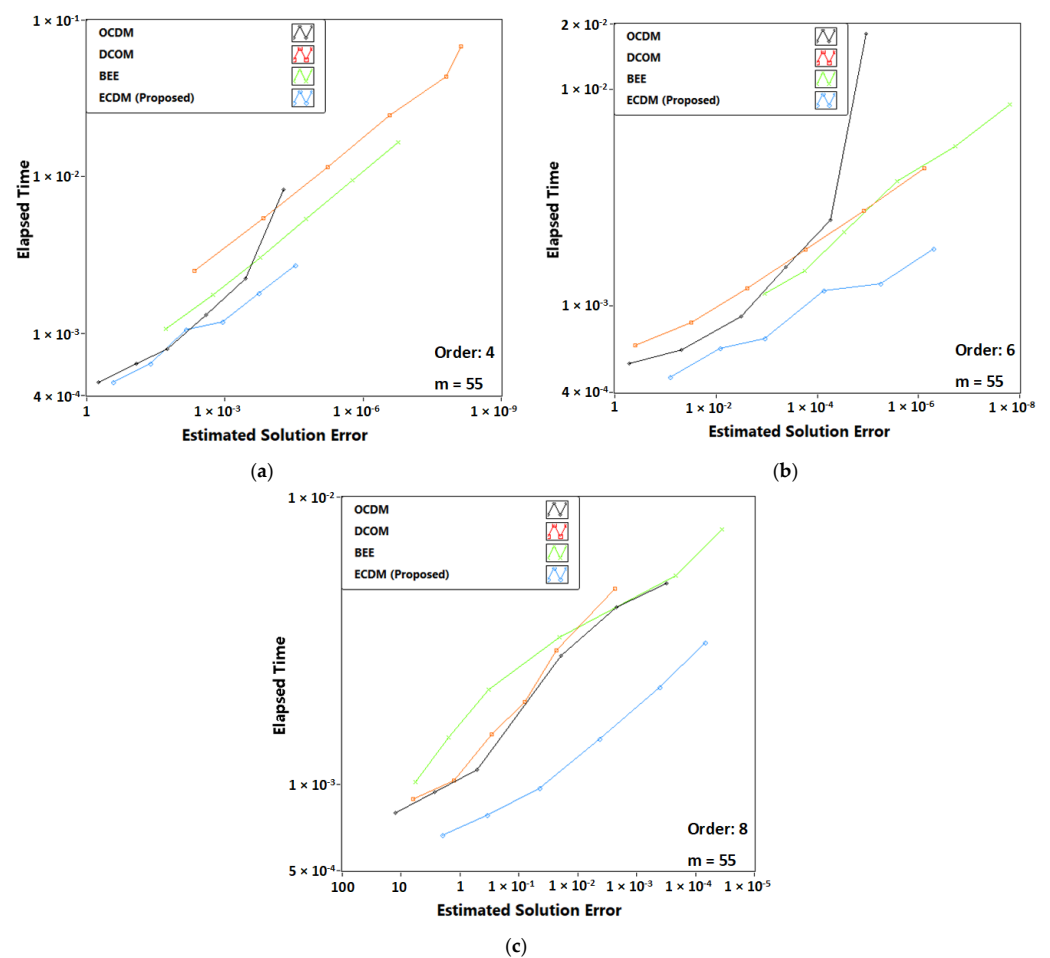


Figure 6. Performance plots for Van der Pol oscillator simulation by solvers of orders 4 (a), 6 (b), and 8 (c).

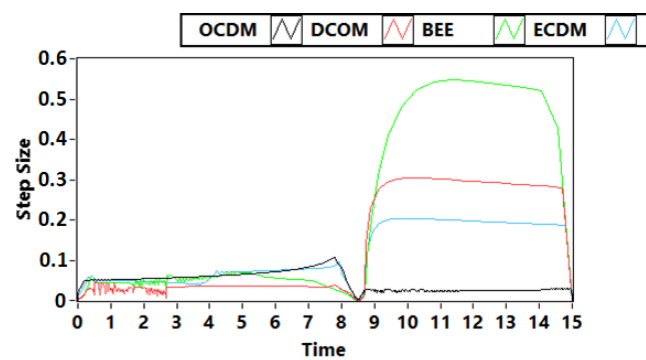


Figure 7. Graphs of step size behavior for different controllers under investigation for Van der Pol oscillator.

Table 5. Dependence between the total number of steps and tolerance for Van der Pol system.

Order	Tolerance	Total Number of Steps			
		OCDM	DCOM	BEE	ECDM
4	1×10^{-5}	115	823	232	133
	1×10^{-6}	162	1760	395	187
	1×10^{-7}	258	3790	687	376
	1×10^{-8}	442	8039	1206	445
	1×10^{-9}	777	14,159	2133	698
	1×10^{-10}	1748	22,080	3795	1009

Table 5. Cont.

Order	Tolerance	Total Number of Steps			
		OCDM	DCOM	BEE	ECDM
6	1×10^{-5}	95	119	177	106
	1×10^{-6}	136	169	265	145
	1×10^{-7}	218	253	404	198
	1×10^{-8}	372	395	635	282
	1×10^{-9}	674	624	1005	414
	1×10^{-10}	2164	991	1595	621
8	1×10^{-4}	56	90	80	65
	1×10^{-5}	67	117	113	82
	1×10^{-6}	92	166	169	118
	1×10^{-7}	237	245	266	170
	1×10^{-8}	306	355	438	253
	1×10^{-9}	377	560	602	336

. The performance plots shown in Figure 6 show that all investigated methods of error estimation perform worse for solvers of higher accuracy order, while the proposed technique is efficient for both low and high accuracy orders. Appending the previous results, we may conclude that reducing the additional costs required for the local error estimation is a key reason for the proposed ECDM step size control algorithm's high performance

3.3. Problem 3: Two-Body Problem

The third test problem is a Hamiltonian system of fourth order that describes the motion of two bodies with similar masses [18]:

$$\begin{aligned}\frac{dx}{dt} &= z, \\ \frac{dy}{dt} &= w, \\ \frac{dz}{dt} &= -\frac{x}{(\sqrt{x^2+y^2})^3}, \\ \frac{dw}{dt} &= -\frac{y}{(\sqrt{x^2+y^2})^3}.\end{aligned}$$

The analytical solution for the initial values $x(0) = 1$, $y(0) = 0$, $z(0) = 0$, $w(0) = 1$ is as follows:

$$\begin{aligned}x(t) &= \cos(t), \\ y(t) &= \sin(t), \\ z(t) &= -\sin(t), \\ w(t) &= \cos(t).\end{aligned}$$

A semi-explicit midpoint algorithm for the system is as follows:

$$\begin{aligned}w^{n+0.5} &= w^n + \frac{h}{2} \left(-\frac{y^n}{(\sqrt{x^n x^n + y^n y^n})^3} \right), \\ z^{n+0.5} &= z^n + \frac{h}{2} \left(-\frac{x^n}{(\sqrt{x^n x^n + y^n y^n})^3} \right), \\ y^{n+0.5} &= y^n + \frac{h}{2} (w^{n+0.5}), \\ x^{n+0.5} &= x^n + \frac{h}{2} (z^{n+0.5}), \\ x^{n+1} &= x^n + h (z^{n+0.5}), \\ y^{n+1} &= y^n + h (w^{n+0.5}), \\ z^{n+1} &= z^n + h \left(-\frac{x^{n+0.5}}{(\sqrt{x^{n+0.5} x^{n+0.5} + y^{n+0.5} y^{n+0.5}})^3} \right), \\ w^{n+1} &= w^n + h \left(-\frac{y^{n+0.5}}{(\sqrt{x^{n+0.5} x^{n+0.5} + y^{n+0.5} y^{n+0.5}})^3} \right).\end{aligned}$$

A semi-implicit CD algorithm is as follows:

$$\begin{aligned} w^{n+0.5} &= w^n + \frac{h}{2} \left(-\frac{y^n}{(\sqrt{x^n x^n + y^n y^n})^3} \right), \\ z^{n+0.5} &= z^n + \frac{h}{2} \left(-\frac{x^n}{(\sqrt{x^n x^n + y^n y^n})^3} \right), \\ y^{n+0.5} &= y^n + \frac{h}{2} (w^{n+0.5}), \\ x^{n+0.5} &= x^n + \frac{h}{2} (z^{n+0.5}), \\ x^{n+1} &= x^{n+0.5} + h (z^{n+0.5}), \\ y^{n+1} &= y^{n+0.5} + h (w^{n+0.5}), \\ z^{n+1} &= z^{n+0.5} + h \left(-\frac{x^{n+1}}{(\sqrt{x^{n+1} x^{n+1} + y^{n+1} y^{n+1}})^3} \right), \\ w^{n+1} &= w^{n+0.5} + h \left(-\frac{y^{n+1}}{(\sqrt{x^{n+1} x^{n+1} + y^{n+1} y^{n+1}})^3} \right). \end{aligned}$$

Figure 8 shows the performance plots for solving the two-body problem through different adaptive step size composition solvers. Parameter values used in the experiments are presented in Table 6. Dependence between the total number of steps and tolerance can be observed in Table 7.

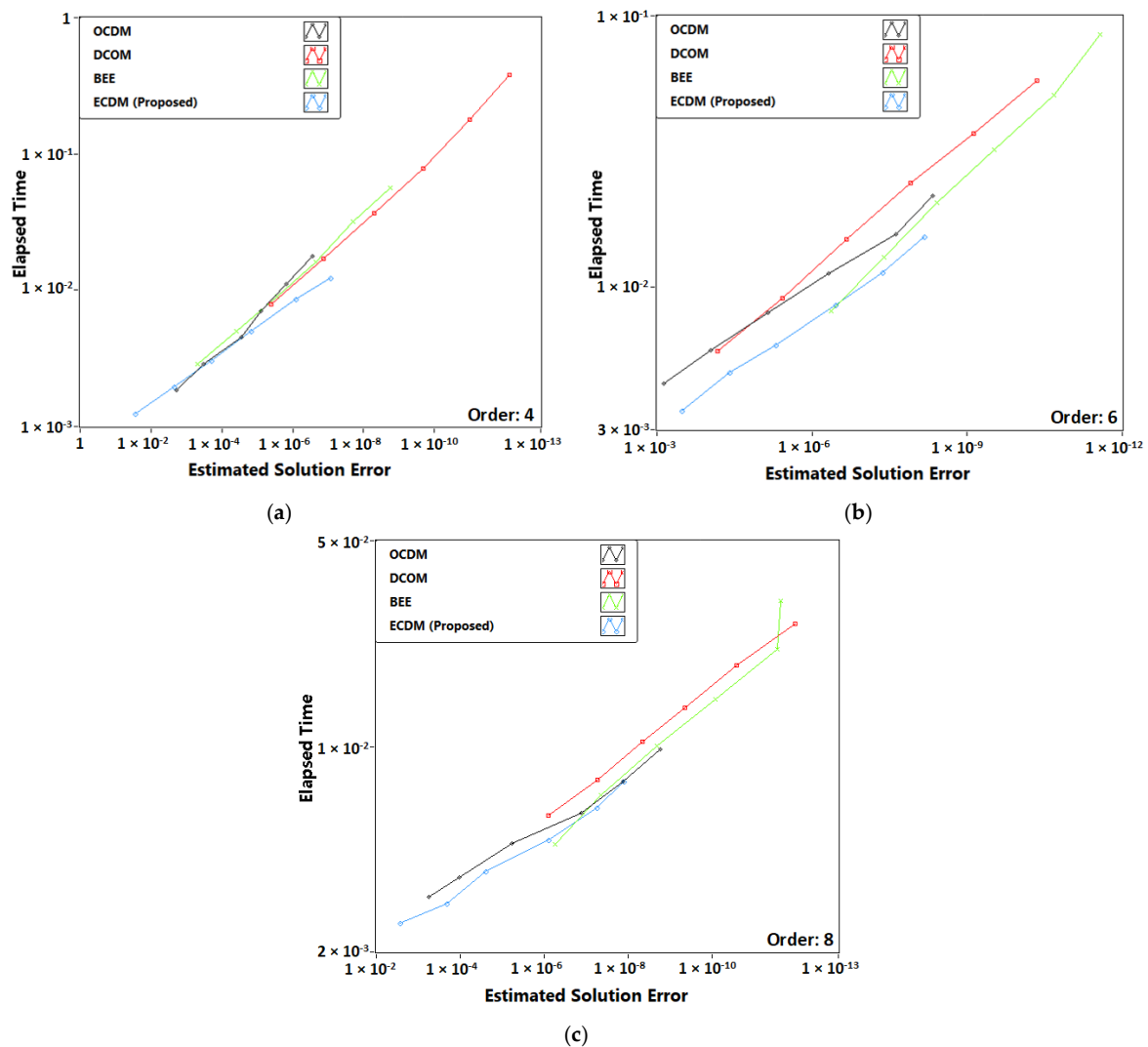


Figure 8. Performance plots for two-body problem simulation by solvers of orders 4 (a), 6 (b), and 8 (c).

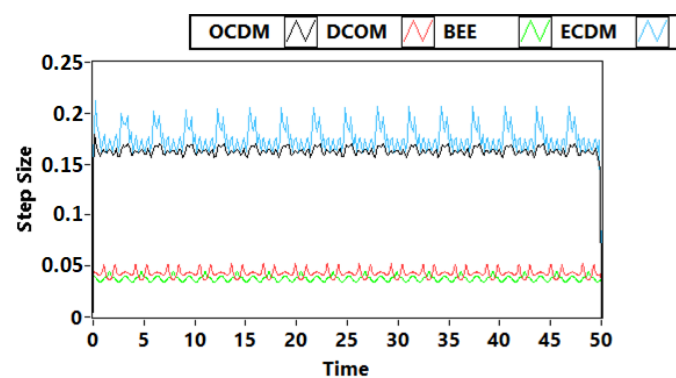
Table 6. Parameter values used in the simulation of two-body problem.

Order	Initial Conditions	T	Step Size			Tolerance
			Min	Start	Max	
4	(1; 0; 0; 1)	50	1×10^{-4}	5×10^{-3}	1	1×10^{-4} – 1×10^{-9}
6	(1; 0; 0; 1)	50	1×10^{-4}	5×10^{-3}	1	1×10^{-6} – 1×10^{-11}
8	(1; 0; 0; 1)	50	1×10^{-4}	5×10^{-3}	1	1×10^{-5} – 1×10^{-10}

Table 7. Dependence between the total number of steps and tolerance while solving two-body problem.

Order	Tolerance	Total Number of Steps			
		OCDM	DCOM	BEE	ECDM
4	1×10^{-4}	177	986	335	156
	1×10^{-5}	279	2121	593	246
	1×10^{-6}	441	4567	1053	389
	1×10^{-7}	697	9837	1871	615
	1×10^{-8}	1102	21,192	3324	972
	1×10^{-9}	1744	45,655	5910	1539
6	1×10^{-6}	309	442	745	289
	1×10^{-7}	427	699	1179	401
	1×10^{-8}	593	1106	1868	547
	1×10^{-9}	823	1751	2959	759
	1×10^{-10}	1143	2774	4689	1048
	1×10^{-11}	1587	4395	7430	1452
8	1×10^{-5}	88	234	197	83
	1×10^{-6}	113	322	288	106
	1×10^{-7}	146	444	421	136
	1×10^{-8}	187	611	616	175
	1×10^{-9}	241	849	904	226
	1×10^{-10}	310	1179	1325	290

Figure 9 illustrates the step size behavior for various local truncation error estimators.

**Figure 9.** Graphs of step size behavior for different controllers under investigation for two-body problem.

While solving the system with a periodic solution, the proposed ECDCM step size control algorithm shows higher performance for low accuracy order composition schemes, while its overall CPU time vs. error features remains the most competitive among all investigated step size control algorithms. The step size plot given in Figure 9 correlates with the periodic behavior of the state variables. The ECDCM selects the largest step sizes, which indirectly indicates that the other local error estimation techniques tend to overestimate the error and select fewer step sizes, which makes them less efficient.

4. Conclusions

The reported study is dedicated to variable step size control algorithms for composition integration methods. The general form of composition methods was considered, and the problem of constructing an efficient local error estimator was outlined. Four algorithms for automatic step size control were compared, including a novel technique that combines a semi-explicit modification of the midpoint method with two adjoint D methods in a composition scheme. This allows the construction of an efficient local error estimator. Performance tests on nonlinear systems including the Rössler system, the Van der Pol system, and the two-body problem have shown that the new approach has the greatest CPU time to truncation error ratio than other known error estimation techniques for composition schemes described in the literature. The investigated techniques included using one composition scheme for different methods, different composition schemes for similar methods, and (as recently reported) Blanes' embedded local error estimator. The experimental part of the study confirmed that the proposed step size control algorithm for semi-implicit composition ODE solvers outperforms the known solutions. The obtained results can be used to develop simulation software based on composition ODE solvers, with increased performance. Our further studies will be dedicated to developing a general theory of embedded composition methods based on the proposed method and finding other efficient and versatile local error estimation algorithms.

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

Funding: This study was supported by the grant of the Russian Science Foundation (RSF), project 22-19-00573.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Sofroniou, M.; Spaletta, G. Derivation of symmetric composition constants for symmetric integrators. *Optim. Methods Softw.* **2005**, *20*, 597–613. [\[CrossRef\]](#)
2. Haier, E.; Lubich, C.; Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*; Springer: Berlin/Heidelberg, Germany, 2006.
3. Leimkuhler, B.; Reich, S. *Simulating Hamiltonian Dynamics* (No. 14); Cambridge University Press: Cambridge, MA, USA, 2004.
4. Roulet, J.; Choi, S.; Vaniček, J. Efficient geometric integrators for nonadiabatic quantum dynamics. II. The diabatic representation. *J. Chem. Phys.* **2019**, *150*, 204113. [\[CrossRef\]](#)
5. Yoshida, H. Construction of higher order symplectic integrators. *Phys. Lett. A* **1990**, *150*, 262–268. [\[CrossRef\]](#)
6. Suzuki, M. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Phys. Lett. A* **1990**, *146*, 319–323. [\[CrossRef\]](#)
7. Murua, A.; Sanz-Serna, J.M. Order conditions for numerical integrators obtained by composing simpler integrators. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **1999**, *357*, 1079–1100. [\[CrossRef\]](#)
8. Butusov, D.N.; Ostrovskii, V.Y.; Karimov, A.I.; Andreev, V.S. Semi-explicit composition methods in memcapacitor circuit simulation. *IJERTCS* **2019**, *10*, 37–52. [\[CrossRef\]](#)
9. Blanes, S.; Casas, F.; Thalhammer, M. Splitting and composition methods with embedded error estimators. *Appl. Numer. Math.* **2019**, *146*, 400–415. [\[CrossRef\]](#)
10. Kahan, W.; Li, R.C. Composition constants for raising the orders of unconventional schemes for ordinary differential equations. *Math. Comput.* **1997**, *66*, 1089–1099. [\[CrossRef\]](#)
11. Hairer, E.; Wanner, G.; Nørsett, S.P. *Solving Ordinary Differential Equations I. Nonstiff Problems*; Springer: Berlin/Heidelberg, Germany, 1987.
12. Butusov, D. Adaptive Stepsize Control for Extrapolation Semi-Implicit Multistep ODE Solvers. *Mathematics* **2021**, *9*, 950. [\[CrossRef\]](#)
13. Butusov, D.; Tutueva, A.; Fedoseev, P.; Terentev, A.; Karimov, A. Semi-implicit multistep extrapolation ODE solvers. *Mathematics* **2020**, *8*, 943. [\[CrossRef\]](#)
14. Süli, E.; David, F.M. *An Introduction to Numerical Analysis*; Cambridge University Press: Cambridge, MA, USA, 2003.

15. Hairer, E.; Lubich, C. Extrapolation at stiff differential equations. *Numer. Math.* **1988**, *52*, 377–400. [[CrossRef](#)]
16. Rössler, O. An equation for continuous chaos. *Phys. Lett. A* **1976**, *57*, 397–398. [[CrossRef](#)]
17. Van der Pol, B. «On relaxation-oscillations», The London, Edinburgh and Dublin Phil. Mag. J. Sci. **1927**, *2*, 978–992. [[CrossRef](#)]
18. Moo, K.W.; Senu, N.; Ismail, F.; Suleiman, M. New phase-fitted and amplification-fitted fourth-order and fifth-order Runge-Kutta-Nyström methods for oscillatory problems. *Abstr. Appl. Anal.* **2013**, *2013*, 939367. [[CrossRef](#)]