See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/355301116

Stability Analysis and Optimization of Semi-Explicit Predictor-Corrector Methods

Article in Mathematics · October 2021

DOI: 10.3390/math9192463

citations 5		READS 876	READS 876		
2 author	S:				
	Aleksandra V. Tutueva 60 PUBLICATIONS 947 CITATIONS SEE PROFILE		Denis Butusov Saint Petersburg Electrotechnical University "LETI" 177 PUBLICATIONS 1,885 CITATIONS SEE PROFILE		





Article Stability Analysis and Optimization of Semi-Explicit Predictor–Corrector Methods

Aleksandra Tutueva ^{1,*} and Denis Butusov ²

- ¹ Department of Computer-Aided Design, Saint Petersburg Electrotechnical University "LETI", 197376 Saint Petersburg, Russia
- ² Youth Research Institute, Saint Petersburg Electrotechnical University "LETI", 197376 Saint Petersburg, Russia; dnbutusov@etu.ru
- * Correspondence: avtutueva@etu.ru; Tel.: +7-952-399-60-96

Abstract: The increasing complexity of advanced devices and systems increases the scale of mathematical models used in computer simulations. Multiparametric analysis and study on long-term time intervals of large-scale systems are computationally expensive. Therefore, efficient numerical methods are required to reduce time costs. Recently, semi-explicit and semi-implicit Adams–Bashforth– Moulton methods have been proposed, showing great computational efficiency in low-dimensional systems simulation. In this study, we examine the numerical stability of these methods by plotting stability regions. We explicitly show that semi-explicit methods possess higher numerical stability than the conventional predictor–corrector algorithms. The second contribution of the reported research is a novel algorithm to generate an optimized finite-difference scheme of semi-explicit and semi-implicit Adams–Bashforth–Moulton methods without redundant computation of predicted values that are not used for correction. The experimental part of the study includes the numerical simulation of the three-body problem and a network of coupled oscillators with a fixed and variable integration step and finely confirms the theoretical findings.

check for **updates**

Citation: Tutueva, A.; Butusov, D. Stability Analysis and Optimization of Semi-Explicit Predictor–Corrector Methods. *Mathematics* **2021**, *9*, 2463. https://doi.org/10.3390/math9192463

Academic Editors: Raimondas Ciegis and Hongyu Liu

Received: 5 August 2021 Accepted: 1 October 2021 Published: 3 October 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). **Keywords:** semi-explicit integration; predictor–corrector method; semi-explicit Adams–Bashforth– Moulton method; three-body problem; ODE solver

1. Introduction

Modern industrial enterprises need to reduce the time and cost of designing and developing new products to maintain a competitive edge. However, while the devices under design become more and more complex, the list of requirements for their characteristics is continuously expanding. One prospective way to solve this contradiction is to use computer simulation, including the initial design stages. It is known that design decisions at the beginning of the project route determine up to 80% of the financial costs of product development [1]. It is essential to increase the computational efficiency of simulation tools to develop multi-component objects described by large-scale and super large-scale mathematical models [2]. Speaking of simpler models, multiple calculations with different parameter values, corresponding to various geometric and physical properties of the device and its operation scenarios, are performed for such systems [3]. Moreover, long-term simulations are often required to detect nonlinear effects and investigate the properties of models that appear over time [4]. Therefore, it is necessary to implement parallel calculations while maintaining accuracy [5–7].

Systems of ordinary differential equations (ODE) are among the most common mathematical models of technical devices or natural phenomena. The common approach to solving ODE systems in simulation environments such as Matlab, LabVIEW, and Wolfram Mathematica is numerical integration [8–10]. Regardless of the ODE system properties, e.g., stiffness, scale, delay, etc., one can often apply the same integration method. However,

this approach can entail high computational costs in long-term simulations or multiparametric research of large-scale models [11]. One of the most common solutions here is linear multistep methods. Multistep algorithms can demonstrate a high speed of computations due to a single call to the right-hand side function at the integration step. Implicit multistep methods, such as the Adams-Moulton method or backward differentiation formula, have significantly larger stability regions than explicit Adams-Bashforth methods. However, the price for this achievement is the computationally expensive embedded Newton's method [12]. Thus, implicit multistep methods are more time-consuming when solving high-dimensional systems, while explicit multistep integration has low suitability for simulating stiff systems [13]. One of the compromise solutions is a predictor–corrector method. This integration technique performs the initial approximation by the Adams–Bashforth method and uses the Adams-Moulton integration to correct the obtained values. Since the stability of such methods is higher than that of the Adams–Bashforth methods, while all calculations are performed explicitly, this technique is prospective for large-scale system simulation [14–17]. However, it is necessary for ubiquitous integration of such methods into software packages to increase their stability to expand the application to mildly stiff differential equations.

Recently, semi-explicit and semi-implicit modifications of the Adams–Bashforth– Moulton (ABM) methods for non-Hamiltonian systems have been proposed [18]. It was experientially shown that such integration is computationally efficient for non-stiff lowdimensional systems simulation. However, no estimation of numerical stability of semiexplicit ABM methods was given due to the complexity of the evaluation procedure for methods, which does not exist for systems of order 2 and lower. In this paper, we perform the theoretical assessment of the stability of semi-explicit Adams–Bashforth–Moulton methods. In addition, we propose an algorithm for analyzing the feedback matrix of the simulated system, which makes it possible to reduce the computational costs of the prediction stage.

The rest of the paper is organized as follows. Section 2 describes the studied semiexplicit Adams–Bashforth–Moulton methods and analyzes the stability regions. Furthermore, an algorithm for synthesizing a minimal predictor–corrector finite-difference scheme is given, and an illustrative example is examined. Section 3 presents the results of the simulation of the three-body problem and a coupled oscillator network. We explicitly demonstrate a higher computational efficiency of solvers based on semi-explicit methods compared to known multistep schemes. Finally, some conclusions and discussions are given in Section 4.

2. Materials and Methods

Let us consider semi-explicit and semi-implicit modifications of the predictor-corrector Adams–Bashforth–Moulton (ABM) formula. In the original ABM integration, all calculations are explicit. Furthermore, the stability of such an integration technique is higher than that of the explicit Adams–Bashforth method [19].

2.1. Semi-Explicit and Semi-Implicit Adams–Bashforth Integration Formula

Semi-explicit and semi-implicit multistep ABM methods can be applied to systems of ODEs of dimension two and higher [18]. Let us consider the following sample system:

$$\begin{cases} \dot{x} = f(x, y, t) \\ \dot{y} = g(x, y, y) \end{cases}$$

where f() and g() are the right-hand side functions.

The prediction stage in the semi-explicit ABM method is calculated by the Adams– Bashforth method as in the original ABM technique. However, to correct the approximate values, we will use not only the values from the prediction stage, but also the state variables calculated at the same step by the Adams–Moulton method [18]:

$$\begin{aligned} x_{n+1} &= x_n + hb_0 f\left(x_{n+1}^p, y_{n+1}^p, t_{n+1}\right) + h\sum_{i=0}^{k-1} b_{i+1} f(x_{n-i}, y_{n-i}, t_{n-i}) \\ y_{n+1} &= y_n + hb_0 g\left(x_{n+1}, y_{n+1}^p, t_{n+1}\right) + h\sum_{i=0}^{k-1} b_{i+1} g(x_{n-i}, y_{n-i}, t_{n-i}) \end{aligned}$$
(1)

where b_i are coefficients of the implicit Adams–Moulton method, k is the number of stages, h is the integration step, and t is the time moment. The superscript p marks the values that were approximated at the prediction stage.

The obtained integration technique is called the semi-explicit Adams–Bashforth– Moulton (SEABM) method. The formula for the semi-implicit Adams–Bashforth–Moulton (SIABM) method can be obtained as a modification of Equation (1), where for each state variable there is one implicit calculation that corresponds to the equation being solved:

$$x_{n+1} = x_n + hb_0 f\left(x_{n+1}, y_{n+1}^p, t_{n+1}\right) + h\sum_{i=0}^{k-1} b_{i+1} f\left(x_{n-i}, y_{n-i}, t_{n-i}\right) y_{n+1} = y_n + hb_0 g\left(x_{n+1}, y_{n+1}, t_{n+1}\right) + h\sum_{i=0}^{k-1} b_{i+1} g\left(x_{n-i}, y_{n-i}, t_{n-i}\right)$$
(2)

In dynamical systems with a nonzero main diagonal of the feedback matrix, values of x_{n+1} and y_{n+1} in (2) can be calculated by iterative methods, including Newton's method or the simple iteration approach. For Hamiltonian problems, the finite-difference schemes of methods (1) and (2) coincide.

The computational costs of the proposed modified Adams–Bashforth–Moulton schemes and the original ABM method are equal. The key hypothesis of our study is that semiexplicit computations will increase the numerical stability of the finite-difference model, leading to a decrease in integration costs during variable-step simulations. Moreover, the corrector part does not require the approximation of all state variables by the Adams– Bashforth method. Therefore, the finite-difference scheme of the prediction stage semiexplicit and semi-implicit modification can contain less algebraic calculations than the original predictor–corrector method. These aspects are further discussed below.

2.2. Stability Analysis

Since the investigated semi-explicit and semi-implicit multistep methods do not exist for first-order ODE systems, Dahlquist's test equation is not suitable for studying their stability. Several new approaches to study numerical stability have recently been proposed [11,20–22]. Following the ideas described in [11,22], we chose the two-dimensional problem to plot stability regions.

Let *B* be the vector of coefficients of the Adams–Bashforth method, *M* is the vector of coefficients of the Adams–Moulton method, and *p* denotes the lengths of *B*, and *M*. Let us compose a matrix *A* of dimension 2×2 for the test problem

x

$$=Ax,$$
(3)

with the eigenvalues equal to $\lambda_{1,2} = \sigma \pm j\omega$. Then we compose the matrix polynomial p(z), where z is the delay operator, as follows:

$$p(z) = (I + B_1 h A) z^{p-1} + B_2 h A z^{p-2} + \dots$$

where *I* is the identity matrix, and *h* is the integration step. The calculation of the matrix coefficients increment at x_1 at the second step of the semi-explicit method is carried out as

$$P_{11}(z) = (I + M_2 h A_{11}) z^{p-1} + M_3 h A_{11} z^{p-2} + \dots$$
$$P_{12}(z) = M_1 (A_{11} p_{12} + A_{12} p_{22}) + (I + M_2 h A_{11}) z^{p-1} + M_3 h A_{11} z^{p-2} + \dots$$

Similarly, the coefficients of the increment polynomial at x_2 are found from

$$P_{21}(z) = M_1(A_{21}p_{11} + A_{12}p_{21}) + (I + M_2hA_{11})z^{p-1} + M_3hA_{11}z^{p-2} + \dots$$

$$P_{22}(z) = (I + M_2hA_{11})z^{p-1} + M_3hA_{11}z^{p-2} + \dots$$

The scalar characteristic polynomial of the matrix P(z) is found from

$$h(z) = ((I - M_1 A_{11})z^p + P_{11}(z))((I - M_1 A_{22})z^p + P_{22}(z)) - P_{21}(z)P_{12}(z).$$

The root of h(z) with the largest modulus gives the value of the stability function $R(h\lambda)$. The stability polynomial for the SIABM method is constructed similarly.

Let $k = A_{11}/A_{22}$ be the ratio of the matrix elements of the test problem (3) on the main diagonal. Let us call k the symmetry coefficient. For a matrix in Jordan normal form, k = 1. For a matrix in Frobenius normal form, the coefficient k can be equal to 0 or ∞ . These are two extreme cases, which give limits for stability functions. Therefore, we can consider only these cases. The calculation of the coefficients of matrix A is carried out according to the following formulas:

$$A_{22} = \frac{2\sigma}{1+k}, A_{21} = -\sqrt{\lambda^2 - (1+k)A_{22}\lambda + kA_{22}^2}, A_{12} = A_{21}, A_{11} = kA_{22}.$$

The stability regions of SEABM and SIABM methods obtained for k = 0 and k = 1 are shown in Figures 1–4. Comparing them with the stability regions of the original ABM methods (Figure 5), one can see that even in the worst case, k = 0, the boundaries of the areas of the proposed methods are wider. Moreover, it can be noted that the size and shape of the stability regions of the fourth and fifth-order methods are comparable with the areas obtained for the ABM methods of the second and third-order while the test problem with symmetric matrix *A* is investigated (Figures 2 and 4). Thus, we can theoretically assume that when implementing variable-step solvers based on semi-explicit and semi-implicit ABM methods, the computational costs over the entire time interval can be less than for predictor–corrector methods of the same order of algebraic accuracy.



Figure 1. Stability regions of the semi-explicit Adams–Bashforth–Moulton methods with k = 0.





Figure 2. Stability regions of the semi-explicit Adams–Bashforth–Moulton methods with k = 1.



Figure 3. Stability regions of the semi-implicit Adams–Bashforth–Moulton methods with k = 0.



Figure 4. Stability regions of the semi-implicit Adams–Bashforth–Moulton methods with k = 1.





Figure 5. Stability regions of Adams–Bashforth–Moulton methods.

Thus, one can conclude that the introduction of semi-explicit integration in the corrector expands the regions of absolute stability of the Adams–Bashforth–Moulton method.

Semi-explicit integration in SEABM/SIABM methods allows one to not approximate state variables, the evaluation of which does not require the first iterations of the finitedifference corrector scheme. Thus, when solving systems of ODEs with lower triangular feedback matrices, the predictor contains only one line of calculations in the semi-explicit algorithm and does not contain any operations when using the semi-implicit method. In practical applications, such systems are rare. However, in large-scale systems simulation, any reduction of computational costs can significantly accelerate multiparametric research and long-term simulation. Let us consider an algorithm for generating a minimal finite-difference scheme for the semi-explicit and semi-implicit Adams–Bashforth–Moulton methods.

2.3. Algorithms for Generating the Minimal Finite-Difference Scheme

To determine which values of state variables do not require approximation by the Adams–Bashforth method, one can define the sequence of calculating the state variables at the correction stage. There are two main requirements to be met:

- 1. Differential equations for which the corresponding right-hand side functions have the least number of occurrences of the different state variables should be calculated first;
- 2. If two or more state equations contain the same number of different variables in the right-hand side functions, then it is worth choosing the one included in the other differential equations.

The first requirement minimizes the number of state variables to calculate them in the predictor stage. The second rule allows using already corrected values to calculate the remaining state variables.

Let us describe the technique that implements the specified requirements using an $N \times N$ feedback matrix, where ones denote feedbacks for each of N variables in the righthand side functions. The feedback matrix is the input of the algorithm. The output is an array of indices of state variables specifying the sequence of calculations in the corrector. The proposed algorithm consists of the following steps:

- 1. A column is added to the feedback matrix end that contains the sums of ones in each row. This number defines the feedbacks number in each equation;
- 2. The feedback matrix is sorted by the last column in ascending order;
- 3. The row with the minimum number in the sum column is considered:
 - (1) If this is the smallest number of all calculated amounts, i.e., the line with the minimum feedbacks number is unique, the index of the corresponding variable is added to the output array;
 - (2) If there are several rows with a minimum feedbacks number, then new sums over all rows are considered for each variable's feedback matrix. Columns corresponding to each of these variables are removed from the matrix. The variable index is chosen to add into the output array, for which at least one of the new calculated sums is the minimum among the sums calculated for all variables. If there are several such variables, then any of them is chosen, for example, the first in order;
- 4. The column and the row corresponding to the variable whose index was added to the output array are removed from the feedback matrix;
- 5. If there are no variables left in the feedback matrix, the algorithm terminates. Otherwise, execution resumes from step 1.

The algorithm for determining the corrector sequence is similar for both semi-explicit and semi-implicit methods. To demonstrate the step-by-step application of the proposed algorithm, we use a hyperchaotic system [23] described by the following system of ordinary differential equations:

$$\begin{cases} x = h(y - x) + u \\ \dot{y} = -fy - xz + w \\ \dot{z} = -l + xy \\ \dot{u} = -y - v \\ \dot{v} = ky + u \\ \dot{w} = gx + my \end{cases}$$

$$(4)$$

where x, y, z, u, v, and w are state variables and h, f, l, k, g, and m are the parameters. This chaotic system demonstrates hidden attractors and multistability [23]. To experimentally investigate such phenomena, multiparametric analysis and long-term simulations are often applied. Therefore, minimizing the finite-difference scheme for such a system is of great importance since it can significantly speed up the research process.

The feedback matrix of system (4) is written as

	x	y		u	v	w	¦Σ
x	1	1		1			3
y	1	1	1			1	4
Z	1	1					2
u		1			1		2
v		1		1			2
w	1	1					2

where the last column corresponds to the number of feedbacks for each state variable.

Following step 2 of the proposed algorithm, we sort the matrix (5) by the last column in ascending order

	x	<i>y</i>	Z	u	v	w	¦Σ
Z	1	1					2
u		1			1		2
v		1		1			2
w	1	1					2
x	1	1		1			3
v	1	1	1			1	4

In matrix (6), four state variables have equal feedback numbers, corresponding to step 3(2) of the algorithm. Let us calculate the number of feedbacks in cases of excluding a column of each such variable and write the result as a matrix, where the excluded variables are located in the columns and the sums in the rows

	Ζ	и	v	W
z	2	2	2	2
и	2	2	(1)	2
v	2	(1)	2	2.
w	2	2	2	2
x	3	2	3	3
<i>y</i>	3	4	4	3

After u and v variables are excluded, the value of the feedbacks number decreases. Therefore, any variable of them can be calculated first in the corrector. Let us choose the variable u and exclude the corresponding column and row from the feedback matrix (5). Then, at the next iteration, the matrix takes the following form:

	x	<i>y</i>		v	w	¦Σ
Z	1	1				2
v		1				1
w	1	1				2
x	1	1				2
y	1	1	1		1	4

After sorting (7), the variable v is chosen for calculation in the corrector since it corresponds to the minimum value of the feedbacks number (step 3(1) in the algorithm):

	<i>x</i>	y	z	v	w	Σ
v		1				1
Z	1	1				2
w	1	1				2
x	1	1				2
y	1	1	1		1	4

Thus, one can write the array determining the order of the calculations at the correction stage after two iterations of the algorithm as [u; v]. Removing the corresponding column and row from the feedback matrix, we obtain

	<i>x</i>	<i>y</i>		w	¦Σ
Z	1	1			2
w	1	1			2
x	1	1			2
y	1	1	1	1	4

The sum column already sorts the obtained matrix. Let us consider a table of feedback sums after removing columns z, w, x:

	z	w	x
Z	2	2	(1)
w	2	2	(1).
x	2	2	(1)
y	3	3	3

When the column corresponding to the variable x is excluded, the number of feedbacks in the matrix (8) decreases. The excluded variable is added to the array [u; v; x]. The new feedback matrix does not require sorting:

	У	Z	W	Σ
Z	1			1
w	1			1
y	1	1	1	3

To exclude from the matrix, the variables z and w are considered, as well as the corresponding table of sums:

	\boldsymbol{z}	w
Z	1	1
w	1	1.
y	2	2

according to which arbitrary variable can be used to calculate in the corrector, for example, *z*. Writing it to the array [*u*; *v*; *x*; *z*], we obtain a feedback matrix consisting of two variables:

From (9), it is possible to determine that the variable w will be calculated next to last in the corrector, and the variable y will be the last one. Thus, the order of calculations when using the semi-explicit or semi-implicit ABM methods for system (4) is given by the array [u; v; x; z; w; y].

Analyzing the matrix of feedbacks and the obtained calculation order of the state variables, one can see that several variables in the corrector, starting from a specific moment, can depend only on values which have already been corrected and not calculated by the predictor. Thus, it is possible to significantly reduce computational costs at each integration step if the corresponding variables are not calculated by the Adams–Bashforth method. To solve this problem, consider an algorithm, the input data of which are the original feedback matrix, and an array of indices that specify the order of calculating the state variables in the corrector. The output data is an array containing indices of the minimum number of variables, the values of which should be approximated in the predictor.

Let the variable *i* set the iteration number of the algorithm. At the first iteration, i = 0, the output array is empty. We also use an auxiliary array, the dimension of which is the same as the variables number and which is initially filled with zeros. The proposed algorithm for the SEABM method consists of the following steps:

- 1. From the array specifying the order of calculations in the corrector, a variable is chosen at the *i*-th index;
- 2. For the chosen variable, using the feedback matrix, the indices of those variables for which there are feedbacks in the right-hand side function are determined;
- 3. In the auxiliary array, in a loop, we go through the values by the indices defined in step 2;
 - (1) If the value from the auxiliary array is 1, then we do nothing;
 - (2) If the value from the auxiliary array is 0, then we change it to 1, and in the output array, we write the variable to which the considered index corresponds;
- 4. If the value of the auxiliary array at the index determined in step 1 is 0, then change it to 1;
- 5. Check the auxiliary array for content 0;
 - (1) If the auxiliary array contains only ones, then the algorithm terminates;
 - (2) If the auxiliary array contains zeros, then increase *i* by 1 and return to step 1.

For semi-implicit integration in the considered algorithm, steps 3 and 4 are reversed, i.e., information about the implicit evaluation of the variable is immediately added to the auxiliary array. This allows elimination of redundant calculations in the predictor in cases where the equation contains feedback on itself.

After completing the steps of the proposed algorithm for system (4), one can find that when applying the SEABM method, approximation of variables [y, v, x] is required, while with the semi-implicit approach, only y and v need to be predicted.

In order to demonstrate the efficiency and value of the proposed approach, we considered two SIABM-based ODE solvers with and without the suggested optimization technique. We estimated the averaged time costs needed for obtaining the solution and the maximal numerical error. System (4) was simulated with h = 5, f = 2.7, l = 5, k = 2, g = -3, m = 1, when simulation time equal 100 s with fixed and variable integration steps. The obtained results are shown in Figure 6. One can see that the proposed optimization technique reduces time costs at least by 25% compared to the original SIABM algorithm. This improvement can significantly speed up the simulation of large-scale systems, as well



as studies of models with different parameter values, and increases the practical value of the proposed SIABM methods.

Figure 6. Performance plots for SIABM ODE solvers with and without the proposed optimization technique. (**a**) Fixed integration step; (**b**) variable integration step.

Let us evaluate the performance of modified semi-explicit ODE solvers in comparison to methods traditionally implemented in popular modeling environments.

3. Experimental Results

3.1. Three-Body Problem

Large-scale ODE systems arise in many fields of science and technology, including simulation of orbital dynamics [24]. One of the known computationally expensive tasks in this area is the simulation of N gravitationally connected bodies, especially when $N \ge 3$. Unlike the two-body problem, there is no general closed-form solution for the three-body case [25], and this dynamical system is chaotic for most initial conditions. Thus, numerical integration is a common choice to investigate this system. We chose this particular case of N-body problem to study the performance of ODE solvers based on semi-explicit multistep methods.

The three-body problem is described by the following system of second-order ODEs:

$$\begin{aligned} \ddot{\mathbf{r}}_{1} &= -Gm_{2} \frac{\mathbf{r}_{1} - \mathbf{r}_{2}}{|\mathbf{r}_{1} - \mathbf{r}_{2}|^{3}} - Gm_{3} \frac{\mathbf{r}_{1} - \mathbf{r}_{3}}{|\mathbf{r}_{1} - \mathbf{r}_{3}|^{3}} \\ \ddot{\mathbf{r}}_{2} &= -Gm_{3} \frac{\mathbf{r}_{2} - \mathbf{r}_{3}}{|\mathbf{r}_{2} - \mathbf{r}_{3}|^{3}} - Gm_{1} \frac{\mathbf{r}_{2} - \mathbf{r}_{1}}{|\mathbf{r}_{2} - \mathbf{r}_{1}|^{3}}, \\ \ddot{\mathbf{r}}_{3} &= -Gm_{1} \frac{\mathbf{r}_{3} - \mathbf{r}_{1}}{|\mathbf{r}_{3} - \mathbf{r}_{1}|^{3}} - Gm_{2} \frac{\mathbf{r}_{3} - \mathbf{r}_{2}}{|\mathbf{r}_{3} - \mathbf{r}_{2}|^{3}} \end{aligned}$$
(10)

where *G* is the gravitational constant, m_i is the mass of the *i*-th body, and $\mathbf{r}_i = (x_i, y_i, z_i)$ are coordinates [26]. Problem (10) can be rewritten as the system of 18 first-order ODEs.

In the experimental study, we evaluated the computational efficiency of the semiexplicit multistep methods to determine the speed-up effect of the proposed optimization technique. Moreover, we considered several ODE solvers, which are common in modern simulation environments such as Wolfram Mathematica, Matlab, LabVIEW, etc. [8]. Solvers under investigation were the fourth-order Adams–Bashforth method (AB), Adams– Moulton method (AM), backward differentiation formula (BDF), and the Adams–Bashforth– Moulton (ABM) technique. For problem (10) we estimated computational costs for semiexplicit (SEABM) modification (recall that for Hamiltonian systems, the SIABM integration reduces to the SEABM method). To keep the implementations equal, all ODE solvers under comparison were programmed in the NI LabVIEW environment with fixed and variable stepsizes using the double-precision data type. The step size is controlled by the following rule:

$$h_{n+1} = h_n \left(\frac{tol}{\|err\|}\right)^{\frac{1}{p+1}},\tag{11}$$

where *err* is the value of local error, *tol* is an error threshold, and *p* is the order of accuracy [27]. The nested recurrent formulas for step control in multistep algorithms lead to the computation of the numerical solution by methods of all orders up to order *p*, which simplifies the application of the Formula (11).

For each of investigated solvers, we estimated averaged time costs for obtaining the solution and the maximal numerical error with respect to the numerical solution obtained by the 8th order Dormand–Prince method with extended precision data. In all experiments, simulation time was 10 s. The obtained results are shown in Figure 7.



Figure 7. Performance plots for investigated multistep ODE solvers while simulating three-body problem with (**a**) fixed integration step; (**b**) variable integration step.

One can see that in the case of fixed-step integration, the modified SEABM method demonstrated the best computational efficiency among the considered algorithms (Figure 7a). Moreover, the time costs for the ODE solver based on semi-explicit calculations were two times less than those of the original ABM method. Such superiority was not observed when simulating low-dimensional systems without applying the proposed optimization techniques [18]. The superiority of the modified semi-explicit method over other explicit and implicit algorithms, including BDF, remained when the integration step was varied (Figure 7b).

3.2. Simulation of Coupled Oscillators Networks

To illustrate the impact of the proposed optimization technique on large-scale systems simulation, we considered a second test problem—a network of N coupled oscillators. Each oscillator can be described using the following equations [28]:

$$\begin{cases} x = a(y - x) + u + rv \\ \dot{y} = -cy - xz \\ \dot{z} = -b + xy \\ \dot{u} = e - dy \\ \dot{v} = k_1 x + k_2 v \end{cases}$$
(12)

where *a*, *b*, *c*, *d*, *e*, k_1 , k_2 , and *r* are free parameters. We investigated an ensemble of identical systems (12) with *a* = 3, *b* = -5, *c* = 1.75, *d* = 1, *e* = 0, k_1 = 6, k_2 = 1, and *r* = -4.

The oscillators were coupled in the ring topology through variable *x* by the following coupling function [29]:

$$F_{x}(i,t) = \frac{\sigma}{2P} \sum_{j=i-P}^{i+P} (x_{j} - x_{i})$$
(13)

where σ is the coupling coefficient, and *P* is the number of the neighbor elements of the ensemble on the right- and left-hand sides of the element with the number *i*. In the experiments, *P* was set to 1.

First, we simulated a single oscillator to estimate the computational cost of semiexplicit and semi-implicit ODE solvers. We explored the same set of methods as for the previous three-body problem. The simulation time was equal to 50 s. The obtained results are shown in Figure 8.



Figure 8. Performance plots for investigated multistep ODE solvers while simulating system (12) with (**a**) fixed integration step; and (**b**) variable integration steps.

One can note that the modified SIABM and SEABM solvers showed the best performance of the considered set of methods.

Then, we measured the time costs of the simulation of the ensemble in the time interval T = 25 s, depending on the network dimension. The obtained results are given in Figure 9. It can be noted that in the case when the number of differential equations was 10^4 , the computational costs for simulation using semi-explicit methods were 25–30% less than for the original ABM technique and were comparable to the costs of the explicit Adams method, which demonstrated a lower accuracy numerical solution, as shown in Figure 8.



Figure 9. Time costs on the simulation with a fixed integration step h = 0.01 depending on the dimension of the network.

Figure 10 shows that the dependence of the time costs on simulation of the ensemble included 2×10^3 oscillators (the total number of equations was 10^4) depending on the simulation interval length. One can see that the semi-explicit modifications of the Adams–Bashforth–Moulton method were 25–30% in computational costs over the other investigated algorithms, both with short and long time intervals.



Figure 10. Simulation time costs dependence on the simulation time. Fixed integration step case, h = 0.01.

Thus, it can be concluded that the application of the SIABM and SEABM methods with the proposed optimization technique can reduce the time costs on solving ODE systems by more than 25% while maintaining the accuracy of the numerical solution in comparison with the original ABM method.

4. Conclusions and Discussion

In this study, we explored the properties of recently proposed semi-explicit and semi-implicit Adams–Bashforth–Moulton methods. It was theoretically predicted and experimentally confirmed that this class of numerical integration methods possesses better numerical stability than original predictor–corrector methods. In addition, we proposed a new technique for synthesizing an optimal finite-difference scheme for semi-explicit multi-step methods. It was shown that the proposed optimization method makes it possible to reduce the number of computations at the prediction stage. Using the three-body problem as an example, we demonstrated that such an approach can decrease computational costs by half compared to the conventional Adams–Bashforth–Moulton method. Moreover, we found that when simulating networks of coupled oscillators that included 2000 elements, it is possible to reduce the time costs by 25% while maintaining the accuracy of the original prediction–correction method.

This paper does not focus on the convergence of the proposed modification of the Adams–Bashforth–Moulton methods, since this property is inherited from the original predictor–corrector method. It is of further interest to study the performance of variable-step solvers based on semi-explicit multistep methods, written in the Nordsieck representation [30]. In such methods, the step control algorithm is based on the reconstruction of the values of the right-hand side functions at the required time points when the integration step is varied. This technique involves the use of vector computations that can be efficiently executed on parallel computing-oriented computers. The study of performance gains when large-scale systems are simulated using this approach is the direction of our further research.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to the anonymous reviewers for their valuable suggestions during the review process.

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

References

- 1. Anderson, S.W.; Sedatole, K. Designing quality into products: The use of accounting data in new product development. *Account. Horiz.* **1998**, *12*, 213.
- Faragó, I.; Georgiev, K.; Havasi, Á.; Zlatev, Z. Efficient numerical methods for scientific applications: Introduction. Comput. Math. Appl. 2013, 65, 297–300. [CrossRef]
- Faragó, I.; Georgiev, K.; Havasi, Á.; Zlatev, Z. Efficient algorithms for large scale scientific computations: Introduction. Comput. Math. Appl. 2014, 67, 2085–2087. [CrossRef]
- 4. Butusov, D.; Karimov, A.; Tutueva, A.; Kaplun, D.; Nepomuceno, E.G. The effects of Padé numerical integration in simulation of conservative chaotic systems. *Entropy* **2019**, *21*, 362. [CrossRef] [PubMed]
- 5. Jaffe, A. Ordering the universe: The role of mathematics. SIAM Rev. 1984, 26, 473–500. [CrossRef]
- 6. Rahrovani, S.; Abrahamsson, T.; Modin, K. An efficient exponential integrator for large nonlinear stiff systems part 1: Theoretical investigation. *Conf. Proc. Soc. Exp. Mech. Ser.* **2014**, *2*, 259–268.
- 7. Jackson, K.R. The numerical solution of large systems of stiff IVPs for ODEs. Appl. Numer. Math. 1996, 20, 5–20. [CrossRef]
- 8. Petcu, D. Software issues in solving initial value problems for ordinary differential equations. *Creat. Math.* **2004**, *13*, 97–110.
- Butusov, D.N.; Ostrovskii, V.Y.; Tutueva, A.V. Simulation of dynamical systems based on parallel numerical integration methods. In Proceedings of the 2015 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), St. Petersburg, Russia, 2–4 February 2015; pp. 56–59.
- 10. Wolfram, S. An elementary introduction to the Wolfram langauge. In *Wolfram Media, Incorporated;* Wolfram Media, Inc.: Champaign, IL, USA, 2017.
- 11. Butusov, D. Adaptive Stepsize Control for Extrapolation Semi-Implicit Multistep ODE Solvers. Mathematics 2021, 9, 950. [CrossRef]
- 12. Faleichik, B.V. Minimal residual multistep methods for large stiff non-autonomous linear problems. *J. Comput. Appl. Math.* **2019**, *389*, 112498. [CrossRef]
- 13. Lopez, S. A predictor-corrector time integration algorithm for dynamic analysis of nonlinear systems. *Nonlinear. Dyn.* **2020**, *101*, 1365–1381. [CrossRef]
- Aboubakr, A.; Shabana, A.A. Efficient and robust implementation of the TLISMNI method. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, Boston, MA, USA, 2–5 August 2015.
- Oghonyon, J.G.; Okunuga, S.A.; Omoregbe, N.A.; Agboola, O.O. A Computational Approach in Estimating the Amount of Pond Pollution and Determining the Long Time Behavioural Representation of Pond Pollution Model. *Glob. J. Pure Appl. Math.* 2015, 11, 2773–2785.
- Hamid, A.M.R.; Ahmad, R.R.; Din, U.K.S.; Ismail, A. Modified predictor-corrector multistep method using arithmetic mean for solving ordinary differential equations. In Proceedings of the AIP Conference Proceedings, American Institute of Physics, Putrajaya, Malaysia, 18–20 December 2012; Volume 1522, pp. 703–707.
- 17. Biasa, T.; Majid, Z.A.; Suleiman, M. Predictor-corrector block iteration method for solving ordinary differential equations. *Sains Malays.* **2011**, *40*, 659–664.
- Tutueva, A.; Karimov, T.; Butusov, D. Semi-Implicit and Semi-Explicit Adams-Bashforth-Moulton Methods. *Mathematics* 2020, *8*, 780. [CrossRef]
- 19. Cellier, F.E.; Kofman, E. Continuous System Simulation; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
- 20. Kannan, R.; Wang, Z.J. A study of viscous flux formulations for a p-multigrid spectral volume Navier stokes solver. *J. Sci. Comput.* **2009**, *41*, 165–199. [CrossRef]
- 21. Kannan, R. A high order spectral volume method for elastohydrodynamic lubrication problems: Formulation and application of an implicit p-multigrid algorithm for line contact problems. *Comput. Fluids* **2011**, *48*, 44–53. [CrossRef]

- 22. Butusov, D.N.; 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]
- 23. Yang, L.; Yang, Q.; Chen, G. Hidden attractors, singularly degenerate heteroclinic orbits, multistability and physical realization of a new 6D hyperchaotic system. *Comm. Nonlinear. Sci. Numer. Simul.* **2020**, *90*, 105362. [CrossRef]
- 24. Kapfer, E.M.; Stapor, P.; Hasenauer, J. Challenges in the calibration of large-scale ordinary differential equation models. *IFAC-Pap.* **2019**, *52*, 58–64. [CrossRef]
- Gowers, T.; Barrow-Green, J.; Leader, I. The Three-Body Problem. In *The Princeton Companion to Mathematics*; Princeton University Press: Princeton, NJ, USA, 2008; pp. 726–728.
- 26. Marchal, C. The Three-Body Problem; Elsevier: Amsterdam, The Netherlands, 2012.
- 27. Hairer, E.; Nørsett, S.P.; Wanner, G. Solving Ordinary Differential Equations I: Nonstiff Problems; Springer: Berlin, Germany, 1993; Volume 8.
- Yang, Q.; Yang, L.; Ou, B. Hidden hyperchaotic attractors in a new 5D system based on chaotic system with two stable node-foci. *Int. J. Bifurc. Chaos* 2019, 29, 1950092. [CrossRef]
- Anishchenko, V.S.; Strelkova, G.I. Chimera structures in the ensembles of nonlocally coupled chaotic oscillators. *Radiophys. Quantum Electron.* 2019, *61*, 659–671. [CrossRef]
- 30. Talebi, B.; Abdi, A. Nordsieck representation of high order predictor-corrector Obreshkov methods and their implementation. *CMDE* **2019**, 7, 16–27.