Applied Mathematics and Computation 362 (2019) 124567

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

An efficient optimized adaptive step-size hybrid block method for integrating differential systems

Gurjinder Singh[a,*], Arvind Garg[b], V. Kanwar[c], Higinio Ramos[d,e]

[a] Department of Mathematical Sciences, I. K. Gujral Punjab Technical University Jalandhar, Main Campus, Kapurthala, Punjab 144603, India
[b] D. A. V. College, Sector-10, Chandigarh 160011, India
[c] University Institute of Engineering and Technology, Panjab University, Chandigarh 160 014, India
[d] Scientific Computing Group, Universidad de Salamanca, Plaza de la Merced Salamanca 37008, Spain
[e] Escuela Politécnica Superior de Zamora, Campus Viriato, Zamora 49022, Spain

ARTICLE INFO

Keywords:
ODEs
Hybrid methods
Block methods
Adaptive step-size
Optimization strategy

ABSTRACT

This paper deals with the development, analysis and implementation of an optimized hybrid block method having different features, for integrating numerically initial value ordinary differential systems. The hybrid nature of the proposed one-step scheme allows us to bypass the first Dahlquist's barrier on linear multi-step methods. The theory of interpolation and collocation has been used in the development of the method. We assume an appropriate polynomial representation of the theoretical solution of the problem and consider three off-step points in a one-step block. One of these three off-step points is fixed and the other two off-step points are optimized in order to minimize the local truncation errors of the main method and other additional formula. The resulting scheme is of order five having the property of $\mathcal{A}$-stability. An embedded-type approach is used in order to formulate the proposed method in adaptive form, showing a high efficiency. The adaptive method is tested on well-known differential systems viz. the Robertson's system, a Gear's system, a system related with Jacobi elliptic functions, the Brusselator system, and the Van der Pol system, and compared with some well-known numerical codes in the scientific literature.

1. Introduction

The mathematical formulation of many physical phenomena results in differential systems which are very difficult or even impossible to solve analytically. In such situations, differential systems are usually dealt numerically in order to find approximate solutions. This article particularly addresses the numerical solution of initial value ordinary differential systems. Conventionally, Runge–Kutta and linear multi-step methods are used in order to get numerical approximation to the theoretical solution of the problem. In these days, many numerical codes are available as built-in functions in Computer Algebra Systems (CAS), like MATLAB or MATHEMATICA, which make the task of obtaining numerical approximations to the theoretical solution accurately and efficiently. These codes are particularly designed to handle problems having different types of solutions, for example stiff, non-stiff, singular etc., in variable step-size mode.

* Corresponding author.
E-mail addresses: gurjinder11@gmail.com (G. Singh), gargarvind.1@gmail.com (A. Garg), vkanwar@pu.ac.in (V. Kanwar), higra@usal.es (H. Ramos).

https://doi.org/10.1016/j.amc.2019.124567
0096-3003/

To develop new efficient and simple structured methods, having good stability characteristics, is a subject of continuous development in this field. In this article, our objective is to develop and analyze a novel efficient numerical integrator for solving initial value systems of the form

$$\mathbf{Z}' = \mathbf{F}(t, \mathbf{Z}); \quad \mathbf{Z}(t_0) = \mathbf{Z_0}, \tag{1}$$

where $t \in [t_0, t_N]$, $\mathbf{Z} : [t_0, t_N] \to \mathbb{R}^m$, $\mathbf{F} : [t_0, t_N] \times \mathbb{R}^m \to \mathbb{R}^m$. Firstly, we assume that the system (1) satisfies the conditions of the *Existence and Uniqueness Theorem* for initial-value problems. In order to solve the system (1) numerically, the interval of integration $[t_0, t_N]$ is discretized as follows: $t_n = t_0 + nh$, $n = 0, 1, 2, 3 \ldots, N$; $h_n = t_{n+1} - t_n$. The step-sizes h_n can be taken as a constant or variable in the interval of interest, according to the implementation of the integrator. The numerical value of the theoretical solution at t_n is denoted by $\mathbf{Z}_n \approx \mathbf{Z}(t_n)$. Besides the Runge–Kutta and linear multi-step methods, other well-known classes of methods for integrating (1) include block methods, hybrid methods, exponentially fitted methods and trigonometrically fitted methods, etc. For a detailed survey on different classes of methods, one can consult the books by Butcher [1] or by Hairer et al. [2,3] (and references therein). One can consult specific references concerning the solution of different types of differential systems, for example see [4–45].

The numerical method in the present paper is a combination of block and hybrid nature. The first Dahlquist's barrier imposes restrictions on number of steps and order of stable linear multi-step methods [22]. According to this barrier, a zero-stable linear multi-step method will have order p with $p \le k+1$ if k is odd, and $p \le k+2$ if k is even. In order to bypass this barrier many authors have proposed hybrid methods that also utilize information of the solution at off-step points. In this way, one can bypass the first Dahlquist's barrier by using information of the solution at off-step points. These methods are also called modified linear multi-step methods [1]. On the other hand, block methods produce information of the solution at several points simultaneously. For more details on block methods, one can consult the book by Brugnano et al. [4] and references therein.

In the present paper, we have developed a one-step hybrid method which possesses the characteristic of $\mathcal{A}$-stability. In the subsequent sections of the article, we will address the theoretical analysis of the method and its implementation on well-known initial value ordinary differential systems in order to show its good performance.

2. Development of the hybrid block method

For convenience, we consider the derivation of the method for solving the problem in (1) in the scalar case, that is, for $m = 1$. Keeping this in mind, we assume an approximation to the theoretical solution of (1) by an appropriate interpolating polynomial of the form

$$Z(t) \approx Q(t) = \sum_{j=0}^{u+v-1} \xi_j \Theta_j(t), \tag{2}$$

on the interval $[t_n, t_{n+1}]$ considering initially a constant step size, $h = t_{n+1} - t_n$. Here, $\xi_j's$ are unknown constants, and $\Theta_j(t) = (t - t_n)^j$ are polynomial basis functions. The values u and v are regarded as the number of interpolation and distinct collocation points, respectively, that are considered to satisfy $0 \le u \le k$, $v > 0$. Further, the positive integer k stands for the number of steps in the block. The method is developed by specifying the following parameters, $u = 1$, $v = 5$ and $k = 1$. We also consider three off-step points $t_{n+\nu_1} = t_n + \nu_1 h$, $t_{n+\nu_2} = t_n + \nu_2 h$ and $t_{n+\nu_3} = t_n + \nu_3 h$ in (t_n, t_{n+1}) with $\nu_2 = 1/2$. The two unknown off-step points will be optimized in order to minimize the local truncation error in the main method and other additional formula.

We impose the following restrictions on the approximating polynomial (2) as follows:

(i) Consider that the approximating polynomial in (2) interpolates the theoretical solution of (1) at the point $t = t_n$, that is,

$$\sum_{j=0}^{5} \xi_j \Theta_j(t_n) = Z(t_n). \tag{3}$$

(ii) Next, we impose that the first derivative of the polynomial in (2) coincides with the derivative of the true solution of (1) at the points t_{n+j}, $j = 0, \nu_1, \nu_2, \nu_3, 1$, that is,

$$\sum_{j=1}^{5} \xi_j \Theta_j'(t_{n+i}) = Z'(t_{n+i}), \ \ i = 0, \nu_1, \nu_2, \nu_3, 1. \tag{4}$$

From the equations in (3) and (4), we get a system of six equations in six unknowns. We solve this system to get the values of the coefficients $\xi_j's$. By inserting the values of $\xi_j's$ into (2), we get the method written in the following convenient form

$$Z(t) \approx \lambda(t) Z_n + h \sum_{j=0}^{1} \delta_j(t) F_{n+j} + h \sum_{j=1}^{3} \delta_{\nu_j}(t) F_{n+\nu_j}, \tag{5}$$

Table 1
Coefficients of the method.

Z	λ	δ_0	δ_{ν_1}	δ_{ν_2}	δ_{ν_3}	δ_1
$Z_{n+\nu_1}$	1	$\frac{83+29\sqrt{3}}{360(3+\sqrt{3})}$	$\frac{171+63\sqrt{3}}{360(3+\sqrt{3})}$	$\frac{32-64\sqrt{3}}{360(3+\sqrt{3})}$	$\frac{81-27\sqrt{3}}{360(3+\sqrt{3})}$	$\frac{-(7+\sqrt{3})}{360(3+\sqrt{3})}$
$Z_{n+\nu_2}$	1	$\frac{31}{480}$	$\frac{72+45\sqrt{3}}{480}$	$\frac{64}{480}$	$\frac{72-45\sqrt{3}}{480}$	$\frac{1}{480}$
$Z_{n+\nu_3}$	1	$\frac{83-29\sqrt{3}}{360(3-\sqrt{3})}$	$\frac{81+27\sqrt{3}}{360(3-\sqrt{3})}$	$\frac{32+64\sqrt{3}}{360(3-\sqrt{3})}$	$\frac{171-63\sqrt{3}}{360(3-\sqrt{3})}$	$\frac{-7+\sqrt{3}}{360(3-\sqrt{3})}$
Z_{n+1}	1	$\frac{1}{15}$	$\frac{3}{10}$	$\frac{4}{15}$	$\frac{3}{10}$	$\frac{1}{15}$

where $\lambda(t), \delta_j(t)$ and $\delta_{\nu_j}(t)$ are continuous coefficients of the method. Here, $F_{n+j} = F(t_{n+j}, Z_{n+j})$ and $F_{n+\nu_j} = F(t_{n+\nu_j}, Z_{n+\nu_j})$. The main formula of the hybrid method is obtained by evaluating (5) at the point $t = t_{n+1}$. The other additional formulas of the method will be obtained by evaluating (5) at the off-step points $t_{n+\nu_1}, t_{n+\nu_2}$ and $t_{n+\nu_3}$. Note that when we evaluate (5) at the points $t = t_{n+1}$ and $t_{n+\nu_2}$, we get the main formula and another formula at the midpoint of the interval $[t_n, t_{n+1}]$, which approximate the values of the theoretical solution $Z(t)$ at t_{n+1} and $t_{n+\nu_2}$. Both formulas depend upon the parameters ν_1 and ν_3 which are related to the off-step points $t_{n+\nu_1}$ and $t_{n+\nu_3}$. The expressions of $Z(t_{n+1})$ and $Z(t_{n+\nu_2})$ are not presented here, as they are very cumbersome expressions, although they can be obtained easily using a computer algebra system. In order to get adequate values of the parameters ν_1 and ν_3, we adopt the following procedure:

(i) Firstly, we obtain the local truncation errors of the formulas for $Z(t_{n+1})$ and $Z(t_{n+\nu_2})$ after expanding them in Taylor series about the point t_n. Those local truncation errors are given respectively by

$$\mathcal{L}(Z(t_{n+1}), h) = \frac{(-1+\nu_1+\nu_3)Z^{(6)}(t_n)h^6}{14400} + \mathcal{O}(h^7) \tag{6}$$

and

$$\mathcal{L}(Z(t_{n+\nu_2}), h) = \frac{(-2+\nu_1(7-30\nu_3)+7\nu_3)Z^{(6)}(t_n)h^6}{230400} + \mathcal{O}(h^7). \tag{7}$$

(ii) In order to find the optimized values of ν_1 and ν_3, we consider the coefficients of the principal terms in the local truncation errors in (6) and (7), and equate them to zero. Thus, we get the following nonlinear system of equations

$$\begin{aligned} -1+\nu_1+\nu_3 &= 0 \\ -2+\nu_1(7-30\nu_3)+7\nu_3 &= 0. \end{aligned}$$

After solving this nonlinear system of equations, we get the optimized values of ν_1 and ν_3 as follow (we note that the above system is symmetric in ν_1 and ν_3, and the solution is unique with the restriction $0 < \nu_1 < \nu_3 < 1$)

$$\nu_1 = \frac{1}{6}(3-\sqrt{3}) \approx 0.211325, \quad \nu_3 = \frac{1}{6}(3+\sqrt{3}) \approx 0.788675\,.$$

Substituting these values in the local truncation errors (6) and (7), we get

$$\mathcal{L}(Z(t_{n+1}), h) = \frac{Z^{(7)}(t_n)h^7}{3628800} + \mathcal{O}(h^8)\,, \tag{8}$$

$$\mathcal{L}(Z(t_{n+\nu_2}), h) = \frac{Z^{(7)}(t_n)h^7}{7257600} + \mathcal{O}(h^8)\,. \tag{9}$$

Using the obtained values of ν_1 and ν_3, after substituting the values of $t = t_{n+\nu_1}, t_{n+\nu_2}, t_{n+\nu_3}, t_{n+1}$ in (5) we obtain the four formulas of the hybrid method. The coefficients of the new method can be written compactly in Table 1.

This method will produce approximate solutions of (1) simultaneously at the points $t_{n+\nu_1}, t_{n+\nu_2}, t_{n+\nu_3}$ and t_{n+1}. We note that this is an implicit method, and thus to get the solution, a system of equations must be solved in each iteration. The usual choice to solve the system is the Newton's method (or its variants [26], although sometimes suitable ad-hoc procedures can be employed, based on a diagonalization of the matrices involved or, alternatively, on the use of suitable splitting techniques [27,28]).

3. Theoretical analysis of the hybrid block method

This section is concerned with theoretical analysis of the proposed hybrid block method given in Table 1.

3.1. Local truncation error, order and consistency

The hybrid block method developed in the previous section may be written in the following form

$$U\ \mathbf{Z}_n = h\ V\ \mathbf{F}_n,$$

where U and V are matrices of coefficients of dimension 4×5 that can be easily obtained from Table 1, and

$$\mathbf{Z}_n = (Z_n, Z_{n+\nu_1}, Z_{n+\nu_2}, Z_{n+\nu_3}, Z_{n+1})^T,$$
$$\mathbf{F}_n = (F_n, F_{n+\nu_1}, F_{n+\nu_2}, F_{n+\nu_3}, F_{n+1})^T.$$

Let $Z(t)$ be a sufficiently smooth function and consider the linear difference operator $\bar{\mathcal{L}}$ associated with the hybrid block method, given by

$$\bar{\mathcal{L}}[Z(t), h] = \sum_j \bar{\alpha}_j Z(t + jh) - h \sum_j \bar{\beta}_j Z'(t + jh), \quad j = 0, \nu_1, \nu_2, \nu_3, 1, \tag{10}$$

where $\bar{\alpha}_j$ and $\bar{\beta}_j$ are respectively designated as columns of the matrices U and V.

Both the proposed hybrid block method given in Table 1 and the difference operator $\bar{\mathcal{L}}$ are said to be of order p if after expanding $Z(t + jh)$ and $Z'(t + jh)$ by Taylor series about t, we have

$$\bar{\mathcal{L}}[Z(t), h] = \bar{\kappa}_0 Z(t) + \bar{\kappa}_1 h Z'(t) + \bar{\kappa}_2 h^2 Z''(t) + \cdots + \bar{\kappa}_p h^p Z^{(p)}(t) + \ldots, \tag{11}$$

with $\bar{\kappa}_0 = \bar{\kappa}_1 = \bar{\kappa}_2 = \cdots = \bar{\kappa}_p = 0$ and $\bar{\kappa}_{p+1} \neq 0$. Here, the $\bar{\kappa}_i's$ are vectors and $\bar{\kappa}_{p+1}$ is called vector of error constants. For the proposed method, we get $\bar{\kappa}_0 = \bar{\kappa}_1 = \bar{\kappa}_2 = \bar{\kappa}_3 = \bar{\kappa}_4 = \bar{\kappa}_5 = 0$ and

$$\bar{\kappa}_6 = \left(\frac{-1}{311040}, \quad 0, \quad \frac{-1}{311040}, \quad 0\right)^T.$$

Thus, the hybrid block method specified in Table 1 has order 5. Hence, the proposed hybrid block method is consistent with the differential equation in (1).

3.2. Zero-stability

The concept of zero-stability is related with the behavior of the method when $h \to 0$. Considering that the step-size tends to zero in the proposed method, we have

$$Z_{n+\nu_1} = Z_n$$
$$Z_{n+\nu_2} = Z_n$$
$$Z_{n+\nu_3} = Z_n$$
$$Z_{n+1} = Z_n.$$

This can be expressed in the following compact form

$$I_4\, \hat{Z}_\eta - \bar{U}\, \hat{Z}_{\eta-1} = 0, \tag{12}$$

where $\hat{Z}_\eta = (Z_{n+\nu_1}, Z_{n+\nu_2}, Z_{n+\nu_3}, Z_{n+1})^T$, $\hat{Z}_{\eta-1} = (Z_{n+\nu_1-1}, Z_{n+\nu_2-1}, Z_{n+\nu_3-1}, Z_n)^T$, I_4 stands for the identity matrix of dimension four, and $\bar{U}$ is the matrix

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

It can be easily verified that the characteristic polynomial is given by $|[I_4\xi - \bar{U}]| = \xi^3(\xi - 1)$ whose roots ξ_i satisfy $|\xi_i| < 1$ and those which satisfy $|\xi_i| = 1$ are just simple roots. Thus, the method is zero-stable [2].

3.3. Convergence

The necessary and sufficient conditions for a method to be convergent are that it is both consistent and zero-stable [29]. As we have established the consistency and zero-stability of the proposed method, then it implies that the method is convergent.

3.4. Linear stability analysis

In order to determine the stability region of the given numerical method, we consider the test equation made famous by Dahlquist

$$Z'(t) = \lambda Z(t), \quad \mathrm{Re}(\lambda) < 0. \tag{13}$$

It can be easily observed that the true solution $Z(t) = e^{\lambda t}$ of this test equation decays as t tends to ∞ and we expect that the numerical solution provided by the proposed scheme for solving the problem in (13) behaves in the same manner. By

applying the proposed method to the Dahlquist's test equation (13), we get a system that may be expressed in a matrix form as

$$U\begin{pmatrix} Z_{n+\nu_1} \\ Z_{n+\nu_2} \\ Z_{n+\nu_3} \\ Z_{n+1} \end{pmatrix} = V\begin{pmatrix} Z_{n+\nu_1-1} \\ Z_{n+\nu_2-1} \\ Z_{n+\nu_3-1} \\ Z_n \end{pmatrix} \tag{14}$$

where the matrix U is given by

$$U = \begin{pmatrix} U_{11} & U_{12} & U_{13} & U_{14} \\ U_{21} & U_{22} & U_{23} & U_{24} \\ U_{31} & U_{32} & U_{33} & U_{34} \\ U_{41} & U_{42} & U_{43} & U_{44} \end{pmatrix}$$

with

$U_{11} = -9(-40(3+\sqrt{3}) + (19+7\sqrt{3})H), U_{12} = 32(-1+2\sqrt{3})H$
$U_{13} = 27(-3+\sqrt{3})H, U_{14} = (7+\sqrt{3})H, U_{21} = \frac{1}{480}(72+45\sqrt{3})H, U_{22} = -1+\frac{2H}{15}$
$U_{23} = \frac{1}{480}(72-45\sqrt{3})H, U_{24} = \frac{H}{480}, U_{31} = 27(3+\sqrt{3})H, U_{32} = 32(1+2\sqrt{3})H$
$U_{33} = -9(-40(-3+\sqrt{3}) + (-19+7\sqrt{3})H), U_{34} = (-7+\sqrt{3})H, U_{41} = \frac{3H}{10},$
$U_{42} = \frac{4H}{15}, U_{43} = \frac{3H}{10}, U_{44} = -1+\frac{H}{15}$ and the matrix V is given by

$$V = \begin{pmatrix} 0 & 0 & 0 & 360(3+\sqrt{3}) + (83+29\sqrt{3})H \\ 0 & 0 & 0 & \frac{-1}{480}(480+31H) \\ 0 & 0 & 0 & 360(-3+\sqrt{3}) + (-83+29\sqrt{3})H \\ 0 & 0 & 0 & \frac{-1}{15}(15+H) \end{pmatrix},$$

where $H = \lambda h$.

Then, the application of the hybrid block method to the Dahlquist's test equation may be expressed as follows

$$\begin{pmatrix} Z_{n+\nu_1} \\ Z_{n+\nu_2} \\ Z_{n+\nu_3} \\ Z_{n+1} \end{pmatrix} = P(H)\begin{pmatrix} Z_{n+\nu_1-1} \\ Z_{n+\nu_2-1} \\ Z_{n+\nu_3-1} \\ Z_n \end{pmatrix} \tag{15}$$

where $P(H) = U^{-1}V$ is called the stability matrix.

Now, in order to study the stability characteristics of the hybrid block method, we consider the spectral radius $\rho[P(H)]$ of the stability matrix given by

$$\rho[P(H)] = \frac{M(H)}{N(H)}, \tag{16}$$

where

$$M(H) = 1440 + 720H + 156H^2 + 18H^3 + H^4,$$
$$N(H) = 1440 - 720H + 156H^2 - 18H^3 + H^4.$$

Thus, we have $|\rho[P(H)]| < 1$ for $\mathrm{Re}(H) < 0$. This establishes that the proposed hybrid block method is $\mathcal{A}$-stable. Fig. 1 represents the absolute stability region of the proposed method.

4. Formulation as an adaptive method

To formulate the present method in an adaptive step-size form, we have adopted the approach considered by Shampine et al. [30]. This formulation can be done by executing a combination of the proposed method and a lower order method simultaneously. The purpose of considering the lower order scheme is just to assess the local error that we want to estimate at the final point of $[t_n, t_{n+1}]$, while the higher order method is used to advance the integration process. This is a similar procedure to the one used in the case of embedded Runge–Kutta methods.

In order not to increase the number of function evaluations in the simultaneous implementation of both methods, the lower order method is obtained considering that it uses the same function evaluations as with the higher order method. In this way, this embedding-like procedure will not increase the computational cost in terms of number of function evaluations.

The following fourth order formula

$$Z_{n+1} = Z_n + \frac{h}{2}(F_{n+\nu_1} + F_{n+\nu_3}), \tag{17}$$

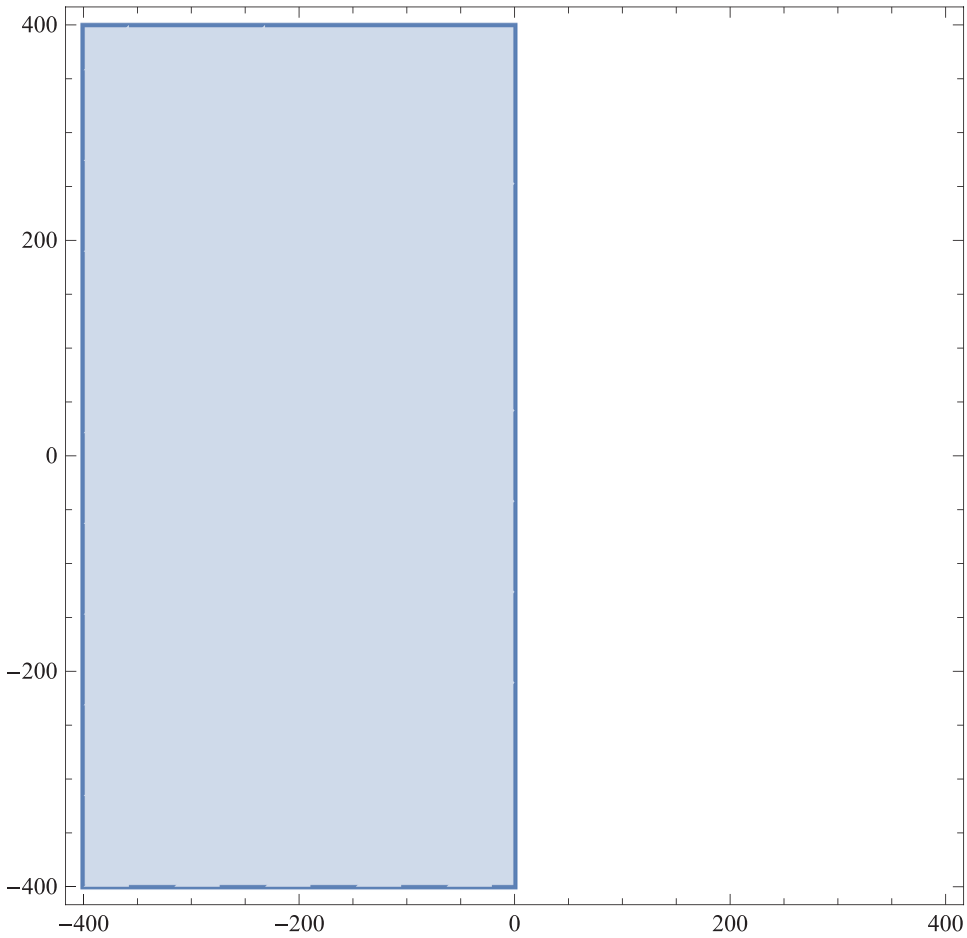

Fig. 1. Stability region of the method given in Table 1.

whose local truncation error is given by LTE$= \frac{Z^{(5)}(t_n)h^5}{4320} + \mathcal{O}\left(h^6\right)$, has been implemented simultaneously with the higher order method. One can observe that the lower order method (17) uses two function evaluations which are already used in the main method. Therefore, there will be no extra function evaluation cost needed in order to estimate the local error. The difference between the approximate solutions obtained by the higher order method and the lower order method will be used as an error estimate (EST) for the local error that will be further used in deciding the appropriate size of the new step. Thus, we consider the following criterion in order to decide the size of the next step:

$$h_{new} = \theta\, h_{old} \left(\frac{Tol}{\|EST\|} \right)^{1/(p+1)}, \tag{18}$$

where p is order of the lower order method, and $0 < \theta < 1$, is called the safety factor. The major objective of the safety factor θ is to avoid large variations in the new step. The notation Tol is designated for user defined tolerance.

We further impose the following conditional structure in the execution of the method

If $h_{min} \leq h_{new} \leq h_{max}$ **then** $h_{old} = h_{new}$.

The above criterion can be expressed in the following simple steps:

1. If $\|EST\| < Tol$, then accept the obtained values and double the step-size in order to advance the integration process.
2. If $\|EST\| \geq Tol$, then redefine the current step-size using the above criterion and redo the calculations.

In the scientific literature, different strategies are used in order to select the appropriate size of the initial step which we denote as h_{ini}, for example see [32]. On the other way, one can choose a small h_{ini} (see for reference [33]) and the algorithm will automatically redefine if it is required.

Table 2
Numerical results for Section 5.1.

h_{ini}	Tol	Method	MaxErr	FEvals	C. time
10^{-2}	10^{-9}	RADAU	5.2187×10^{-10}	504	0.080
		ode15s	2.6289×10^{-9}	351	0.088
		Opthbm	1.3022×10^{-13}	290	0.071
10^{-3}	10^{-10}	RADAU	6.0419×10^{-12}	735	0.094
		ode15s	1.3305×10^{-9}	475	0.097
		Opthbm	2.0650×10^{-14}	435	0.092

5. Numerical experiments

This section deals with the implementation of the proposed adaptive step-size method on some well-known differential systems existing in the scientific literature. Some notations have been used in the following Tables which are designated as follows: Tol:Tolerance, FEvals: Number of function evaluations, MaxErr: Maximum absolute error in all the components of the solution at the integration interval or at the end point, and C. time: CPU time measured in seconds. The following ODE solvers have been considered for comparisons:

1. **ode15s:** ode15s is a variable-step, variable-order (VSVO) IVP solver. This code is based on the numerical differentiation formulas (NDFs) of orders 1 to 5. This code is a built-in ODE solver in MATLAB which is specifically designed for solving stiff systems. For more details on this code, one can consult the references [30,31].
2. **RADAU:** This code is based on implicit Runge–Kutta methods (Radau-IIa) with variable order (1, 5, 9, 13) and step size control. This code is also specifically designed for solving stiff systems. We have used a MATLAB code of this scheme (see MatlabStiff package in http://www.unige.ch/hairer/software.html).
3. **Opthbm:** This is the new proposed scheme in this article which is implemented in adaptive size mode using the step-size strategy given in Section 4. At each step of integration, one has to solve a system of linear/nonlinear equations arising due to the implicit nature of the scheme. We have developed a MATHEMATICA code of this scheme and enhanced the code using the FindRoot command in order to solve the resulting nonlinear system and the NSolve command in case of linear systems.

Note. The used MATLAB codes ode15s and RADAU are variable-step, variable-order methods which use different criteria for changing the step size and order of the schemes. The new scheme presented here uses the change of step-size strategy given in Section 4. For solving the test problems, we have considered $Tol = AbsTol = RelTol$, where the notations *AbsTol* and *RelTol* stand for absolute error tolerance and relative error tolerance respectively. Recall that *AbsTol* and *RelTol* are used by the ODE solvers ode15s and RADAU, whereas the *Tol* will be used by the new scheme in order to estimate the error at each step of integration. The codes ode15s and RADAU are implemented in MATLAB2013a on a laptop with configuration i7-7500U, 2.70 GHz, Win10. The new scheme is implemented in MATHEMATICA, version 2011 on the laptop with same configuration.

5.1. Robertson's chemistry problem

Firstly, we have taken the well-known Robertson's nonlinear stiff differential system arising in chemical sciences [34] that describes the kinetics of an autocatalytic reaction. It is considered in the well-known 'Test Set for Initial Value Problem Solvers' [37] and also studied by Cash [38,39].

$$\begin{aligned} Z_1' &= -0.04\, Z_1 + 10^4\, Z_2\, Z_3, & Z_1(0) &= 1 \\ Z_2' &= 0.04\, Z_1 - 10^4\, Z_2\, Z_3 - 3 \times 10^7\, Z_2^2, & Z_2(0) &= 0 \\ Z_3' &= 3 \times 10^7\, Z_2^2, & Z_3(0) &= 0. \end{aligned} \tag{19}$$

The above problem is integrated over the interval [0, 40] as in [38]. The numerical results in Table 2 have been obtained by using

$$(h_{ini}, Tol) = (10^{-\kappa}, 10^{-(\kappa+7)}),\ \kappa = 2, 3.$$

In order to compare the errors, the following reference solution at the end point $t_N = 40$

$$\begin{aligned} Z_1 &= 0.7158270687194135 \\ Z_2 &= 9.185534764558135 \times 10^{-6} \\ Z_3 &= 0.28416374574582 \end{aligned} \tag{20}$$

considered in [40] is used for the problem. It can be viewed from the results in Table 2 that the proposed adaptive step-size code performs better among the ode solvers considered for comparison.

Table 3
Numerical results for Section 5.2.

h_{ini}	Tol	Method	MaxErr	FEvals	C. time
10^{-1}	10^{-11}	RADAU	1.8130×10^{-11}	285	0.047
		ode15s	3.2508×10^{-11}	324	0.081
		Opthbm	3.3306×10^{-15}	215	0.046
10^{-2}	10^{-12}	RADAU	4.7764×10^{-12}	349	0.057
		ode15s	4.4668×10^{-12}	406	0.097
		Opthbm	5.3290×10^{-15}	315	0.053

5.2. Gear's problem

Consider the problem suggested by Gear [41]

$$\begin{aligned} Z_1' &= -0.013\ Z_1 - 1000\ Z_1\ Z_3, & Z_1(0) &= 1 \\ Z_2' &= -2500\ Z_2\ Z_3, & Z_2(0) &= 1 \\ Z_3' &= -0.013\ Z_1 - 1000\ Z_1\ Z_3 - 2500\ Z_2\ Z_3, & Z_3(0) &= 0. \end{aligned} \tag{21}$$

The problem is integrated over the interval [0, 50] as in [41]. The numerical results in Table 3 have been obtained by considering

$$(h_{ini}, Tol) = (10^{-\kappa}, 10^{-(\kappa+10)}), \kappa = 1, 2.$$

In order to compare errors, the following reference solution at end point $t_N = 50$

$$\begin{aligned} Z_1 &= 0.59765469806558128638 \\ Z_2 &= 1.40234340854787827842 \\ Z_3 &= -1.8933865404351958485 \times 10^{-6} \end{aligned}$$

obtained by using the NDSolve command with a 10th order Implicit Runge–Kutta method in MATHEMATICA, has been used. The proposed adaptive hybrid code again performs well for this problem.

5.3. Jacobi elliptic functions

As a next example, we consider the well-known system concerned with Jacobi elliptic functions sn, cn and dn which is frequently used in numerical experimentation [35]

$$\begin{aligned} sn'(t) &= cn(t).dn(t), & sn(0) &= 0 \\ cn'(t) &= -sn(t).dn(t), & cn(0) &= 1 \\ dn'(t) &= -m.sn(t).cn(t), & dn(0) &= 1, \end{aligned} \tag{22}$$

where $m = \frac{1}{2}$. The theoretical solution of the system can be written in terms of series as follows

$$\begin{aligned} sn(t) &= \frac{2\pi}{\sqrt{\frac{1}{2}}K} \sum_{n=0}^{\infty} \frac{q^{n+\frac{1}{2}}}{1-q^{2n+1}} \sin((2n+1)v) \\ cn(t) &= \frac{2\pi}{\sqrt{\frac{1}{2}}K} \sum_{n=0}^{\infty} \frac{q^{n+\frac{1}{2}}}{1+q^{2n+1}} \cos((2n+1)v) \\ dn(t) &= \frac{\pi}{2K} + \frac{2\pi}{K} \sum_{n=1}^{\infty} \frac{q^n}{1+q^{2n}} \cos(2nv), \end{aligned} \tag{23}$$

here $q = e^{-\pi}$, $v = \frac{\pi t}{2K}$ and $K = \int_0^{\frac{\pi}{2}} \frac{d\ \theta}{\sqrt{1-\frac{1}{2}\sin^2\theta}} \approx 1.85$. We have solved this problem over the integration interval [0, 50]. For this problem, we have considered

$$(h_{ini}, Tol) = (10^{-\kappa}, 10^{-(\kappa+3)}), \kappa = 1, 2.$$

The numerical results presented in Table 4 demonstrate the good performance of the new scheme.

5.4. Brusselator system

Now, we consider a diffusion-free nonlinear Brusselator system [2] which is also studied in [6]

$$\begin{aligned} Z_1' &= V + Z_1^2\ Z_2 - (U+1)\ Z_1, & Z_1(0) &= Z_1^0 \\ Z_2' &= U\ Z_1 - Z_1^2\ Z_2, & Z_2(0) &= Z_2^0. \end{aligned} \tag{24}$$

Table 4
Numerical results for Section 5.3.

h_{ini}	Tol	Method	MaxErr	FEvals	C. time
10^{-1}	10^{-4}	RADAU	1.0737×10^{-3}	1115	0.105
		ode15s	8.5313×10^{-3}	646	0.112
		Opthbm	8.6642×10^{-6}	430	0.102
10^{-2}	10^{-5}	RADAU	2.6296×10^{-4}	1323	0.115
		ode15s	1.0373×10^{-3}	771	0.117
		Opthbm	2.0913×10^{-7}	670	0.110

Table 5
Numerical results for Section 5.4.

h_{ini}	Tol	Method	MaxErr	FEvals	C. time
10^{-3}	10^{-6}	RADAU	4.0992×10^{-7}	1692	0.088
		ode15s	2.2601×10^{-5}	894	0.097
		Opthbm	1.2513×10^{-8}	695	0.077
10^{-4}	10^{-7}	RADAU	9.3148×10^{-9}	2371	0.113
		ode15s	3.3122×10^{-6}	1129	0.118
		Opthbm	9.6196×10^{-10}	1070	0.107

Table 6
Numerical results for Section 5.5.

h_{ini}	Tol	Method	MaxErr	FEvals	C. time
10^{-3}	10^{-5}	RADAU	4.6122×10^{-6}	72	0.014
		ode15s	2.9788×10^{-5}	39	0.018
		Opthbm	5.0900×10^{-8}	30	0.011
10^{-4}	10^{-6}	RADAU	6.0016×10^{-7}	145	0.016
		ode15s	4.8015×10^{-6}	63	0.021
		Opthbm	2.8070×10^{-9}	45	0.014

Here, U and V are positive real constants [44]. The critical point of the differential system is $(Z_1^*, Z_2^*) = (V, U/V)$. In the experiments, we have taken $U = 3,\ V = 1$ and initial values $Z_1^0 = 1.5,\ Z_2^0 = 3$, and the integration interval [0, 20] as it was used by Butcher and Podhaisky [45]. The reference solution at end point $t_N = 20$,

$$Z_1 = 0.49863707126834784833331816235$$
$$Z_2 = 4.59678034945201118264298037 73 \tag{25}$$

is obtained by the NDSolve command with a 10th order Implicit Runge–Kutta method in MATHEMATICA. The problem is solved by considering

$$(h_{ini}, Tol) = (10^{-\kappa}, 10^{-(\kappa+3)}),\ \kappa = 3, 4.$$

The results in Table 5 give a numerical evidence of the good performance of the proposed method.

5.5. Van Der Pol Ssystem

As a last example, consider the following well-known Van der Pol system [36]

$$Z_1' = Z_2, \qquad Z_1(0) = 2$$
$$Z_2' = \frac{(1 - Z_1^2)Z_2 - Z_1}{\epsilon}, \qquad Z_2(0) = \frac{-2}{3} + \frac{10}{81}\epsilon - \frac{292}{2187}\epsilon^2 - \frac{1814}{19683}\epsilon^3. \tag{26}$$

This problem is solved over the integration interval [0, 0.55139] and $\epsilon = 10^{-1}$ as was done in [36]. The numerical results have been obtained by considering

$$(h_{ini}, Tol) = (10^{-\kappa}, 10^{-(\kappa+2)}),\ \kappa = 3, 4.$$

The reference solution $Z_1 = 1.5633739442300918,\ Z_2 = -1.0000208318542727$ at the end of the integration interval is used for numerical experiments. The numerical results in Table 6 support the good performance of the adaptive step-size scheme compared with the other codes considered for comparisons.

6. Conclusions

In this article, we have developed a one-step hybrid block method in optimized version for integrating differential systems. An embedded-type approach is used in order to formulate the proposed scheme in an adaptive step-size mode. Some well-known differential systems are solved in order to illustrate the good performance of the proposed scheme.

Acknowledgments

The authors would like to thank the anonymous reviewers for their work and constructive comments that greatly contributed to improve the manuscript.

References

[1] J.C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley & Sons, 2008.
[2] E. Hairer, S.P. Nörsett, G. Wanner, Solving Ordinary Differential Equations-I, Springer, Berlin, 1993.
[3] E. Hairer, G. Wanner, Solving Ordinary Differential Equations-II, Springer, Berlin, 1996.
[4] L. Brugnano, D. Trigiante, Solving Differential Problems by Multi-Step Initial and Boundary Value Methods, Gordan and Breach Science Publishers, 1998.
[5] S. Natesan, J. JayaKumar, J. Vigo-Aguiar, Parameter uniform numerical method for singularly perturbed turning point problems exhibiting boundary layers, J. Comput. Appl. Math. 158 (1) (2003) 121–134.
[6] J. Vigo-Aguiar, H. Ramos, A new eighth-order $\mathcal{A}$-stable method for solving differential systems arising in chemical reactions, J. Math. Chem. 40 (1) (2006) 71–83.
[7] C.F. Curtiss, J.O. Hirschfelder, Integration of stiff equation, Proc. Natl. Acad. Sci. 38 (1952) 235–243.
[8] J. Vigo-Aguiar, H. Ramos, On the choice of the frequency in trigonometrically-fitted methods for periodic problems, J. Comput. Appl. Math. 277 (2015) 94–105.
[9] J. Vigo-Aguiar, H. Ramos, Variable-step-size Chebyshev-type methods for the integration of second-order I.V.P.'s, J. Comput. Appl. Math. 204 (1) (2007) 102–113.
[10] H. Ramos, J. Vigo-Aguiar, S. Natesan, R. Garc-a Rubio, M.A. Queiruga, Numerical solution of nonlinear singularly perturbed problems on nonuniform meshes by using a non-standard algorithm, J. Math. Chem. 48 (1) (2010) 38–54.
[11] J. Vigo-Aguiar, H. Ramos, A variable-step numerov method for the numerical solution of the Schrödinger equation, J. Math. Chem. 37 (3) (2005) 255–262.
[12] T.E. Simos, J. Vigo-Aguiar, A symmetric high order method with minimal phase-lag for the numerical solution of the Schrödinger equation, I. J. Mod. Phys. 12 (7) (2001) 1035–1042.
[13] T.E. Simos, J. Vigo-Aguiar, An exponentially-fitted high order method for long-term integration of periodic initial-value problems, Comput. Phys. Commun. 140 (3) (2001) 358–365.
[14] H. Ramos, C. Lorenzo, Review of explicit Falkner methods and its modifications for solving special second-order I.V.P.s, Comput. Phys. Commun. 181 (11) (2010) 1833–1841.
[15] H. Ramos, P. Popescu, How many k-step linear block methods exist and which of them is the most efficient and simplest one? Appl. Math. Comput. 316 (2018) 296–309.
[16] H. Ramos, G. Singh, A tenth order a-stable two-step hybrid block method for solving initial value problems of ODEs, Appl. Math. Comput. 310 (2017) 75–88.
[17] H. Ramos, Z. Kalogiratou, T. Monovasilis, T.E. Simos, An optimized two-step hybrid block method for solving general second order initial-value problems, Numer. Alg. 72 (4) (2016) 1089–1102.
[18] H. Ramos, G. Singh, V. Kanwar, S. Bhatia, An efficient variable step-size rational Falkner-type method for solving the special second-order IVP, Appl. Math. Comput. 291 (2016) 39–51.
[19] H. Ramos, M.A. Rufai, Third derivative modification of k-step block Falkner methods for the numerical solution of second order initial-value problems, Appl. Math. Comput. 333 (2018) 231–245.
[20] H. Ramos, S. Mehta, J. Vigo-Aguiar, A unified approach for the development of k-step block Falkner-type methods for solving general second-order initial-value problems in ODEs, J. Comput. Appl. Math. 318 (2017) 550–564.
[21] L. Brugnano, D. Trigiante, Solving Differential Problems by Multi-step Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, 1998.
[22] G. Dahlquist, Convergence and stability in the numerical integration of ordinary differential equations, Math. Scand. 4 (1956) 33–53.
[23] W.E. Milne, Numerical Solution of Differential Equations, John Wiley and Sons, 1953.
[24] D. Sarafyan, Multi-Step Methods for the Numerical Solution of Ordinary Differential Equations Made Self-Starting, Tech. Report 495, Math. Res. Center, Madison, 1995.
[25] J.B. Rosser, A Runge–Kutta for all seasons, SIAM Rev. 9 (1967) 417–452.
[26] L. Brugnano, D. Trigiante, Block implicit methods for ODEs, in: D. Trigiante (Ed.), Recent Trends in Numerical Analysis, Nova Science Publ. Inc., New York, 2001, pp. 81–105.
[27] L. Brugnano, C. Magherini, Recent advances in linear analysis of convergence for splittings for solving ODE problems, Appl. Numer. Math. 28 (1998) 107–126.
[28] F. Iavernaro, F. Mazzia, Solving ordinary differential equations by generalized adams methods: properties and implementation techniques, Appl. Numer. Math. 59 (2009) 542–557.
[29] J.D. Lambert, B. Shaw, On the numerical solution of $y' = f(x, y)$ by a class of formulae based on rational approximation, Math. Comput. 19 (1965) 456–462.
[30] L.F. Shampine, I. Gladwell, S. Thompson, Solving ODEs with MATLAB, Cambridge University Press, 2003.
[31] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 18 (1997) 1–22.
[32] H.A. Watts, Starting step-size for an ODE solver, J. Comput. Appl. Math. 9 (1983) 177–191.
[33] A.E. Sedgwick, An Effective Variable Order Variable Step Adams Method, Dept. of Computer Science. Rept. 53, University of Toronto, Toronto, Canada, 1973.
[34] H.H. Robertson, The Solution of a Set of Reaction Rate Equations, Academic Press, 1966.
[35] A. Glaser, V. Rokhlin, A new class of highly accurate solvers for ordinary differential equations, J. Sci. Comput. 38 (2009) 368–399.
[36] M. Bras, G. Izzo, Z. Jackiewicz, Accurate implicit-explicit general linear methods with inherent Runge–Kutta stability, J. Sci. Comput. (2016), doi:10.1007/s10915-016-0273-y.
[37] F. Mazzia, F. Iavernaro, Test Set for Initial Value Problem Solvers, Report 40/2003.
[38] J.R. Cash, Second derivative extended backward differentiation formulas for the numerical integration of stiff systems, SIAM. J. Numer. Anal. 18 (1) (1981) 21–36.
[39] J.R. Cash, On the integration of stiff systems of ODEs using extended backward differentiation formulae, Numer. Math. 34 (1980) 235–246.
[40] H. Ramos, G. Singh, A note on variable step-size formulation of a Simpson's-type second derivative block method for solving stiff systems, Appl. Math. Lett. 64 (2017) 101–107.
[41] C.W. Gear, Algorithm 407, DIFSUB for solution of ordinary differential equations, Commun. ACM. 14 (1971) 176–179.
[42] J.R. Dormand, P.R. Prince, A family of embedded Runge–Kutta formuale, J. Comput. Appl. Math. 6 (1980) 19–26.
[43] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, 1998.
[44] E.H. Twizell, A.B. Gumel, Q. Cao, A second order scheme for Brusselator reaction diffusion system, J. Math. Chem. 26 (1999) 333–344.
[45] J.C. Butcher, H. Podhaisky, On error estimation in general linear methods for stiff ODEs, Appl. Numer. Math. 56 (2006) 345–357.