Hybrid Block Method for Numerical Solution of First Order Ordinary Differential Equations

Ibrahim Mohammed Dibal[a,*] and Yeak Su Hoe[b]
[a]*Department of General Studies, School of General and Remedial Studies, Federal Polytechnic Damaturu, Yobe State. Nigeria.*
[b]*Department of Mathematical Science, Faculty of Science, Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru. Malaysia.*

Abstract

This research introduces a novel single-step hybrid block method with four intra-step points that attains six-order accuracy, ensures A-stability, consistency, and convergence, and provides an efficient, accurate, and computationally economical tool for solving first-order ordinary differential equations. The formulation incorporates interpolation techniques to approximate function values at points where terms are not explicitly defined on the computational grid. In addition to the construction of the scheme, the paper rigorously investigates its theoretical properties. The results obtained show that the method not only achieves high accuracy but also performs competitively when compared with other established numerical techniques reported in the literature.

Keywords: *Hybrid block method, zero stability, Intra-step points, Initial value problem.*

1. Introduction

Block hybrid methods are numerical schemes that integrate linear multistep techniques with power series representations through interpolation, and were initially introduced by [1]. These methods introduce an extra interpolation point at each step of the formula, thereby increasing the accuracy of differential equation solutions and contributing to improved convergence rates. Since the pioneering work by [2], block methods have attracted significant interest in the literature because of their proven effectiveness in solving both initial value problems (IVPs) and boundary value problems (BVPs), attributed to their flexibility, high accuracy, and computational efficiency in handling complex systems. [3] introduced a hybrid overlapping grid block method that combines equally spaced and optimally chosen grid points for the efficient solution of both linear and nonlinear first-order initial value problems (IVPs). Similarly, [4] applied a block hybrid method with equally spaced grid points to address both linear and nonlinear first-order IVPs, demonstrating convergence rates that surpassed those of the classical fourth-order Runge–Kutta method. [5] developed an efficient five-point hybrid block method, adapted from ODE solvers and enhanced through interpolation to manage piecewise constant delays in first-order differential equations. The method was formulated as a single-

* Corresponding author. Tel.: +2348037334001; fax: +601116426402
E-mail address: mohammeddibal@graduate.utm.my

Manuscript History:
Received 20 September, 2025, Revised 16 October, 2025, Accepted 17 October, 2025, Published 31 October, 2025

https://doi.org/10.33736/jaspe.10889.2025

e-ISSN: 2289-7771

step implicit block scheme with three intra-step grid points, optimized to minimize local truncation error while maintaining low computational cost, as well as ensuring consistency, stability, and convergence. [6]. [7] proposed a hybrid block method in variable step-size mode for the solution of first-order initial value problems of ODEs. These hybrid methods provide notable advantages, such as adaptive step-size control, the use of information from off-step points, and, most importantly, the ability to surpass the Dahlquist barrier of zero stability [8]. Recently, however, a number of researchers, including [9], [10], [11], [12], and [13] have been emphasizing the development of single-step hybrid block methods. Single-step and multi-step numerical methods for solving ordinary differential equations in water tank drainage systems have been compared in terms of accuracy, stability, computational efficiency, and optimal step size [14]. A simple approach for constructing a general class of A-stable explicit second-order one-step methods for stiff problems, inspired by Mickens' nonstandard finite difference methodology, was presented [15]. A fourth-order accurate and efficient predictor–corrector method for solving initial value problems of ordinary differential equations was introduced, developed through integral transformation and numerical quadrature techniques [16]. A predictor-corrector scheme based on a semi-open and closed-Cotes quadrature process for solving initial value problems of ordinary differential equations was also proposed [17]. A third-order, three-stage Trigonometrically-Fitted Improved Runge-Kutta (TFIRK3(3)) method for solving oscillatory ordinary differential equations was developed [18]. Finally, an examination of the Runge-Kutta and Multistep methods for solving initial-value problems highlighted their strengths and limitations, and recommended the development of hybrid methods that combine the properties of both classes [19]. In his work, [20] presented a comprehensive treatment of computational methods in ODEs, emphasizing the practical aspects of stability, convergence, and error control in linear multistep and one-step schemes. [21] focused specifically on a special stability problem for linear multistep methods.

The adoption of single-step hybrid block methods has gained significant attention in numerical analysis due to their balance between simplicity, stability, and computational efficiency. These methods are particularly attractive because they require less derivation effort and produce algorithms that are straightforward to implement on digital computers, thereby minimizing sources of numerical error. Moreover, their strong stability properties resulting from fixed-step discretization allow for the use of smaller step sizes without considerable error amplification from perturbations, making them reliable for solving a wide range of ordinary differential equations (ODEs) [22]. In terms of computational complexity, single-step hybrid block methods generally demand fewer arithmetic operations per step compared to multi-step or predictor–corrector schemes, leading to faster execution times and lower memory usage. This efficiency is especially beneficial when solving large systems or stiff problems where stability and accuracy must be maintained over many iterations. Despite these advantages, ongoing research continues to propose improved block methods aimed at achieving higher-order accuracy and enhanced stability, further reducing computational errors relative to existing techniques. In this context, the present study applies a hybrid block method to the solution of first-order ordinary differential equations of the form;

$$y' = \frac{d}{dt}y(x) = f(x,y), y(0) = y_0.\ x \in [a, b]. \tag{1}$$

Note that $t_0 = a < t_1 < \cdots < t_{N-1} < a$ denotes a set of equally spaced points in the interval $[a, b]$, where $t_j = a + jh,\ j = 0, \dots, N$ and $h = \frac{t_N - t_0}{N}$ is the constant step size. In this study, we review several contributions from the literature, with particular emphasis on numerical methods developed for solving equation (1), focusing particularly on the special class of single-step block methods.

2. Methodology

In this section, we construct an implicit six-point hybrid block method for the numerical solution of Equation (1). The derivation is carried out under the assumption that the differential equation is expressed in the standard form given in Equation (1).

2.1. Specification of the method

The method proposed in this study is a single-step block scheme incorporating four intra-step points, as illustrated in Figure 1.

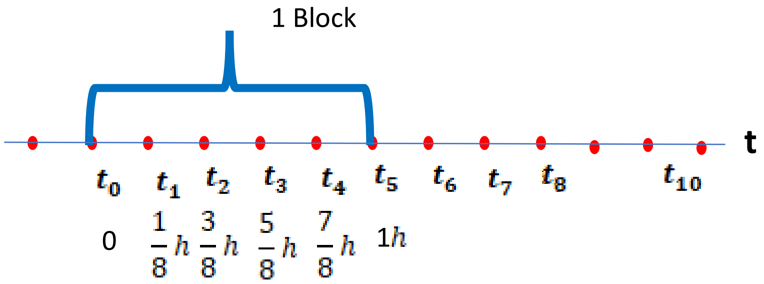

Figure 1. Single-step method with four hybrid points.

2.2. Derivation of the block method

In this section, we present the development of an implicit hybrid block method designed to obtain numerical solutions for Equation (1). To construct the proposed block formulation, we begin by considering the general first-order ordinary differential equation. The hybrid block formulation combines both interpolation and collocation strategies, which help achieve higher accuracy while maintaining numerical stability. By evaluating the function and its derivatives at selected hybrid points within the subinterval, we obtain a system of simultaneous equations representing the implicit block structure. This formulation ultimately leads to a computationally efficient scheme capable of providing accurate and stable solutions for both stiff and non-stiff differential equations, assuming that the exact solution $p(t)$ is;

$$p(t) = \sum_{i=0}^{6} \alpha_i t^i \ . \tag{2}$$

where $\alpha_i \in \mathbb{R}$ denote real-valued unknown coefficients, and

$$p'(t) = \sum_{i=1}^{6} i\alpha_i t^{i-1}. \tag{3}$$

To determine the coefficients α_i, the following conditions are imposed:

$$p(t_n) = f_n \ . \tag{4}$$

$$p'(t_{n+\sigma}) = f_{n+\sigma}, \quad \sigma = 0, r_1, r_2, r_3, r_4, 1. \tag{5}$$

where $0 < r_1 < r_2 < r_3 < r_4 < 1$ and $t_{n+r_i} = t_n + r_i h, \ i = 1, 2, \ldots 4.$

To establish the hybrid collocation points, a set of arbitrary collocation parameters is introduced to define the specific locations within each subinterval where the differential equation is enforced. These parameters provide flexibility in determining the optimal distribution of points that enhance the accuracy and stability of the resulting block method. To construct the hybrid collocation points, we introduce arbitrary collocation parameters. Evaluating Equation (2) at $t_n$and Equation (2) at points, $[t_{n+r_i} = t_n + r_i h]$, $i = 1, 2, \ldots 4$, we obtain the following system:

$$\begin{bmatrix} 1 & t_n & t_n^2 & t_n^3 & t_n^4 & t_n^5 & t_n^6 \\ 0 & 1 & 2t_n & 3t_n^2 & 4t_n^3 & 5t_n^4 & 6t_n^5 \\ 0 & 1 & 2t_{n+r_1} & 3t_{n+r_1}^2 & 4t_{n+r_1}^3 & 5t_{n+r_1}^4 & 6t_{n+r_1}^5 \\ 0 & 1 & 2t_{n+r_2} & 3t_{n+r_2}^2 & 4t_{n+r_2}^3 & 5t_{n+r_2}^4 & 6t_{n+r_2}^5 \\ 0 & 1 & 2t_{n+r_3} & 3t_{n+r_3}^2 & 4t_{n+r_3}^3 & 5t_{n+r_3}^4 & 6t_{n+r_3}^5 \\ 0 & 1 & 2t_{n+r_4} & 3t_{n+r_4}^2 & 4t_{n+r_4}^3 & 5t_{n+r_4}^4 & 6t_{n+r_4}^5 \\ 0 & 1 & 2t_{n+1} & 3t_{n+1}^2 & 4t_{n+1}^3 & 5t_{n+1}^4 & 6t_{n+1}^5 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \begin{bmatrix} y_n \\ f_n \\ f_{n+r_1} \\ f_{n+r_2} \\ f_{n+r_3} \\ f_{n+r_4} \\ f_{n+1} \end{bmatrix} \text{ and}$$

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} \begin{bmatrix} y_n \\ f_n \\ f_{n+r_1} \\ f_{n+r_2} \\ f_{n+r_3} \\ f_{n+r_4} \\ f_{n+1} \end{bmatrix} = \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} B \rightarrow a_i = R_i(t_n, h)B, \;\; B = \begin{bmatrix} y_n \\ f_n \\ f_{n+r_1} \\ f_{n+r_2} \\ f_{n+r_3} \\ f_{n+r_4} \\ f_{n+1} \end{bmatrix} \text{ and } [R_i(t_n, h)]_{i=0\ldots4} = \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} \quad (6)$$

This resulting system of equations must be solved to determine the unknown coefficients α_j, j = 0, 1, …, 4, which define the structure of the hybrid block method. Once these coefficients are obtained, they are substituted back into Equation (2) to construct the complete numerical formulation of the method. By introducing an appropriate change of variable, $t = t_n + \sigma h$ this simplifies the representation of the local solution within the subinterval, the approximate solution $p(t_n + \sigma h)$ can be expressed in a compact analytical form as;

$$p(t_n + \lambda h) = C_0 y_n + h\big(C_n(t) f_n + C_{r_1}(t) f_{n+r_1} + C_{r_2}(t) f_{n+r_2} + C_{r_3}(t) f_{n+r_3} + C_{r_4}(t) f_{n+r_4} + +C_1(t) f_{n+1}\big) \quad (7)$$

$$p(t) = \sum_{i=0}^{6} a_i t^i = \sum_{i=0}^{6} R_i(t_n, h) B t^i = \sum_{i=0}^{6} t^i R_i(t_n, h) \begin{bmatrix} y_n \\ f_n \\ f_{n+r_1} \\ f_{n+r_2} \\ f_{n+r_3} \\ f_{n+r_4} \\ f_{n+1} \end{bmatrix}$$

Finally, the resulting form of Equation (3) is obtained as follows:

$$p(t_n + xh) = \sum_{i=0}^{6} \alpha_i (t_n + xh)^i = \sum_{i=0}^{6} R_i(t_n, h) B(t_n + xh)^i = \sum_{i=0}^{6} (t_n xh)^i R_i(t_n, h) \begin{bmatrix} y_n \\ f_n \\ f_{n+r_1} \\ f_{n+r_2} \\ f_{n+r_3} \\ f_{n+r_4} \\ f_{n+1} \end{bmatrix}$$ and

$$p(t_n + xh) = C_0 y_n + h\left(C_1 f_n + C_2 f_{n+r_1} + C_3 f_{n+r_2} + C_4 f_{n+r_3} + C_5 f_{n+r_4} + C_6 f_{n+1}\right)$$ where

$$C_0 = \sum_{i=0}^{6} (t_n + xh)^i R_{i,0}(t_n, h), \ \ hC_k = \sum_{i=0}^{6} (t_n + xh)^i R_{i,k}(t_n, h), \ \ k = 1,2,\dots 4.$$

With

$$\begin{cases} C_0 = 1 \\ C_n = -\dfrac{1}{3150} h \left(\begin{matrix} 20480h^5 - 73728h^4 + 102720h^3 - 69120h^2 + \\ 22695h - 3150 \end{matrix} \right) \\ C_{r_1} = \dfrac{2}{315} h^2 (2560h^4 - 8832h^3 + 11460h^2 - 6730h + 1575) \\ C_{r_2} = -\dfrac{2}{225} h^2 (2560h^4 - 8064h^3 + 9060h^2 - 4110h + 525) \\ C_{r_3} = \dfrac{2}{225} h^2 (2560h^4 - 7296h^3 + 7140h^2 - 2690h + 315) \\ C_{r_4} = -\dfrac{2}{315} h^2 (2560h^4 - 6528h^3 + 5700h^2 - 1990h + 225) \\ C_{r_1} = \dfrac{32}{3150} h^2 (20480h^4 - 49152h^3 + 41280h^2 - 14080h + 1575) \end{cases}.$$

To derive the main form of the hybrid block method, the polynomial approximation $p(t_n + xh)$ is evaluated at selected hybrid points corresponding to $x = r_1, r_2, r_3, r_4, 1$. These evaluation points are carefully chosen within the normalized interval $[0, 1]$ to balance accuracy, stability, and computational efficiency. Substituting these values into the polynomial expression produces a system of simultaneous equations that collectively defines the required block structure of the method as follows;

$$\begin{cases} y_{n+r_1} = y_n + h\left(\frac{9679}{201600}f_n + \frac{14339}{161280}f_{n+r_1} - \frac{2203}{115200}f_{n+r_2} + \frac{409}{38400}f_{n+r_3} - \frac{851}{161280}f_{n+r_4} + \frac{41}{22400}f_{n+1}\right) \\ y_{n+r_2} = y_n + h\left(\frac{663}{22400}f_n + \frac{3963}{17920}f_{n+r_1} + \frac{1869}{12800}f_{n+r_2} - \frac{381}{12800}f_{n+r_3} + \frac{213}{17920}f_{n+r_4} - \frac{87}{22400}f_{n+1}\right) \\ y_{n+r_3} = y_n + h\left(\frac{295}{8064}f_n + \frac{2125}{10752}f_{n+r_1} + \frac{1325}{4608}f_{n+r_2} + \frac{515}{4608}f_{n+r_3} - \frac{125}{10752}f_{n+r_4} + \frac{25}{8064}f_{n+1}\right) \\ y_{n+r_4} = y_n + h\left(\frac{889}{28800}f_n + \frac{4949}{23040}f_{n+r_1} + \frac{28469}{115200}f_{n+r_2} + \frac{10633}{38400}f_{n+r_3} + \frac{2779}{23040}f_{n+r_4} - \frac{49}{3200}f_{n+1}\right) \\ y_{n+1} = y_n + h\left(\frac{103}{3150}f_n + \frac{22}{105}f_{n+r_1} + \frac{58}{225}f_{n+r_2} + \frac{58}{225}f_{n+r_3} + \frac{22}{105}f_{n+r_4} + \frac{103}{3150}f_{n+1}\right) \end{cases} \tag{8}$$

The implicit Equation (8) can be solved using two approaches. The first approach applies fixed-point iteration, provided the convergence condition is satisfied.

Let $Y_n = \begin{bmatrix} y_{n+r_1} \\ y_{n+r_2} \\ y_{n+r_3} \\ y_{n+r_4} \\ y_{n+1} \end{bmatrix}$, we get $Y_n = \begin{bmatrix} y_{n+r_1} \\ y_{n+r_2} \\ y_{n+r_3} \\ y_{n+r_4} \\ y_{n+1} \end{bmatrix} = G(Y) = \begin{bmatrix} y_n \\ y_n \\ y_n \\ y_n \\ y_n \end{bmatrix} + \begin{bmatrix} c_1 f_n \\ c_2 f_n \\ c_3 f_n \\ c_4 f_n \\ c_5 f_n \end{bmatrix} + \begin{bmatrix} g_1(Y_n) \\ g_2(Y_n) \\ g_3(Y_n) \\ g_4(Y_n) \\ g_5(Y_n) \end{bmatrix}$

The second approach employs Jacobi iteration, which proceeds as follows: starting with the initial guess Y_n^0, we obtain:

$$Y_n^{k+1} = G\left(Y_n^k\right).$$

The final solution is obtained as Y_n^K subject to the stopping criterion $|Y_n^K - Y_n^{K-1}| < \epsilon$. By applying the step-size transformation $t_{n+r_i} = t_n + r_i h,\ i = 1, 2, \dots 4$ and setting $r_i K = R_i$, we obtain the following transformation with $K = 5$:

$$\left[r_1 = \frac{1}{8}, \text{j } r_2 = \frac{3}{8}, \quad r_3 = \frac{5}{8}, r_4 = \frac{7}{8}\right] \to K[r_1, r_2, r_3, r_4] = 5[r_1, r_2, r_3, r_4] = [R_1, R_2, R_3, R_4] = [1,2,3,4]$$

Let $h = KH, y' = \frac{d}{dt}y = f(t, y), y_n = y(t_n)$.

$$y(t_n + rh) = y(t_n + rKH) = y_{n+rK} = y_{n+R}, \qquad rK = R \in z.$$

For $\frac{d}{dt}y = f(t, y)$, we get

$$y_{n+R_1} = y_n + h\left(\frac{9679}{201600}f_n + \frac{14339}{161280}f_{n+R_1} - \frac{2203}{115200}f_{n+R_2} + \frac{409}{38400}f_{n+R_3} - \frac{851}{161280}f_{n+R_4} + \frac{41}{22400}f_{n+1}\right)$$

Or

$$\alpha_{R_1} y_{n+R_1} + \alpha_0 y_n = KH\left(\beta_0 f_n + \beta_{R_1} f_{n+R_1} + \beta_{R_2} f_{n+R_2} + \beta_{R_3} f_{n+R_3} + \beta_{R_4} f_{n+R_4} + \beta_1 f_{n+K}\right).$$

And we get

$$\alpha_{R_1} y_{n+R_1} + \alpha_0 y_n = KH\left(\beta_0 f_n + \beta_{r_1} f_{n+R_1} + \beta_{R_2} f_{n+R_2} + \beta_{R_3} f_{n+R_3} + \beta_{R_4} f_{n+R_4} + \beta_1 f_{n+K}\right).$$

The truncation error is:

$$C_0 = \alpha_0 + \alpha_1 + \cdots + \alpha_j,$$

$$C_1 = \alpha_1 + 2\alpha_2 + \cdots + k\alpha_j - (\beta_0 + \beta_1 + \cdots + \beta_j),$$

$$\vdots$$

$$C_q = \frac{1}{q!}(\alpha_1 + 2^q\alpha_2 + \cdots + k^q\alpha_j) - \frac{1}{(q-1)!}(\beta_1 + 2^{q-1}\beta_2 + \cdots + k^{q-1}\beta_j), \qquad q \geq 2.$$

Hence, the truncation error is expressed as $C_q H^q y^{(q)}(t_n)$.

$$H = \frac{1}{K}h \rightarrow C_q H^q y^{(q)}(t_n) = \frac{C_q}{K^q} h^q y^{(q)}(t_n).$$

Obviously, we find that $y_{n+R_i} = y_{n+i}, i = 1, \cdots, 4$.

Thus

$$y_{n+5} = y_n + KH\left[C_{R,1}f_n + C_{R,2}f_{n+1} + C_{R,3}f_{n+2} + C_{R,4}f_{n+3} + C_{R,5}f_{n+4}\right].$$

where $\alpha_0 = -1, \alpha_5 = 1, \beta_j = C_{k,j+1}$.

The linear multistep method is obtained in the form:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = KH \sum_{j=0}^{k} \beta_j f_{n+j}.$$

By using $\beta^* = K\beta = 5\beta$ we get:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = KH \sum_{j=0}^{k} \beta_j f_{n+j} = H \sum_{j=0}^{k} \beta_j^* f_{n+j}.$$

$$\alpha_{R_1} y_{n+R_1} + \alpha_0 y_n = H\left(\beta_0^* f_n + \beta_{r_1}^* f_{n+R_1} + \beta_{R_2}^* f_{n+R_2} + \beta_{R_3}^* f_{n+R_3} + \beta_{R_4}^* f_{n+R_4} + \beta_{R_5}^* f_{n+R_5} + \beta_{R_6}^* f_{n+R_6} + \beta_1^* f_{n+K}\right)$$

2.3. Order and error constant of the method

In this section, we investigate the fundamental properties of Equation (8) to establish the theoretical foundation of the proposed numerical method. The analysis focuses primarily on determining the order of accuracy, which reflects how closely the numerical approximation matches the exact analytical solution. To achieve this, we introduce the concept of the local truncation error, which quantifies the error committed in a single computational step of the method. Furthermore, we make use of linear difference operators to express the relationship between successive numerical approximations in a compact and systematic form. By applying these operators, the algebraic structure of the method becomes clearer, allowing for a more rigorous analysis of consistency and convergence. To evaluate the local truncation error explicitly, we expand the exact solution $y(t)$ about the point t_n using Taylor's series, retaining terms up to the desired order. This expansion provides the basis for comparing the analytical and numerical representations, thereby enabling the determination of the method's order and overall accuracy. By the linear difference operator, we obtain:

$$L_{\sigma}[y(t_n),h] = y(t_n+\sigma h) - h\sum_j \beta_{j,\sigma} y'(t_n+jh),\ \ \sigma = r_1, r_2, r_3, r_4, \mathrm{K}\ .$$

where $\beta_{j,\sigma}$ are the corresponding coefficients. We expand:

$y(t_n+\sigma h)\ and\ \ y'(t_n+jh),\ j = R_1, R, R_3, R_4, \mathrm{k}$ about t_n using Taylor's series as;

$$l[y(t);h] = C_0 y(t) + C_1 h y^{(1)}(t) + C_2 h^2 y^{(2)}(t) + \cdots + C_p h^p y^{(p)}(t) + \cdots$$

Where $C_j, j = 0,1,\ldots$ are vectors. Hence $C_0 = C_1 = \cdots = C_6 = 0,\ C_{p+1} = C_7 \neq 0$, so C_7 is the error constant of the method. After collecting the coefficients of h, the expressions for the local truncation error corresponding to each formula are derived as follows:

$$L_{R_1}[y(t_n),h] = -\frac{1217}{19818086400} h^6 y^{(6)}(t_n) + O(h^7),$$

$$L_{R_2}[y(t_n),h] = \frac{67}{734003200} h^6 y^{(6)}(t_n) + O(h^7),$$

$$L_{R_3}[y(t_n),h] = -\frac{25}{792723456} h^6 y^{(6)}(t_n) + O(h^7),$$

$$L_{R_4}[y(t_n),h] = \frac{343}{2831155200} h^6 y^{(6)}(t_n) + O(h^7),$$

$$L_1[y(t_n),h] = \frac{37}{619315200} h^7 y^{(7)}(t_n) + O(h^8),$$

Consequently, from the above results, the order of the newly developed method is six, and the corresponding vector of error constants is given by:

$$C_7 = \left(-\frac{1217}{19818086400}, \frac{67}{734003200}, -\frac{25}{792723456}, \frac{343}{2831155200}, \frac{37}{619315200}\right)^T.$$

2.4. Consistency

Definition 2.1: According to [20], a linear multistep method (LMM) is consistent if its order satisfies $p > 1$. Meaning $p > 1$, this ensures that the local truncation error tends to zero as the step $h \to 0$. Since the proposed hybrid block method possesses an order of $p = 6 > 1$, it satisfies this fundamental criterion of consistency. Therefore, it can be concluded that the newly developed method is indeed consistent. This property confirms that the method provides an accurate approximation of the exact solution, forming a key requirement for overall convergence.

2.5 Zero stability

Definition 2.2. A linear multistep method is said to be zero-stable if no root of the first characteristic polynomial $p(z)$ has modulus greater than one. Moreover, it is simple if all roots lie on the unit circle. For convergence, a linear multistep method must be both zero-stable and consistent [20]. Since the proposed block method satisfies $|z_j| \leq 1$, it is therefore zero-stable.

According to Definition 2.2, a linear multistep method is considered zero-stable if none of the roots of its first characteristic polynomial have a modulus greater than one. This condition ensures that errors introduced at one step of the computation do not grow uncontrollably as the numerical process proceeds. Moreover, the method is said to be simple if all roots of the characteristic polynomial lie exactly on the unit circle, indicating that the numerical solution remains bounded over successive iterations. Zero-stability is a crucial requirement for ensuring the reliability of any linear multistep method, as instability can lead to divergence even when the method is otherwise accurate. In this study, the proposed block method has been verified to satisfy the necessary root condition of the characteristic polynomial as given below.

For the stability analysis, the hybrid block method in equation (7) is reformulated into the following matrix form:

$$A^1 Y_{n+1} = A^0 Y_n + h(B^0 F_n - B^1 F_{n+1}), \tag{9}$$

Note that

$$Y_{n+1} = \left[y_{n+R_1}, y_{n+R_2}, y_{n+R_3}, y_{n+R_4}, y_{n+1}\right]^T,$$

$$Y_n = \left[y_{n+R_1-1}, y_{n+R_2-1}, y_{n+R_3-1}, y_{n+R_4-1}, y_n\right]^T,$$

$$F_{n+1} = \left[f_{n+R_1}, f_{n+R_2}, f_{n+R_3}, f_{n+R_4}, f_{n-1}\right]^T,$$

$$F_n = \left[f_{n+R_1-1}, f_{n+R_2-1}, f_{n+R_3-1}, f_{n+R_4-1}, f_n\right]^T.$$

where

$$A^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad A^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B^1 = \begin{bmatrix} \frac{14339}{161280} & -\frac{2203}{115200} & \frac{409}{38400} & -\frac{851}{161280} & \frac{41}{22400} \\ \frac{3963}{17920} & \frac{1869}{12800} & -\frac{381}{12800} & \frac{213}{17920} & -\frac{87}{22400} \\ \frac{2125}{10752} & \frac{1325}{4608} & \frac{515}{4608} & -\frac{125}{10752} & \frac{25}{8064} \\ \frac{4949}{23040} & \frac{28469}{115200} & \frac{10633}{38400} & \frac{2779}{23040} & -\frac{49}{3200} \\ \frac{22}{105} & \frac{58}{225} & \frac{58}{225} & \frac{22}{105} & \frac{103}{3150} \end{bmatrix}$$

$$B^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{9679}{201600} \\ 0 & 0 & 0 & 0 & \frac{663}{22400} \\ 0 & 0 & 0 & 0 & \frac{295}{8064} \\ 0 & 0 & 0 & 0 & \frac{889}{28800} \\ 0 & 0 & 0 & 0 & \frac{103}{3150} \end{bmatrix}$$

$$f_{n+R_1} = f(t_n + z\Delta t, y(t_n + z\Delta t)) = f(t_{n+R_1}, y_{n+R_1}), y' = f(t, y),$$

$$A^1 Y_{n+1} = A^0 Y_n + h(B^0 F_n + B^1 F_{n+1}),$$

Let $F_n = F_{n+1} = 0,$

$$A^1 Y_{n+1} = A^0 Y_n + h(B^0 F_n + B^1 F_{n+1}) \rightarrow A^1 Y_{n+1} - A^0 Y_n = 0,$$

Let $y_i = z^i$, we get;

$$Y_{n+1} = [y_{n+R_1}, y_{n+R_2}, y_{n+R_3}, y_{n+R_4}, y_{n+1}]^T = [z^{n+R_1}, z^{n+R_2}, z^{n+R_3}, z^{n+R_4}, z^{n+1}]^T,$$

$$A^1 z Y_n - A^0 Y_n = 0 \rightarrow (A^1 z - A^0) Y_n = 0,$$

To obtain a nontrivial solution for Y_n, it is required that $(A^1 z - A^0)$ be non-singular. Hence, we impose $det(A^1 z - A^0) = 0$. We then obtain:

$$det(A^1 z - A^0) = z \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= z^4(z-1) = 0 \rightarrow z_i = (0, 0, 0, 0, 1).$$

To analyse linear stability, we apply the method to the standard test problem:

$$y' = f(t, y) = \tau y, \quad Re(\tau) < 0 \ . \qquad (10)$$

Let

$$F_{n+1} = [f_{n+R_1}, f_{n+R_2}, f_{n+R_3}, f_{n+R_4}, f_{n+1}]^T = [\tau y_{n+R_1}, \tau y_{n+R_2}, \tau y_{n+R_3}, \tau y_{n+R_4}, \tau y_{n+1}]^T$$
$$= \tau [y_{n+R_1}, y_{n+R_2}, y_{n+R_3}, y_{n+R_4}, y_{n+1}]^T = \tau y_{n+1}$$

Substituting

$y' = \tau y$ in $A^1 Y_{n+1} = A^0 Y_n + h(B^0 F_n + B^1 F_{n+1}).$

we obtain

$$A^1Y_{n+1} = A^0Y_n + h(B^0F_n + B^1F_{n+1}) \rightarrow A^1Y_{n+1} = A^0Y_n + h(B^0\tau Y_n + B^1\tau Y_{n+1}),$$

$$A^1Y_{n+1} - hB^1\tau Y_{n+1} = A^0Y_n + hB^0\tau Y_n \rightarrow (A^1 - hB^1\tau)Y_{n+1} = (A^0 + hB^0\tau)Y_n,$$

$$\rightarrow Y_{n+1} = (A^1 - hB^1\tau)^{-1}(A^0 + hB^0\tau)Y_n,$$

But $\hat{h} = \tau h$

$$\rightarrow Y_{n+1} = (A^1 - \hat{h}B^1)^{-1}(A^0 + \hat{h}B^0)Y_n,$$

And $Y_{n+1} = M(\hat{h})Y_n$,

$$\rightarrow M(\hat{h})Y_n = (A^1 - \hat{h}B^1)^{-1}(A^0 + \hat{h}B^0)Y_n,$$

$$\rightarrow M(\hat{h}) = (A^1 - \hat{h}B^1)^{-1}(A^0 + \hat{h}B^0),$$

The behavior of the numerical solution Y_{n+1} is governed by the eigenvalues of.$M(\hat{h})$. The stability matrix $M(\hat{h})$ admits eigenvalues $\{0, 0, 0, 0, \psi(\hat{h})\}$ where the dominant eigenvalue $\psi(\hat{h})$ is given in the following rational form:

$$\psi(\hat{h}) = \frac{R(\hat{h})}{Q(\hat{h})}.$$

$$R(\hat{h}) = -105h^5 - 3026h^4 - 41472h^3 - 328704h^2 - 1474560h - 2949120.$$

$$Q(\hat{h}) = 105h^5 - 3026h^4 + 41472h^3 - 328704h^2 + 1474560h - 2949120.$$

Let $Y_{n+1} = M(\hat{h})Y_n = MY_n$ then we obtain

$$Y_1 = MY_0,$$

$$Y_2 = MY_1 = M(MY_0) = M^2Y_0,$$

Finally, we get;

$$Y_j = M^jY_0,$$

Therefore, the numerical solution is stable provided that

$$\lim_{j\rightarrow\infty} Y_j = 0 \rightarrow \lim_{j\rightarrow\infty} M^j = 0,$$

Alternatively, it can be expressed in norm form as follows:

$$||M|| < 1.$$

2.6. Convergence of the method

Theorem 1: [8] Consistency and zero stability are sufficient conditions for a linear multi-step method to be convergent. Since consistency and zero-stability of the proposed method have been established, the newly developed method (8) is convergent.

Consistency and zero-stability are the two fundamental requirements for ensuring that a linear multistep method converges to the true solution of an ordinary differential equation. Consistency means that the local truncation error of the method tends to zero as the step size decreases, ensuring the numerical formula accurately represents the original differential equation**.** Zero-stability ensures that errors introduced at any step do not grow uncontrollably as computations proceed, maintaining the stability of the numerical solution. When both conditions are satisfied, the method is convergent, meaning the numerical solution approaches the exact analytical solution as the step size tends to zero.

2.7 Stability region

Definition 2.3: Following [21], a block hybrid method is said to be A-stable if the entire left-half of the complex plane lies within its region of absolute stability. Based on this definition, the proposed block method is A-stable, as illustrated in Figure 2.

The concept of A-stability is crucial in the analysis of numerical methods for solving stiff ordinary differential equations (ODEs). According to Definition 2.3, a block hybrid method is said to be A-stable if its region of absolute stability completely contains the left-half of the complex plane**,** meaning that for any test equation of the form:

$$y' = f(t, y) = \tau y, \quad Re(\tau) < 0$$

The numerical solution remains stable regardless of the step size h. In other words, the method does not produce growing numerical oscillations or instabilities when applied to problems where the exact solution decays over time.

In the context of the proposed block hybrid method, the A-stability property ensures that the method can handle stiff equations effectively, allowing for larger step sizes without sacrificing numerical stability. This is particularly important for systems where rapid decay or oscillatory behavior occurs, as explicit methods typically fail or require excessively small steps in such cases. The stability region, often depicted in the complex plane, shows where the magnitude of the amplification factor $|R(z)| < 1$with $z = h\lambda$.

As illustrated in Figure 2, the stability region of the proposed method encompasses the entire left-half of the complex plane, confirming that it satisfies the A-stability condition. This means that for all z with negative real parts, the numerical solution remains bounded and converges to the true solution. Consequently, the proposed block method is highly reliable for stiff initial value problems, providing both stability and accuracy over a wide range of step sizes.

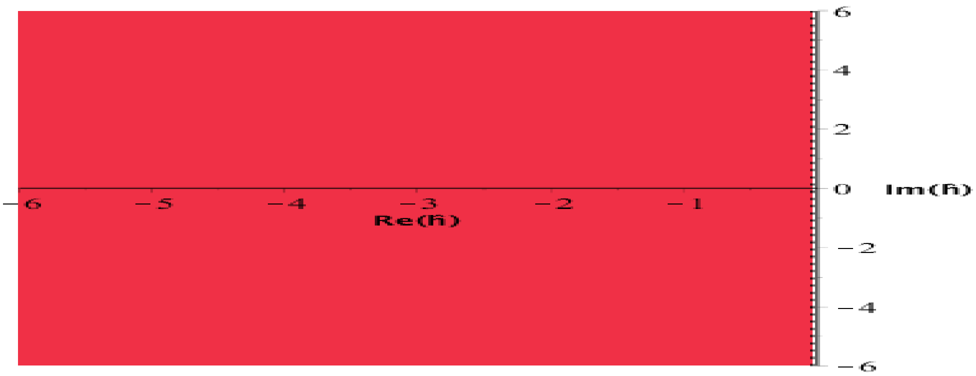

Figure 2. Region of absolute stability

3. Numerical results

In this section, the performance of the proposed method is thoroughly evaluated by applying it to a set of first-order ordinary differential equations comprising both stiff and non-stiff problems. These examples are carefully selected to test the method's reliability, stability, and adaptability across a wide range of problem characteristics. The numerical experiments are implemented using MATLAB to ensure precision and reproducibility of results. For comparison purposes, the well-established MATLAB solver ode45 is employed as a benchmark, allowing for an objective assessment of the proposed method's performance. The accuracy of the computed solutions is examined through absolute error measurements, while computational efficiency is evaluated in terms of execution time. The results obtained are organized and presented in a series of tables for clarity and ease of interpretation.

Table 1: The list of test problems along with their initial condition, integration interval, and exact solution, respectively

S/N	Problem	Initial condition	Interval	Exact solution	Source
1	Linear stiff equation $y_1' = -y_1 + 10y_2$ $y' = 10y_1 - y_2$	$y_1(0) = 1$ $y_2(0) = 0$	$0 \le t \le 10$	$y_1(t) = e^{-t}\cos(10t)$ $y_2(t) = e^{-t}\sin(10t)$	[23]
2	Nonlinear differential equation $y' = 1 - y^2$	$y(0) = 2$	$0 \le t \le 1$	$y(t) = \frac{e^{2t} - 1}{e^{2t} + 1}$	[24]
3	Nonlinear stiff equation $y_1' = -12y_1 + 10y_2^2$ $y_2' = y_1 - y_2 - y_1y_2$	$y_1(0) = 1$ $y_2(0) = 1$	$0 \le t \le 1$	$y_1(t) = e^{-2t}$ $y_2(t) = e^{-t}$	[25]
4	Stiff differential equation $y_1' = y_3 - cost$ $y_2' = y_3 - e^t$ $y_3' = y_1 + y_2$	$y_1(0) = 1$ $y_2(0) = 0$	$0 \le t \le 1$	$y_1(t) = e^t$ $y_2(t) = sint$	[26]

		$y_3(0) = 2$		$y_3(t) = e^t + cost$	
5	Stiff differential equation $y_1' = -2y_1 + y_2$ $y_2' = -3y_1 + 2y_2 - cost + sint$	$y_1(0) = 2$ $y_2(0) = 3$	$0 \leq t \leq 1$	$y_1(t) = 2e^{-t} + sint$ $y_2(t) = 2e^{-t} + cost$	[26]
6	Nonlinear stiff chemical reaction problem of Robertson: $y_1' = -0.04y_1 + 10^4 y_2 y_3,$ $y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2,$ $y_3' = 3 \times 10^7 y_2^2$	$y_1(0) = 1$ $y_2(0) = 0$ $y_3(0) = 0$	$0 \leq t \leq 40$	$y_1(t) = 0.7158270687194135$ $y_2(t) = 9.185534764558135 \times 10^{-6}$ $y_3(t) = 0.28416374574582$	[27]
7	The SIR model is an epidemiological model that computes the theoretical number of people infected with a contagious illness in a closed population over time. The name of this class of models derives from the fact that they involve coupled equations relating the number of susceptible people *S(t)*, number of people infected *I(t) and* the number of people who have recovered *R(t)*. Defined as: $Y = S + I + R$, the evolution equation is $y'(t) = \mu(1 - y) = 0$ with $\mu = \frac{1}{2},$	$y(0) = \frac{1}{2}$	$0 \leq t \leq 1$	$y(t) = 1 - 0.5e^{-0.5t}$	[7]

Table 2: Comparison between the absolute error and CPU time for problem 1 at various values of h

h	Method	Absolute Error	CPU time
10^{-2}	Proposed method	$6.5023e^{-07}$	0.1053
	[5]	$1.5406e^{-06}$	0.1241
	[6]	$5.1953e^{-06}$	0.1023
	[7]	$2.6613e^{-06}$	0.1044
10^{-3}	Proposed method	$6.5096e^{-10}$	0.0944
	[5]	$1.5430e^{-09}$	0.0970
	[6]	$5.2070e^{-09}$	0.0849
	[7]	$2.6661e^{-09}$	0.0960
10^{-4}	Proposed method	$6.5103e^{-13}$	0.0863
	[5]	$1.5432e^{-12}$	0.0904
	[6]	$5.2082e^{-12}$	0.0908
	[7]	$2.6666e^{-12}$	0.0893

Table 3: Comparison between the absolute error and CPU time for problem 2 at various values of h

h	Method	Absolute Error	CPU time
10^{-2}	Proposed method	$5.2042e^{-18}$	0.073253
	[5]	$3.4044e^{-17}$	0.088921
	[6]	$5.2909e^{-17}$	0.081897
	[7]	$4.5699e^{-17}$	0.087229
10^{-4}	Proposed method	$9.7578e^{-19}$	0.072765
	[5]	$5.7463e^{-18}$	0.087937
	[6]	$6.5052e^{-18}$	0.080570
	[7]	$7.4810e^{-18}$	0.081080
10^{-5}	Proposed method	$5.7598e^{-19}$	0.069652
	[5]	$8.0618e^{-18}$	0.085403
	[6]	$6.4781e^{-18}$	0.083292
	[7]	$9.6697e^{-18}$	0.080763

Table 4: Comparison between the absolute error and CPU time for problem 3 at various values of h

h	Method	Absolute Error	CPU time
10^{-2}	Proposed method	$6.4658e^{-09}$	0.1371
	[5]	$1.5291e^{-08}$	0.2311
	[6]	$5.1373e^{-08}$	0.1577
	[7]	$2.6375e^{-08}$	0.1630
10^{-3}	Proposed method	$6.5060e^{-12}$	0.1008
	[5]	$1.5418e^{-11}$	0.1725
	[6]	$5.2012e^{-11}$	0.1247
	[7]	$2.6637e^{-11}$	0.1288
10^{-4}	Proposed method	$6.6613e^{-15}$	0.0765
	[5]	$1.5321e^{-14}$	0.0875
	[6]	$5.2180e^{-14}$	0.0934
	[7]	$2.6645e^{-14}$	0.0909

Table 5: Comparison between the absolute error and CPU time for problem 4 at various values of h

h	Method	Absolute Error	CPU time
10^{-2}	Proposed method	$6.5104e^{-10}$	0.0856
	[5]	$1.5432e^{-09}$	0.1140
	[6]	$5.2083e^{-09}$	0.0862
	[7]	$2.6667e^{-09}$	0.0965
10^{-3}	Proposed method	$6.5103e^{-13}$	0.0918
	[5]	$1.5437e^{-12}$	0.0944
	[6]	$5.2081e^{-12}$	0.0939
	[7]	$2.6665e^{-12}$	0.0949
10^{-4}	Proposed method	$6.6613e^{-16}$	0.0906
	[5]	$1.3323e^{-15}$	0.0921
	[6]	$5.1070e^{-15}$	0.0916
	[7]	$2.8866e^{-15}$	0.0940

Table 6: Comparison between the absolute error and CPU time for problem 5 at various values of h

h	Method	Absolute Error	CPU time
10^{-2}	Proposed method	$7.8027e^{-07}$	0.090094
	[5]	$1.3866e^{-06}$	0.093049
	[6]	$3.1172e^{-06}$	0.095068
	[7]	$1.9960e^{-06}$	0.092819
10^{-3}	Proposed method	$7.8115e^{-09}$	0.081289
	[5]	$1.3887e^{-08}$	0.093814
	[6]	$3.1242e^{-08}$	0.086677
	[7]	$1.9996e^{-08}$	0.088596
10^{-4}	Proposed method	$7.8124e^{-11}$	0.076882
	[5]	$1.3889e^{-10}$	0.093780
	[6]	$3.1249e^{-10}$	0.085224
	[7]	$2.0000e^{-10}$	0.091410

Table 7: Comparison between the absolute error and CPU time for problem 6 at various values of h

h	Method	Absolute Error	CPU time
10^{-3}	Proposed method	$3.2474e^{-12}$	0.0874
	[5]	$1.0207e^{-11}$	0.1559
	[6]	$2.8477e^{-11}$	0.0946
	[7]	$1.7861e^{-11}$	0.1500
2.0×10^{-4}	Proposed method	$9.3547e^{-13}$	0.0873
	[5]	$2.0065e^{-12}$	0.1269
	[6]	$2.0553e^{-12}$	0.1085
	[7]	$2.1024e^{-12}$	0.1100

Table 8: Comparison between our method and other established methods for problem 7.

h	Method	Absolute Error	CPU time
10^{-1}	NANNM [28]	$7.379e^{-06}$	0.1466
	OHBM5A [29]	$7.071068e^{-02}$	0.1423
	Proposed HBM	$4.5219e^{-13}$	0.0843
10^{-2}	NANNM [28]	$7.757e^{-08}$	0.0464
	OHBM5A [29]	$2.3895e^{-03}$	0.0585
	Proposed HBM	$3.3307e^{-16}$	0.0284

Abbreviations

h	Step Size,
CPU	Central Processing Unit,
HBM	Proposed Hybrid Block Method,
NANNM	New Adaptive Nonlinear Numerical Method,
OHBM5A	Optimized Hybrid Block Method with fifth-order, Adaptive and fixed step-size.

The figures presented below provide a graphical illustration of the maximum absolute errors obtained for Test Problems 1–7 using the proposed hybrid block method. These visualizations serve to demonstrate the accuracy and reliability of the developed numerical scheme when applied to different types of ordinary differential equations. Figures 3-7 display the maximum absolute error distribution for Problems 1-5 at a step size of $h = 0.01$, revealing that the error remains minimal and uniformly distributed across the integration intervals. Figure 8 presents the corresponding results for Problem 6 at a step size of $h = 0.001$, where the method maintains excellent numerical stability with only slight variations in error magnitude. Similarly, Figures 9-10 illustrate the maximum absolute error for Problem 7**,** computed at $h = 0.1$, showing a consistent pattern of accuracy throughout the solution domain. Collectively, these figures confirm that the proposed method delivers high precision across all test cases. The close agreement between the numerical results and exact solutions provides the robustness and efficiency of the developed hybrid block scheme.

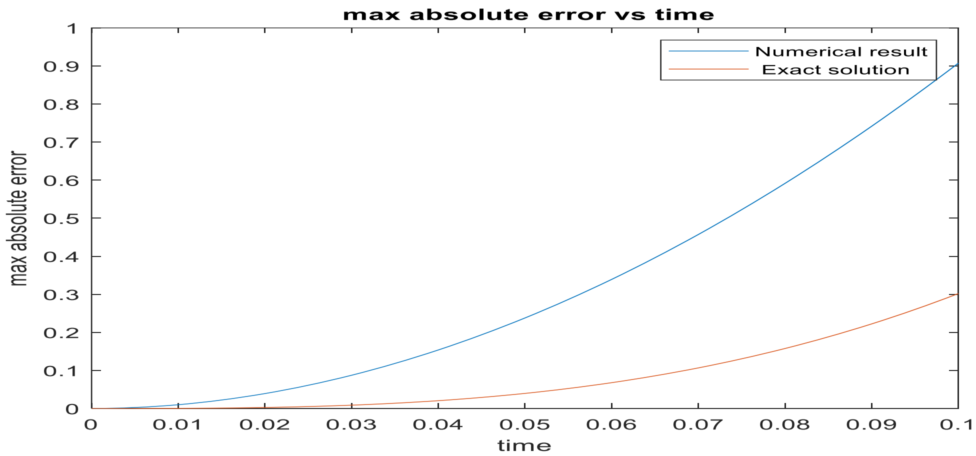

Figure 3. Efficiency curve for problem 1 at $h = 0.01$.

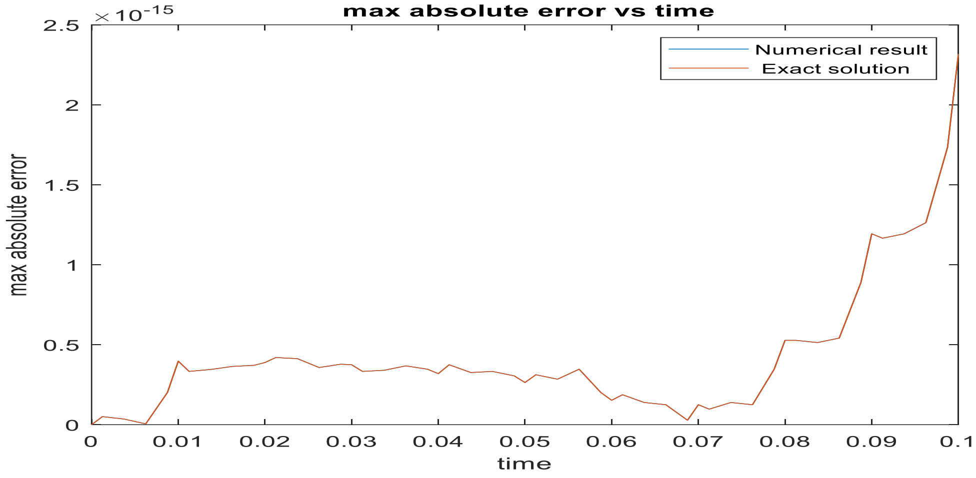

Figure 4. Efficiency curve for problem 2 at $h = 0.01$.

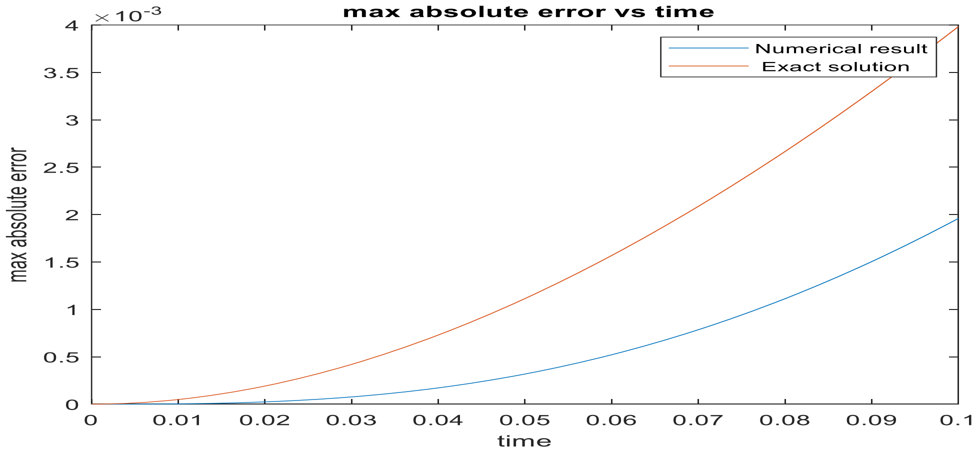

Figure 5. Efficiency curve for problem 3 at $h = 0.01$.

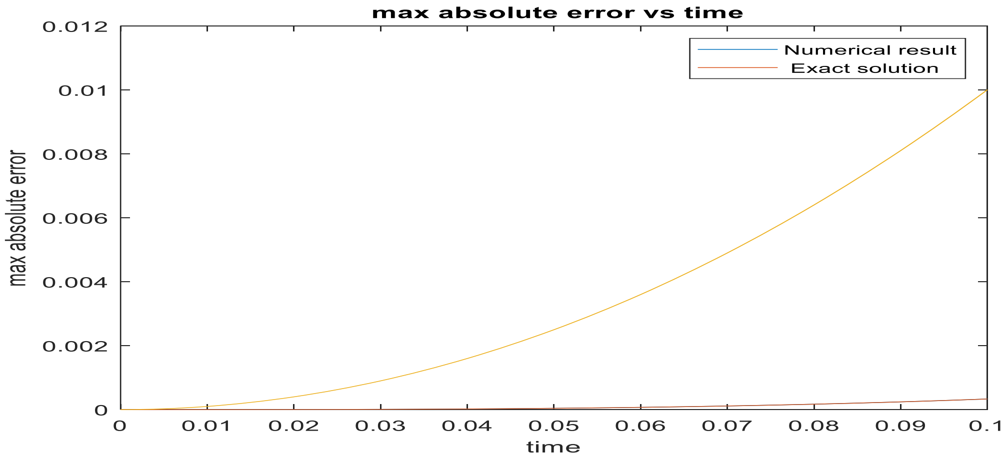

Figure 6. Efficiency curve for problem 4 at $h = 0.01$.

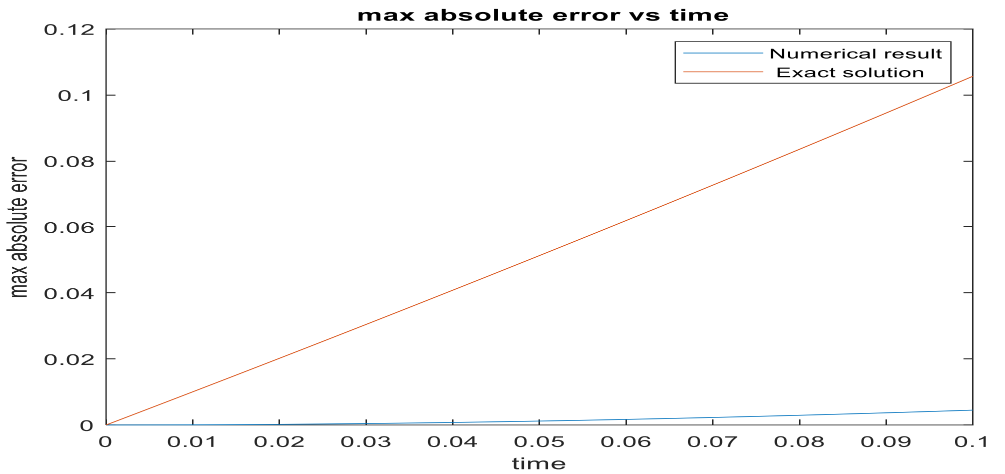

Figure 7. Efficiency curve for problem 5 at $h = 0.01$.

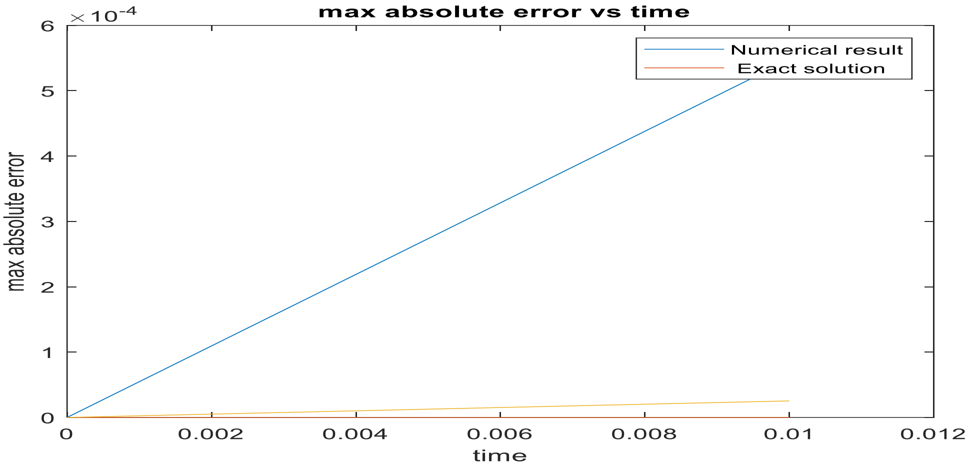

Figure 8. Efficiency curve for problem 6 at $h = 0.001$.

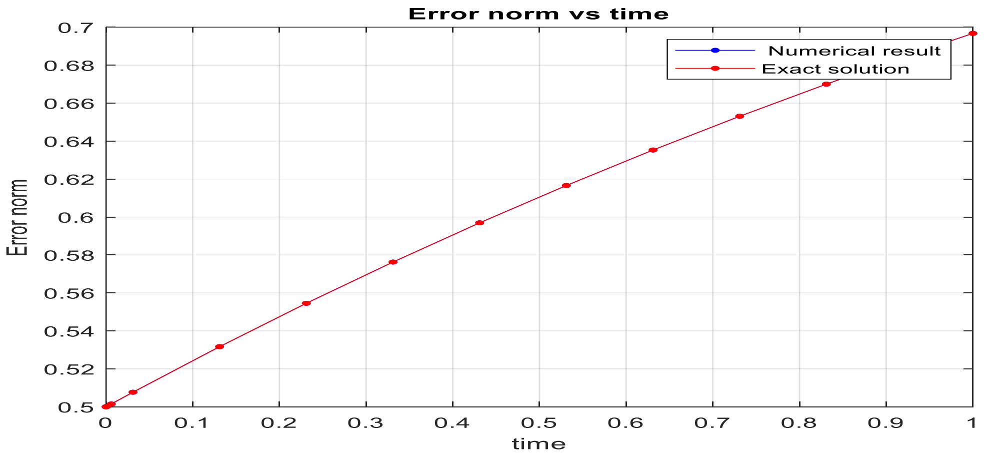

Figure 9. NANNM curve for problem 7 at $h = 0.1$.

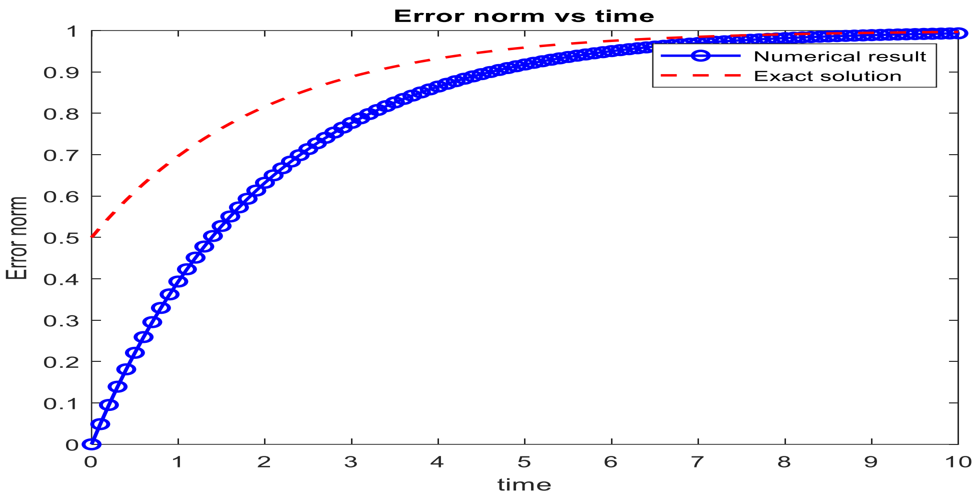

Figure 10. OHBM5A curve for problem 7 at $h = 0.1$.

3.1. Discussion of results

The use of an implicit solver in the proposed single-step hybrid block method significantly influences both accuracy and computational efficiency. Implicit solvers are generally more stable and permit larger step sizes without compromising accuracy. This enhanced accuracy results from the implicit formulation's ability to effectively manage stiffness and suppress numerical instabilities commonly encountered in explicit schemes. However, this comes with a trade-off: implicit methods typically require solving nonlinear or linear systems at each step, leading to increased CPU time due to the additional computational overhead. Despite this, MATLAB implementation results demonstrate that the gain in accuracy often outweighs the modest increase in computational time, as evidenced by the error versus CPU time plots shown in Figures 3–10. The proposed method was compared with those in [5], [6] and [7], considering both absolute error and CPU time for various step sizes h. In addition, comparisons were made with two established methods: the New Adaptive Nonlinear Numerical Method (NANNM) [28] and the Optimized Hybrid Block Method with fifth-order, adaptive and fixed step-size (OHBM5A) [29]. The plots of time versus maximum absolute error for the proposed method across Problems 1–6 are shown in Figures 3–8, while corresponding results for the established methods for Problem 7 are presented in Figures 9–10. Overall, the proposed hybrid block method produced the smallest absolute errors compared to other methods, indicating that it competes favourably with existing approaches.

4. Conclusion

This paper introduces a computational hybrid block method with four intra-step points for solving first-order stiff and non-stiff differential equations**.** The method extends existing block hybrid schemes originally designed for delay ordinary differential equations and adapts them to stiff problems through interpolation techniques for terms not defined at grid points.

Analytical properties such as zero-stability, consistency, convergence, and A-stability are rigorously established, confirming the method's theoretical soundness. Numerical experiments on several benchmark problems validate the approach, showing that it achieves high accuracy (sixth-order convergence) and competitive computational efficiency, outperforming existing methods in terms of absolute error and CPU time.

Practically, the proposed method is significant for engineering, biological, and physical models where system dynamics depend on past states with processes that evolve at vastly different rates (some components change very rapidly while others vary slowly). Its simplicity of implementation and robust stability characteristics make it well-suited for such applications.

However, potential limitations include the implicit nature of the scheme, which may increase computational cost due to the need for solving algebraic systems at each step, and dependence on accurate interpolation for stiff terms, which can introduce minor errors if not handled carefully. Additionally, CPU times were not comprehensively or consistently reported, limiting full quantitative comparison of efficiency across all test cases.

Overall, the paper makes a clear methodological contribution by extending hybrid block methods to a class of stiff and non-stiff differential equations, demonstrating strong performance and reliability for solving such problems.

Acknowledgement

Special thanks go to Almighty God for giving us direction, protection and guidance throughout this research. I sincerely appreciate the efforts of my supervisor, Associate Professor Yeak Su Hoe of the Department of Mathematical Science, Faculty of Science, Universiti Teknologi Malaysia, for giving me the necessary advice and guidance, which made this research successful. I also appreciate the Tertiary Education Trust Fund (TETFUND) of Nigeria for sponsoring my PhD program.

Conflicts of Interest

The authors declare no conflicts of interest concerning the publication of this research.

References

[1] Shampine, L. F., & Watts, H. A. (1969). Block implicit one-step methods. *Mathematics of Computation*, *23*(108), 731-740. http://dx.doi.org/10.1090/S0025-5718-1969-0264854-5

[2] Gear, C. W. (1965). Hybrid methods for initial value problems in ordinary differential equations. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, *2*(1), 69-86. doi: 102.90.43.100

[3] Motsa, S. (2022). Overlapping grid-based optimized single-step hybrid block method for solving first-order initial value problems. *Algorithms*, *15*(11), 427. doi: 10.3390/a15110427.

[4] Shateyi, S. (2023). On the application of the block hybrid methods to solve linear and non-linear first order differential equations. *Axioms*, *12*(2), 189.

[5] Bisheh-Niasar, M., & Ramos, H. (2025). A computational block method with five hybrid-points for differential equations containing a piece-wise constant delay. *Differential Equations and Dynamical Systems*, *33*(1), 1-14. doi: 10.1007/s12591-022-00624-9.

[6] Tassaddiq, A., Qureshi, S., Soomro, A., Hincal, E., & Shaikh, A. A. (2022). A new continuous hybrid block method with one optimal intrastep point through interpolation and collocation. *Fixed Point Theory and Algorithms for Sciences and Engineering*, *2022*(1), 22. doi: 10.1186/s13663-022-00733-8.

[7] Rufai, M. A., Carpentieri, B., & Ramos, H. (2023). A new hybrid block method for solving first-order differential system models in applied sciences and engineering. *Fractal and Fractional*, *7*(10), 703. https://doi.org/10.3390/fractalfract7100703

[8] Henrici, P. (1962). Discrete variable methods in ordinary differential equations. *John Wiley Sons, New York: Wiley*. doi: http://dx.doi.org/10.1002/zamm.19660460521

[9] Dhandapani, P. B., Baleanu, D., Thippan, J., & Sivakumar, V. (2019). Fuzzy type RK4 solutions to fuzzy hybrid retarded delay differential equations. *Frontiers in Physics*, *7*, 168. doi: 10.3389/fphy.2019.00168.

[10] Rosli, N., Ariffin, N. A. N., Hoe, Y. S., & Bahar, A. (2019, March). Stability Analysis of Explicit and Semi-implicit Euler Methods for Solving Stochastic Delay Differential Equations. In *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017) Transcending*

Boundaries, Embracing Multidisciplinary Diversities (pp. 179-186). Singapore: Springer Singapore. doi: 10.1007/978-981-13-7279-7_22.

[11] Duromola, M. K., Momoh, A. L., & Akinmoladun, O. M. (2022). Block extension of a single-step hybrid multistep method for directly solving fourth-order initial value problems. *American Journal of Computational Mathematics*, *12*(4), 355-371. doi: 10.4236/ajcm.2022.124026.

[12] Obarhua, F. O. (2023). Three-step Four-point Optimized Hybrid Block Method for Direct Solution of General Third Order Differential Equations. *Asian Research Journal of Mathematics*, *19*(6), 25-44. doi: 10.9734/arjom/2023/v19i6664.

[13] Jimoh, A. K. (2024). Two-step hybrid block method for solving second order initial value problem of ordinary differential equations. *Earthline Journal of Mathematical Sciences*, *14*(5), 1047-1065. doi: 10.34198/ejms.14524.10471065.

[14] Alemu Wendimu, A., Matušů, R., Gazdoš, F., & Shaikh, I. (2025). A Comparative Study of One-Step and Multi-Step Numerical Methods for Solving Ordinary Differential Equations in Water Tank Drainage Systems. *Engineering Reports*, *7*(3), e70080. doi: 10.1002/eng2.70080.

[15] Hoang, M. T., & Ehrhardt, M. (2024). A general class of second-order L-stable explicit numerical methods for stiff problems. *Applied Mathematics Letters*, *149*, 108897. doi: 10.1016/j.aml.2023.108897.

[16] Nwaigwe, C., & Atangana, A. (2025). Fourth-order predictor-corrector method for initial value ordinary differential equation problems. *Numerical Algorithms*, 1-34. doi: 10.1007/s11075-025-02043-7.

[17] Al-towaiq, M. (2023). Improved predictor-corrector method for the initial-value problem, Journal of Propulsion Technology, 44(6), 5832–5838, 2023. ISSN: 1001-4055

[18] Senu, N., Lee, K. C., Wan Ismail, W. F., Ahmadian, A., Ibrahim, S. N., & Laham, M. (2021). Improved Runge-Kutta method with trigonometrically-fitting technique for solving oscillatory problem. *Malaysian Journal of Mathematical Sciences*, *15*(2), 253-266. https://einspem.upm.edu.my/journal

[19] Abdulkarimova, A. R. (2025). On some comparison the methods of runge-kutta and multi-step types, *Int. J. Eng. Sci. Technol.*, *9(2)*, 103–109. doi: 10.29121/ijoest.v9.i2.2025.670.

[20] Lambert, J.D. (1974). Computational methods in ordinary differential equations, *J. Appl. Math. Mech.*, 54(7), 523–523. doi: https://doi.org/10.1002/zamm.19740540726.

[21] Dahlquist, G.G. (1963). A special stability problem for linear multistep methods, *Bit*, 3(1), 27–43. doi: 10.1007/BF01963532.

[22] Wang, L., Wang, F.& Y. Cao, Y. (2019). A hybrid collocation method for fractional initial value problems, *J. Integr. Equations Appl.*, 31(1), 105–129. doi: 10.1216/JIE-2019-31-1-105.

[23] Ramos, H., Qureshi, S. & Soomro, A.(2021). Adaptive step-size approach for Simpson's-type block methods with time efficiency and order stars, *Comput. Appl. Math.*, 40(6), 219. doi: 10.1007/s40314-021-01605-4.

[24] H. Soomro,H, Zainuddin, N., Daud, H. &. Sunday, J. (2022). Optimized hybrid block Adams method for solving first order ordinary differential equations, *Comput. Mater. Contin.*, 72(2). 2947–2961. doi: 10.32604/cmc.2022.025933.

[25] Areo, E. A., & Adeniyi, R. B. (2014). Block implicit one-step method for the numerical integration of initial value problems in ordinary differential equations. *International Journal of Mathematics and Statistics Studies*, *2*(3), 4-13. Available: www.ea-journals.org

[26] Qayyum, M., & Fatima, Q. (2022). Solutions of stiff systems of ordinary differential equations using residual power series method. *Journal of Mathematics*, *2022*(1), 7887136. doi: 10.1155/2022/7887136

[27] Singh, G., Garg, A., Kanwar, V., & Ramos, H. (2019). An efficient optimized adaptive step-size hybrid block method for integrating differential systems. *Applied Mathematics and computation*, *362*, 124567. doi: 10.1016/j.amc.2019.124567.

[28] August, N. (2025). Adaptive Nonlinear Numerical Method for Coupled Nonlinear IVP Systems: Theory, Experiments, and Error Analysis. *Journal of Computational Mathematics and Engineering*, *2*(3), 3–5

[29] Omar, A. (2016). *Hybrid Block Collocation–Interpolation Method for Stiff Problems: Sixth-Order Example*. IOSR Journal of Mathematics, *12*(5), 1–3