Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-21
DOI:10.22034/cmde.2025.65372.3001

An Advanced Numerical Approach for Solving Stiff Initial Value Problems Using a Self-Starting Two-Stage Composite Block Scheme

Badrul Amin Jaafar[1,2] **and Iskandar Shah Mohd Zawawi**[1,*]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.

[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Sarawak, Kampus Samarahan 2, 94300 Kota Samarahan, Sarawak, Malaysia.

Abstract

In this study, a two-point composite block method based on the backward differentiation formula (CBBDF) is introduced to solve stiff ordinary differential equations. The CBBDF method incorporates an additional intermediate point among the interpolating points, developed in two stages: the first stage employs the Euler's method as a fundamental building block, while the second stage utilizes CBBDF of order three. A key distinction of the method with the classical block method is the introduction of an independent parameter γ, which eliminates the need for an external startup calculation, while maintaining the accuracy and stability of numerical solutions. The theoretical analysis verifies that the proposed method is convergent and A-stable. It fulfills the essential properties of consistency and zero-stability, and it lies within the A-stability region. To demonstrate the effectiveness of the proposed approach, several stiff initial value problems of linear and non-linear are solved. For validation, the results are compared with existing literature. While approximating the solution at multiple points simultaneously, the composite block method offers the ability to use larger step sizes for solution approximation. The CBBDF method shows promising results, achieving a reliable degree of accuracy as indicated by its maximum error and average error measurements.

Keywords. Backward differentiation formula, Block method, Composite scheme, Linear multi-sub-step method, Stiff differential equations.
2010 Mathematics Subject Classification. 47E05, 65L04, 65L05.

1. Introduction

Differential equations are mathematical representations that play a crucial role in various fields of physical sciences and engineering worldwide over a significant period. Practical applications across various technological disciplines, including modeling electrical flow, analyzing pendulum dynamics, studying chemical reactions, and examining vibrations, among others. To gain a deeper insight into the underlying principles, scientists and engineers frequently analyze the changes of variables within a system of differential equations for a better understanding of physical phenomena. Among the most prominent types of differential equations that rely on a single independent variable are ordinary differential equations (ODEs), represented by the general form

$$y^n = F\left(x, y, y', y'', \ldots, y^{(n-1)}\right), \tag{1.1}$$

where F represents a specified function of x, y and the derivatives of y, and n denotes the order of ODEs. The order of ODEs is determined by the highest derivative, n present in the equation. Eq. (1.1) is classified as linear if the dependent variable, $y(x)$, and its derivatives do not appear in any multiplicative form. If such products are present, the equation is referred to as non-linear. However, many of these ODEs are classified as stiff, which are difficult or impossible to solve analytically. This difficulty stems from their inherent complexities and the coexistence of transient and steady-state solutions, which makes finding analytical solutions challenging [1, 31]. The concept of stiffness in

Received: 08 January 2025 ; Accepted: 23 April 2025.
∗ Corresponding author. Email: iskandarshah@uitm.edu.my.

ODEs is described in multiple ways in the literature, as there is no singular definition that is universally acknowledged. Generally, stiffness occurs when the equations exhibit significantly different time scales, meaning that some parts of the solution decay at a much faster rate than others [1, 24]. This results in a situation where solutions change slowly in certain regions while nearby solutions change rapidly. This disparity requires numerical methods to use very small time steps to maintain accuracy, leading to computational inefficiencies [4, 22]. For simplicity, the definition of a stiff problem provided by Lambert [15] is referred to, as many authors cite this definition to understand the concept of stiffness in ODEs.

Definition 1.1. Eq. (1.1) is classified as stiff if

- $\mathrm{Re}\,(\lambda_t) < 0,\ t = 1, 2, \ldots, m$, and
- $\max_t |\mathrm{Re}\,\lambda_t| \gg \min_t |\mathrm{Re}\,\lambda_t|$ where λ_t are the eigenvalues of the Jacobian matrix, $J = \left(\frac{\partial f}{\partial y}\right)$.

Otherwise, it is defined as non-stiff.

From Definition 1.1, stiffness can be measured by the ratio of the largest to the smallest eigenvalue of the Jacobian matrix associated with the system [4, 24, 27]. A larger ratio signifies a stiffer system, which complicates the stability conditions for explicit methods. As a result, explicit methods are often impractical for stiff systems. Although these methods typically incur lower computational costs, they require extremely small time steps to maintain numerical stability due to their stringent stability conditions.

To overcome this challenge, the employment of implicit methods becomes essential for effectively estimating solutions to stiff problems. There is a strong preference for implicit methods as they offer optimal stability. While implicit methods are well-suited for stiff systems, they demand greater computational effort for each step. Alternatively, numerical methods for solving ODEs are typically classified as either single-step or multi-step methods. The single-step method utilizes information from a single initial point to produce an approximation at a new point. In contrast, the multi-step method relies on a sequence of prior solution and derivative values [20].

One of the most widely used implicit multi-step methods in the literature is the backward differentiation formula (BDF), acclaimed for its efficacy in addressing stiff problems due to its A-stability region [22, 28]. Although these methods offer better accuracy, they often entail higher computing costs [20]. This is due to the utilization of more steps which leads to increased computational complexity and longer execution times. To address these issues, many studies have shifted from solving (1.1) at a single point per step to adopting a strategy that employs two or more points per step, commonly referred to as the block method. The essence of the block method lies in its ability to generate a block of approximations $y_{n+1}, y_{n+2}, \ldots, y_{n+N}$ simultaneously, where N indicates the number of points in the block formula [6, 18]. The number of points is determined by the configuration of the block method. As a result, utilizing these methods can lead to faster solutions and reduce both computational time and the overall number of steps, while achieving the desired level of accuracy. Among the notable block-based methods for solving stiff problems is the block backward differentiation formula (BBDF) introduced by [7], which offers the advantage of approximating the solution at multiple points simultaneously in each step by incorporating backward values from previous blocks, resulting in efficient computations. Numerous enhancements and extensions have been developed for the conventional multi-step block methods (see [1, 5, 8, 21, 24]).

A recently emerging category of multi-step methods, gaining notable attention, is the multi-sub-step method, also recognized as composite time integration method. The fundamental concept of the composite method involves breaking down each time step into several smaller sub-steps and employing different numerical integration schemes for each sub-step [10, 26, 34]. One of the pioneering composite methods is the Bathe method, introduced by [2]. This method develops a direct time integration technique that utilizes a composite single-step strategy. It uniquely combines the trapezoidal rule for the initial sub-step and the three-point backward difference method for subsequent sub-steps. This combination significantly enhances the precision of lower-frequency modes while providing improved damping for higher-frequency modes. Its strong dissipation and reliable accuracy make it effective for transient analysis. Since then, further studies have developed extended implicit composite methods (see [9, 12, 13, 17, 25, 33]). [30] developed the trapezoidal rule and second-order BDF (TR-BDF2) composite scheme, which is equivalent to specific singly diagonally implicit Runge-Kutta methods. This scheme provides a high-order one-step analogue of multi-step methods and is self-starting. Past few years, [3] introduced two fully implicit methods: backward Euler and composite backward

differentiation formula (CBDF2), used for solving bidomain equations. CBDF2 offers better accuracy and efficiency with larger timesteps, unlike backward Euler, which requires smaller steps. Meanwhile, [11] proposed the established composite method by combining the Eulers method and BDF for solving stiff ODEs.

To the best of the author's knowledge, no existing study has integrated the standard BBDF with composite schemes, highlighting a significant research gap in applying composite block methods for stiff ODEs. This study fills this gap by introducing the novel composite block backward differentiation formula (CBBDF), aiming to improve accuracy and optimize stability in numerical solutions. Both the standard BBDF and CBBDF methods compute solutions at two points simultaneously, but the CBBDF offers two key advantages: it is self-starting, eliminating the need for external startup computations, and it allows for larger integration steps while maintaining accuracy. This ultimately reduces computational costs for solving stiff ODEs.

The structural framework of this study unfolds as follows: First, the fundamentals of composite block schemes are introduced, and the formulation of the proposed method is derived in section 2. Subsequently, section 3 delves into a comprehensive analysis of the method's order, error constants, convergence behavior, and stability properties of the derived method. Next, section 4 provides an in-depth exploration of the implementation of the proposed method utilizing Newton's iteration. To demonstrate the accuracy and applicability of composite block scheme algorithm, various numerical examples for simulating linear and non-linear, single and system of first-order stiff ODEs are considered in section 5. Finally, the results and potential future research are discussed in section 6.

2. Derivation of the method

In this section, the CBBDF scheme is developed with two sub-steps by Eulers method serves as a fundamental building block in the first stage and a two-point CBBDF of order three in the second stage as illustrated in Figure 1. The step size for the first point is divided into two segments, $[x_n, x_{n+\gamma}]$ and $[x_{n+\gamma}, x_{n+1}]$ in which h represents

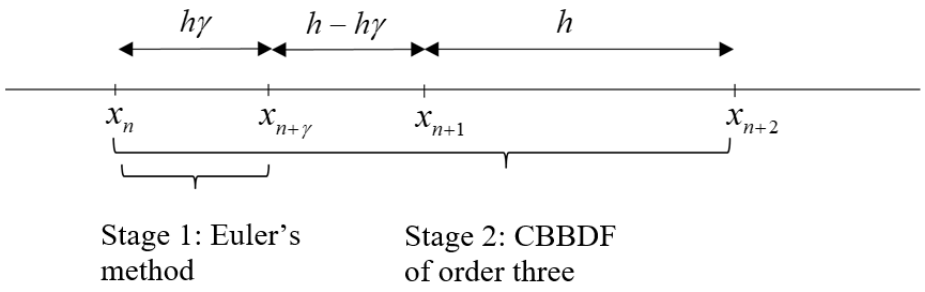

Figure 1. The interpolating points of the CBBDF.

the size of the integration step and parameter γ functions as an independent parameter in the intermediate point, $x_{n+\gamma}$. The contribution of this newly developed method is the introduction of intermediate point, which serves as an effective strategy to refine future solutions, which enhance both accuracy and stability in the numerical solution process. This point is not included in the main sequence of calculated points; rather, it serves as an additional step that improves the overall approximation of (1.1). By evaluating the function at this intermediate point, the algorithm can adjust its subsequent steps based on more accurate information. The incorporation of this intermediate point with an independent parameter enhances the iterative process that refines the subsequent values to achieve better approximations. Subsequently, the following point uses constant step size for the interval, $[x_{n+1}, x_{n+2}]$. To approximate the solutions of ODEs, the known value y_n and the intermediate value $y_{n+\gamma}$ serve as the preceding block values, with the intermediate value calculated in the first stage. These two values are then employed to simultaneously compute the future values y_{n+1} and y_{n+2} in the second stage. The CBBDF scheme to be derived is a k-step linear

multi-sub-step method (LMM), defined as follows:

$$\sum_{j=0}^{k-2} A_j y_{n+j} + A_\gamma y_{n+\gamma} = h \left[\sum_{j=0}^{k-2} B_j f_{n+j} + B_\gamma f_{n+\gamma} \right]. \tag{2.1}$$

The local approximation of CBBDF will be obtained using the Lagrange interpolation polynomial, $P_k(x)$ of degree k by considering one intermediate point in the interpolating points. The general expression for the Lagrange interpolation polynomial is presented below:

$$P_k(x) = \sum_{j=0}^{k-2} L_{k,j}(x)\, y(x_{n+2-j}) + L_{k,\gamma}(x)\, y(x_{n+\gamma}), \tag{2.2}$$

whereas

$$L_{k,j}(x) = \prod_{i=0, i\neq j}^{k-2} \frac{(x - x_{n+2-i})}{(x_{n+2-j} - x_{n+2-i})} \cdot \frac{(x - x_{n+\gamma})}{(x_{n+2-j} - x_{n+\gamma})},$$
$$L_{k,\gamma}(x) = \prod_{i=0}^{k-2} \frac{(x - x_{n+2-i})}{(x_{n+\gamma} - x_{n+2-i})},$$

for $j = 0, \ldots, k-2$.

2.1. **Derivation of the method for stage one.** In the first stage, the set of interpolation points consists of x_n and $x_{n+\gamma}$. The polynomial that results from this process is expressed as follows:

$$P(x) = \frac{(x - x_{n+\gamma})}{(x_n - x_{n+\gamma})} y_n + \frac{(x - x_n)}{(x_{n+\gamma} - x_n)} y_{n+\gamma}. \tag{2.3}$$

Replace $x = sh + x_{n+\gamma}$ into (2.3) and differentiate the coefficients of interpolation polynomial with respect to s will achieve

$$P'(sh + x_{n+\gamma}) = -\left(\frac{1}{\gamma}\right) y_n + \left(\frac{1}{\gamma}\right) y_{n+\gamma}. \tag{2.4}$$

Evaluate $s = -\gamma$ in (2.4) and consider $hf_n = P'(x_n)$ yields

$$y_{n+\gamma} = y_n + \gamma h f_n. \tag{2.5}$$

2.2. **Derivation of the method for stage two.** In the second stage, the set of interpolation points includes x_n, $x_{n+\gamma}$, x_{n+1} and x_{n+2}. The polynomial generated from these points is expressed as follows:

$$\begin{aligned} P(x) = {} & \frac{(x - x_{n+\gamma})(x - x_{n+1})(x - x_{n+2})}{(x_n - x_{n+\gamma})(x_n - x_{n+1})(x_n - x_{n+2})} y_n \\ & + \frac{(x - x_n)(x - x_{n+1})(x - x_{n+2})}{(x_{n+\gamma} - x_n)(x_{n+\gamma} - x_{n+1})(x_{n+\gamma} - x_{n+2})} y_{n+\gamma} \\ & + \frac{(x - x_n)(x - x_{n+\gamma})(x - x_{n+2})}{(x_{n+1} - x_n)(x_{n+1} - x_{n+\gamma})(x_{n+1} - x_{n+2})} y_{n+1} \\ & + \frac{(x - x_n)(x - x_{n+\gamma})(x - x_{n+1})}{(x_{n+2} - x_n)(x_{n+2} - x_{n+\gamma})(x_{n+2} - x_{n+1})} y_{n+2}. \end{aligned} \tag{2.6}$$

Substitute $x = sh + x_{n+\gamma}$ into (2.6) and differentiate it once with respect to s. Then, substitute $s = 1-\gamma$ and $s = 2-\gamma$ in succession to obtain the following equations:

$$\begin{aligned} P'(x_{n+1}) &= \left(\frac{1-\gamma}{2\gamma}\right) y_n + \left(\frac{1}{\gamma(1-\gamma)(\gamma-2)}\right) y_{n+\gamma} \\ &\quad + \left(\frac{1}{1-\gamma}\right) y_{n+1} + \left(\frac{1-\gamma}{4-2\gamma}\right) y_{n+2}, \\ P'(x_{n+2}) &= \left(\frac{\gamma-2}{2\gamma}\right) y_n + \left(\frac{2}{\gamma(\gamma-1)(\gamma-2)}\right) y_{n+\gamma} \\ &\quad + \left(\frac{4-2\gamma}{\gamma-1}\right) y_{n+1} + \left(\frac{8-3\gamma}{4-2\gamma}\right) y_{n+2}. \end{aligned} \tag{2.7}$$

Consider $hf_{n+1,n+2} = P'(x_{n+1,n+2})$ in (2.7) will obtain

$$\begin{aligned} y_{n+1} &= \left(\frac{(\gamma-1)(1-\gamma)}{2\gamma}\right) y_n + \left(\frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)}\right) y_{n+\gamma} \\ &\quad + \left(\frac{(\gamma-1)(1-\gamma)}{4-2\gamma}\right) y_{n+2} + (1-\gamma) hf_{n+1}, \\ y_{n+2} &= \left(\frac{(2-\gamma)(4-2\gamma)}{2\gamma(8-3\gamma)}\right) y_n + \left(\frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)}\right) y_{n+\gamma} \\ &\quad + \left(\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)}\right) y_{n+1} + \left(\frac{4-2\gamma}{8-3\gamma}\right) hf_{n+2}. \end{aligned} \tag{2.8}$$

The algorithm is implemented in $PECE$ mode, where P denotes predictor, E refers to the evaluation of (1.1), and C represents corrector, as derived in (2.8). The predictor formula is formulated similarly to the corrector formula, but without the differentiation step. The interpolation points used are x_n and $x_{n+\gamma}$, and the Lagrange polynomial can be expressed as shown in (2.3). Consequently, the predictor formulas for simultaneously calculating the predicted values of y_{n+1} and y_{n+2} in stage two are obtained:

$$\begin{aligned} y_{n+1} &= \left(\frac{\gamma-1}{\gamma}\right) y_n + \left(\frac{1}{\gamma}\right) y_{n+\gamma}, \\ y_{n+2} &= \left(\frac{\gamma-2}{\gamma}\right) y_n + \left(\frac{2}{\gamma}\right) y_{n+\gamma}. \end{aligned} \tag{2.9}$$

3. Properties of the method

This section analyzes the fundamental properties of the proposed method by determining its order and error constants, examining its convergence properties, which include consistency and zero-stability, and illustrating the absolute stability region.

3.1. Order and error constant.

Definition 3.1 (Order of the method)**.** A linear difference operator, $L(y(x))$ and the associated LMM are classified as having order q if the coefficients $C_0 = C_1 = \ldots = C_q$ are all zero, while the error constant, C_{q+1} is non-zero.

As stated in [15], the order of a numerical method, along with its associated error constant, is often assessed by examining the local truncation error, which is associated with (2.1). To facilitate this assessment, the linear difference operator, denoted as $L(y(x))$, can be described in the following manner:

$$L[y(x); h] = \sum_j A_j y(x + jh) - hB_j y'(x + jh), \tag{3.1}$$

where $j = 0, \gamma, 1, 2$. By applying Taylor series expansions to (3.1) for $y(x+jh)$ and its derivative $y'(x+jh)$ around the point x, the following results will be obtained:

$$\begin{aligned} L[y(x);h] &= \sum_j A_j \left[y(x) + \left(\frac{jh}{1!}\right) y'(x) + \left(\frac{(jh)^2}{2!}\right) y''(x) + \left(\frac{(jh)^3}{3!}\right) y'''(x) + \ldots \right] \\ &\quad - \sum_j hB_j \left[y'(x) + \left(\frac{jh}{1!}\right) y''(x) + \left(\frac{(jh)^2}{2!}\right) y'''(x) + \left(\frac{(jh)^3}{3!}\right) y^{(4)}(x) + \ldots \right] \\ &= \sum_j [A_j]\, y(x) + \sum_j [(jA_j) - (B_j)]\, hy'(x) + \sum_j \left[\frac{1}{2}(j^2 A_j) - (jB_j)\right] h^2 y''(x) \\ &\quad + \sum_j \left[\frac{1}{6}(j^3 A_j) - \frac{1}{2}(j^2 B_j)\right] h^3 y'''(x) + \sum_j \left[\frac{1}{24}(j^4 A_j) - \frac{1}{6}(j^3 B_j)\right] h^4 y^{(4)}(x) \\ &\quad + \ldots . \end{aligned} \tag{3.2}$$

The common terms from the series expansions in (3.2) can be combined as follows:

$$L[y(x);h] = C_0 y(x) + C_1 hy'(x) + \ldots + C_q h^q y^{(q)}(x) + \ldots, \tag{3.3}$$

where C_q are constants given by

$$\begin{aligned} C_0 &= \sum_j A_j, \\ C_q &= \sum_j \left(\frac{1}{q!} j^q A_j - \frac{1}{(q-1)!} j^{q-1} B_j\right), \quad q = 1, 2, \ldots, \end{aligned}$$

where $j = 0, \gamma, 1, 2$. To ascertain the order of the method, (2.8) can be expressed as follows:

$$\begin{aligned} &y_{n+1} - \left[\frac{(\gamma-1)(1-\gamma)}{2\gamma}\right] y_n - \left[\frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)}\right] y_{n+\gamma} - \left[\frac{(\gamma-1)(1-\gamma)}{4-2\gamma}\right] y_{n+2} = [1-\gamma]\, hf_{n+1}, \\ &y_{n+2} - \left[\frac{(2-\gamma)(4-2\gamma)}{(2\gamma)(8-3\gamma)}\right] y_n - \left[\frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)}\right] y_{n+\gamma} - \left[\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)}\right] y_{n+1} = \left[\frac{4-2\gamma}{8-3\gamma}\right] hf_{n+2}. \end{aligned} \tag{3.4}$$

The matrix form of (3.4) is given by

$$\begin{aligned} &\begin{bmatrix} -\frac{(\gamma-1)(1-\gamma)}{2\gamma} & -\frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)} \\ -\frac{(2-\gamma)(4-2\gamma)}{(2\gamma)(8-3\gamma)} & -\frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)} \end{bmatrix} \begin{bmatrix} y_n \\ y_{n+\gamma} \end{bmatrix} + \begin{bmatrix} 1 - \frac{(\gamma-1)(1-\gamma)}{4-2\gamma} & \\ -\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} \\ &= h \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\gamma} \end{bmatrix} + h \begin{bmatrix} 1-\gamma & 0 \\ 0 & \frac{4-2\gamma}{8-3\gamma} \end{bmatrix} \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}, \end{aligned} \tag{3.5}$$

where

$$\begin{aligned} A_0 &= \begin{bmatrix} -\frac{(\gamma-1)(1-\gamma)}{2\gamma} \\ -\frac{(2-\gamma)(4-2\gamma)}{(2\gamma)(8-3\gamma)} \end{bmatrix}, A_\gamma = \begin{bmatrix} -\frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)} \\ -\frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)} \end{bmatrix}, \\ A_1 &= \begin{bmatrix} 1 \\ -\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} \end{bmatrix}, A_2 = \begin{bmatrix} -\frac{(\gamma-1)(1-\gamma)}{4-2\gamma} \\ 1 \end{bmatrix}, B_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ B_\gamma &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, B_1 = \begin{bmatrix} 1-\gamma \\ 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ \frac{4-2\gamma}{8-3\gamma} \end{bmatrix}. \end{aligned}$$

The associated set of coefficients from (3.5) is implemented to achieve

$$\begin{aligned}
C_0 =& A_0 + A_\gamma + A_1 + A_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_1 =& ((0 \cdot A_0) + (\gamma \cdot A_\gamma) + (1 \cdot A_1) + (2 \cdot A_2)) \\
& - (B_0 + B_\gamma + B_1 + B_2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_2 =& \frac{1}{2}\left((0^2 \cdot A_0) + (\gamma^2 \cdot A_\gamma) + (1^2 \cdot A_1) + (2^2 \cdot A_2)\right) \\
& - ((0 \cdot B_0) + (\gamma \cdot B_\gamma) + (1 \cdot B_1) + (2 \cdot B_2)) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_3 =& \frac{1}{6}\left((0^3 \cdot A_0) + (\gamma^3 \cdot A_\gamma) + (1^3 \cdot A_1) + (2^3 \cdot A_2)\right) \\
& - \frac{1}{2}\left((0^2 \cdot B_0) + (\gamma^2 \cdot B_\gamma) + (1^2 \cdot B_1) + (2^2 \cdot B_2)\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_4 =& \frac{1}{24}\left((0^4 \cdot A_0) + (\gamma^4 \cdot A_\gamma) + (1^4 \cdot A_1) + (2^4 \cdot A_2)\right) \\
& - \frac{1}{6}\left((0^3 \cdot B_0) + (\gamma^3 \cdot B_\gamma) + (1^3 \cdot B_1) + (2^3 \cdot B_2)\right) = \begin{bmatrix} \frac{\gamma^2 - 2\gamma + 1}{24} \\ \frac{(\gamma-2)^2}{18\gamma - 48} \end{bmatrix}.
\end{aligned}$$

According to Definition 3.1, $C_0 = C_1 = C_2 = C_3 = [0, 0, 0, 0]^T$ and $C_4 \neq 0$. Therefore, this proves that the derived method has the order of three.

3.2. **Convergence.** The convergence of numerical methods, particularly LMM, is a critical aspect that ensures that the numerical solutions approximate the exact solutions of differential equations as the step size approaches zero. The relevant definition regarding the characteristics of convergence properties of the LMM is as follows [24]:

Definition 3.2 (Convergence of the method)**.** For the LMM to converge, it is essential and sufficient that the method is both consistent and zero-stable.

3.2.1. *Consistency.*

Definition 3.3 (Consistency)**.** The LMM is said to be consistent if and only if

$$\begin{aligned}
\sum_j A_j =& 0, \\
\sum_j j A_j =& \sum_j B_j,
\end{aligned} \tag{3.6}$$

where $j = 0, \gamma, 1, 2$.

The consistency of the CBBDF is determined by substituting the set of coefficients from (3.5) into the conditions in Definition 3.3. This process leads to

$$\sum_j A_j = A_0 + A_\gamma + A_1 + A_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Hence, the first condition in (3.6) is satisfied.

$$\sum_j j A_j = (0 \cdot A_0) + (\gamma \cdot A_\gamma) + (1 \cdot A_1) + (2 \cdot A_2) = \begin{bmatrix} 1 - \gamma \\ \frac{4 - 2\gamma}{8 - 3\gamma} \end{bmatrix},$$

$$\sum_j B_j = B_0 + B_\gamma + B_1 + B_2 = \begin{bmatrix} 1-\gamma \\ \frac{4-2\gamma}{8-3\gamma} \end{bmatrix}.$$

Hence, $\sum_j jA_j = \sum_j B_j$. Consequently, the second condition in (3.6) is fulfilled as well. As a result, the consistency criteria are satisfied, confirming that the CBBDF is consistent.

3.2.2. *Zero-stability.*

Definition 3.4 (Zero-stability)**.** The LMM is said to be zero-stable if and only if no root of the first characteristic polynomial has a modulus greater than 1, and that every root with modulus 1 is simple.

The zero-stability of the CBBDF is conducted by utilizing the standard linear test differential equation expressed in the following form:

$$y' = \lambda y, \tag{3.7}$$

where $\lambda < 0$ is complex. By substituting (3.7) into (2.8) produces

$$\begin{aligned} &y_{n+1} - \left(\frac{(\gamma-1)(1-\gamma)}{4-2\gamma}\right) y_{n+2} - ((1-\gamma)\lambda h)\, y_{n+1} \\ &\quad = \left(\frac{(\gamma-1)(1-\gamma)}{2\gamma}\right) y_n + \left(\frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)}\right) y_{n+\gamma}, \\ &y_{n+2} - \left(\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)}\right) y_{n+1} - \left(\left(\frac{4-2\gamma}{8-3\gamma}\right)\lambda h\right) y_{n+2} \\ &\quad = \left(\frac{(2-\gamma)(4-2\gamma)}{2\gamma(8-3\gamma)}\right) y_n + \left(\frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)}\right) y_{n+\gamma}. \end{aligned} \tag{3.8}$$

Consider $\bar{h} = \lambda h$, (3.8) can be represented in matrix form as follows:

$$\begin{bmatrix} 1-(1-\gamma)\bar{h} & -\frac{(\gamma-1)(1-\gamma)}{4-2\gamma} \\ -\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} & 1-\left(\frac{4-2\gamma}{8-3\gamma}\right)\bar{h} \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} = \begin{bmatrix} \frac{(\gamma-1)(1-\gamma)}{2\gamma} & \frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)} \\ \frac{(2-\gamma)(4-2\gamma)}{2\gamma(8-3\gamma)} & \frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)} \end{bmatrix} \begin{bmatrix} y_n \\ y_{n+\gamma} \end{bmatrix}. \tag{3.9}$$

Eq. (3.9) is equivalent to

$$\alpha Y_m = \beta Y_{m-1},$$

where

$$\alpha = \begin{bmatrix} 1-(1-\gamma)\bar{h} & -\frac{(\gamma-1)(1-\gamma)}{4-2\gamma} \\ -\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} & 1-\left(\frac{4-2\gamma}{8-3\gamma}\right)\bar{h} \end{bmatrix}, \quad \beta = \begin{bmatrix} \frac{(\gamma-1)(1-\gamma)}{2\gamma} & \frac{1-\gamma}{\gamma(\gamma-1)(\gamma-2)} \\ \frac{(2-\gamma)(4-2\gamma)}{2\gamma(8-3\gamma)} & \frac{2(2\gamma-4)}{\gamma(\gamma-1)(\gamma-2)(8-3\gamma)} \end{bmatrix}.$$

The computation of $R\left(t, \bar{h}, \gamma\right) = \det\left(\alpha t - \beta\right)$ leads to the stability polynomial of the CBBDF, which is given by

$$R\left(t, \bar{h}, \gamma\right) = \frac{1}{\gamma(8-3\gamma)} \begin{pmatrix} 2t^2\gamma^3 + 2\gamma^3 t^2\bar{h}^2 + \gamma^3 t\bar{h} - 2t\gamma^3 - 3\gamma^3 t^2\bar{h} + 9t\gamma^2 + 13\gamma^2 t^2\bar{h} \\ -9t^2\gamma^2 - 6\gamma^2 t^2\bar{h}^2 - 4\gamma^2 t\bar{h} + 5\gamma t\bar{h} + 4\gamma t^2\bar{h}^2 + 12t^2\gamma - 12\gamma t \\ -12\gamma t^2\bar{h} - 1 + t - 6t\bar{h} \end{pmatrix}. \tag{3.10}$$

Letting $\bar{h} = 0$ in (3.10) to obtain the first characteristics polynomial given by

$$R\left(t, \gamma\right) = \frac{1}{\gamma(8-3\gamma)}\left(-2t\gamma^3 + 2t^2\gamma^3 + 9t\gamma^2 - 9t^2\gamma^2 + 12t^2\gamma - 12t\gamma + t - 1\right). \tag{3.11}$$

The following roots of the first characteristic polynomial are obtained after solving (3.11) as

$$t_1 = 1, \quad t_2 = \frac{1}{\gamma(-2\gamma^2 + 9\gamma - 12)}.$$

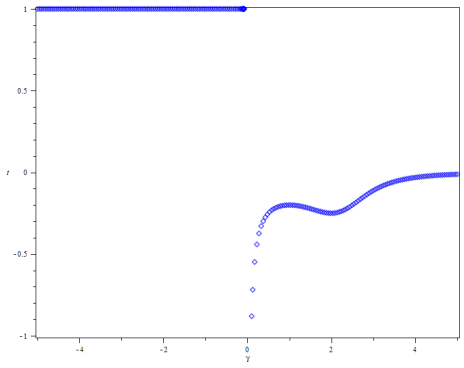

FIGURE 2. Graph of t_2 versus γ for CBBDF.

Since γ is included in t_2, it is required that the graph of t_2 be illustrated over a certain interval of γ to meet the criteria of Definition 3.4. For a clearer representation of the graph, the values of γ within the range of $[-5, 5]$ will be considered. Figure 2 shows that when $\gamma \leq 0.1$, then $t_2 = 1$. According to Definition 3.4, a method is considered zero-stable if all roots on the unit circle are simple, which means they have a multiplicity of one. However, in this case, when $\gamma \leq 0.1$, there are two roots, t_1 and t_2, both equal to 1, indicating that the method lacks zero-stability. Therefore, it can be concluded that the CBBDF method exhibits zero-stable when $\gamma \in (0.1, \infty)$, as none of the roots have a modulus exceeding 1, and the root $t = 1$ is simple, thereby satisfying the stability conditions. Hence, the parameter γ is expected to lie within the range $\gamma \in (0.1, \infty)$ to ensure that the underlying splitting ratio of the CBBDF meets the necessary condition for stiff stability.

3.2.3. *Stability region.*

Definition 3.5 (A-stable)**.** The LMM is said to be A-stable (often referred to as stiffly stable) when the stability region encompasses the entire left half of the complex plane.

The stability region is defined as the area bounded by the set of points where $|t| = 1$. To delineate the boundary of this region, one can substitute $t = e^{i\theta}$ for $0 \leq \theta \leq 2\pi$ into the stability polynomial. The resulting graphs of the stability region are generated using MAPLE software. Figure 3 presents the stability regions for the CBBDF method with parameters $\gamma = 20, 50, 100$. In these graphs, the stable region is located outside the circular symbol line, while the unstable region is enclosed within it. Figure 4 shows the stability regions for BBDF and CBBDF with $\gamma = 20, 50, 100$ to compare their stability behavior.

The figures demonstrate that the CBBDF method has a smaller unstable region in comparison to the BBDF method. According to Definition 3.5, the CBBDF method exhibits A-stability, as its stability regions extend across the entire left half-plane. Therefore, this method is A-stable and well-suited for solving stiff problems. For the implementation of the method, it is important to choose an appropriate value of γ without sacrificing accuracy or A-stability, since γ influences the error constant C_4, which might affects the magnitude of the truncation errors.

4. Implementation of the method

The CBBDF method is implemented within a predictor-corrector computation framework and is self-starting, meaning it does not require an external startup calculation to initiate the predictor-corrector scheme. This method is structured in two stages. In the first stage, the known value at point x_n is utilized to calculate the intermediate value at $x_{n+\gamma}$ using the Euler method. Once this initial sub-step is completed, the values at both y_n and $y_{n+\gamma}$ are established. Following this, the approximate solutions at points x_{n+1} and x_{n+2} can be determined using the data from these two points. This is done by initially employing the predictor formula outlined in (2.9) to estimate the values. Subsequently, these predicted values are refined using the corrector formula presented in (2.8). The implementation of the corrector formula is carried out through Newton's iteration technique. Therefore, this section presents the

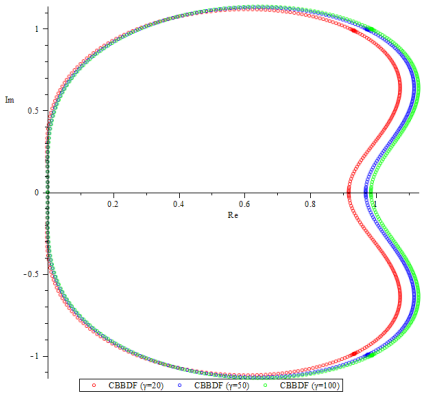

FIGURE 3. The graphs of stability regions for CBBDF.

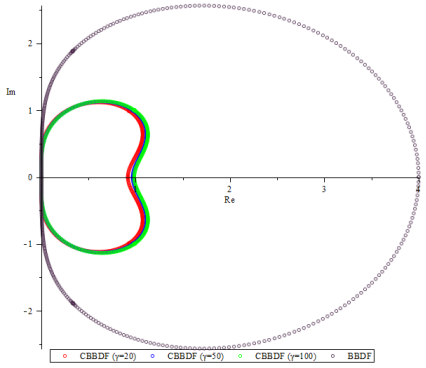

FIGURE 4. The graphs of stability regions for BBDF and CBBDF.

formulation of the Newton's iteration method to concurrently obtain the approximate solutions for y_{n+1} and y_{n+2} during stage two at each step. The corrector formula associated with the CBBDF method can be written in a general equation as follows:

$$\begin{aligned} y_{n+1} &= \phi_1 y_{n+2} + \omega_1 f_{n+1} + \psi_1, \\ y_{n+2} &= \phi_2 y_{n+1} + \omega_2 f_{n+2} + \psi_2, \end{aligned} \tag{4.1}$$

with ψ_1 and ψ_2 are the previous values. Then, (4.1) is transformed into a matrix-vector form in the following manner:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} = \begin{bmatrix} 0 & \phi_1 \\ \phi_2 & 0 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} + h \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix} \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix} + \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}. \tag{4.2}$$

Simplifying (4.2) yields $(I - \Phi)\, Y_{n+1,n+2} = h\Omega F_{n+1,n+2} + \Psi$ with

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \Phi = \begin{bmatrix} 0 & \phi_1 \\ \phi_2 & 0 \end{bmatrix}, \ Y_{n+1,n+2} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix},$$
$$\Omega = \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix}, \ F_{n+1,n+2} = \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}, \Psi = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}.$$

Eq. (4.2) is written in the form of

$$\hat{F} = (I - \Phi)\, Y_{n+1,n+2} - h\Omega F_{n+1,n+2} - \Psi. \tag{4.3}$$

Therefore, the formula for generalized Newton's iteration is expressed as

$$y^{(i+1)}_{n+1,n+2} = y^{(i)}_{n+1,n+2} - \left[\frac{\hat{F}_{1,2}\left(y^{(i)}_{n+1,n+2}\right)}{\hat{F}'_{1,2}\left(y^{(i)}_{n+1,n+2}\right)} \right], \tag{4.4}$$

where the indices (i) and $(i+1)$ denote the previous and current iterations, respectively. The equation of (4.4) can be rearranged as follows:

$$\hat{F}'_{1,2}\left(y^{(i)}_{n+1,n+2}\right) \cdot \left[y^{(i+1)}_{n+1,n+2} - y^{(i)}_{n+1,n+2} \right] = -\hat{F}_{1,2}\left(y^{(i)}_{n+1,n+2}\right). \tag{4.5}$$

By substituting (4.3) into (4.5), the following equation will achieve

$$\left[(I - \Phi) - h\Omega \frac{\partial F}{\partial Y}\left(Y^{(i)}_{n+1,n+2}\right) \right] \cdot \left[y^{(i+1)}_{n+1,n+2} - y^{(i)}_{n+1,n+2} \right] = -\left[(I - \Phi)\, Y^{(i)}_{n+1,n+2} - h\Omega F\left(Y^{(i)}_{n+1,n+2}\right) - \Psi \right], \tag{4.6}$$

where $\frac{\partial F}{\partial Y}\left(Y^{(i)}_{n+1,n+2}\right)$ represents the Jacobian matrix of F with respect to Y. The following notation will be used:

$$e^{(i+1)}_{n+1,n+2} = y^{(i+1)}_{n+1,n+2} - y^{(i)}_{n+1,n+2}, \tag{4.7}$$

where $e^{(i+1)}_{n+1,n+2}$ indicates the differences between the values of $y_{n+1,n+2}$ at the $(i)^{th}$ and $(i+1)^{th}$ iterations. Substituting (4.7) into (4.6) transforms the Newton's iteration formula into the following form:

$$(I-\Phi) - h\Omega\frac{\partial F}{\partial Y}\left(Y^{(i)}_{n+1,n+2}\right)\cdot\left[e^{(i+1)}_{n+1,n+2}\right] = -\left[(I-\Phi)\,Y^{(i)}_{n+1,n+2} - h\Omega F\left(Y^{(i)}_{n+1,n+2}\right) - \Psi\right]. \tag{4.8}$$

Thus, the estimated values of $y_{n+1,n+2}$ are calculated using the formula $y^{(i+1)}_{n+1,n+2} = y^{(i)}_{n+1,n+2} + e^{(i+1)}_{n+1,n+2}$, where the absolute error is defined as follows:

$$\text{error}^{(i+1)} = \left|y^{(i+1)}_{\text{exact}} - y^{(i+1)}_{\text{approximate}}\right|.$$

The assessment of maximum error, MAXE is given by

$$\text{MAXE} = \max\left[\text{error}^{(i+1)}\right].$$

Rearranging the formula (2.8) into general form (4.3) yields

$$\begin{aligned} F_1 &= y_{n+1} - \left(\frac{(\gamma-1)(1-\gamma)}{4-2\gamma}\right)y_{n+2} - (1-\gamma)\,hf_{n+1} - \psi_1,\\ F_2 &= y_{n+2} - \left(\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)}\right)y_{n+1} - \left(\frac{4-2\gamma}{8-3\gamma}\right)hf_{n+2} - \psi_2, \end{aligned} \tag{4.9}$$

where ψ_1 and ψ_2 are the previous values. Substituting (4.9) into (4.8) will produce

$$\begin{aligned} \left(1-(1-\gamma)\,h\frac{\partial f_{n+1}}{\partial y_{n+1}}\right)e^{(i+1)}_{n+1} &= -y^{(i)}_{n+1} + \left(\frac{(\gamma-1)(1-\gamma)}{4-2\gamma}\right)y^{(i)}_{n+2} + (1-\gamma)\,hf^{(i)}_{n+1} + \psi_1,\\ \left(1-\left(\frac{4-2\gamma}{8-3\gamma}\right)h\frac{\partial f_{n+2}}{\partial y_{n+2}}\right)e^{(i+1)}_{n+2} &= -y^{(i)}_{n+2} + \left(\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)}\right)y^{(i)}_{n+1} + \left(\frac{4-2\gamma}{8-3\gamma}\right)hf^{(i)}_{n+2} + \psi_2. \end{aligned} \tag{4.10}$$

Eq. (4.10) are written in matrix form:

$$\begin{aligned} \begin{bmatrix} 1-(1-\gamma)\,h\frac{\partial f_{n+1}}{\partial y_{n+1}} & -\frac{(\gamma-1)(1-\gamma)}{4-2\gamma} \\ -\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} & 1-\left(\frac{4-2\gamma}{8-3\gamma}\right)h\frac{\partial f_{n+2}}{\partial y_{n+2}} \end{bmatrix}\begin{bmatrix} e^{(i+1)}_{n+1} \\ e^{(i+1)}_{n+2} \end{bmatrix} &= \begin{bmatrix} -1 & \frac{(\gamma-1)(1-\gamma)}{4-2\gamma} \\ \frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} & -1 \end{bmatrix}\begin{bmatrix} y^{(i)}_{n+1} \\ y^{(i)}_{n+2} \end{bmatrix}\\ &\quad + h\begin{bmatrix} 1-\gamma & 0 \\ 0 & \frac{4-2\gamma}{8-3\gamma} \end{bmatrix}\begin{bmatrix} f^{(i)}_{n+1} \\ f^{(i)}_{n+2} \end{bmatrix} + \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}. \end{aligned} \tag{4.11}$$

In this study, the computation follows a predict-evaluate-correct-evaluate, $PE(CE)^r$ mode to simultaneously approximate the solutions of initial value problems (IVPs) at two distinct points. The power $r = 2$ denotes the number of iterations required to converge the solutions of the corrector formulas. Therefore, the following steps are carried out:

Step 1: The predicted values are computed using the predictor formulas.
$P : y^{(p)}_{n+1,n+2}$.

Step 2: The predicted values are used to find the derivatives.
$E : y'_{n+1,n+2} = f\left(x_{n+1,n+2}, y^{(p)}_{n+1,n+2}\right)$.

Step 3: The iteration matrices are applied to find the increments.
$C : y^{(c)}_{n+1,n+2}$.

Step 4: The corrected values are used to evaluate the values of derivatives.
$E : y'_{n+1,n+2} = f\left(x_{n+1,n+2}, y^{(c)}_{n+1,n+2}\right)$.

For approximating the solutions of $y^{(c)}_{n+1,n+2}$, a two-stage Newton's iteration scheme, as described in (4.11), is applied. The iterative process proceeds as follows:

Step 1: Initially, calculate the values for $e_{n+1,n+2}^{(i+1)} = J^{-1}K$, where in general;

$$J = \begin{bmatrix} 1-(1-\gamma)\,h\frac{\partial f_{n+1}}{\partial y_{n+1}} & -\frac{(\gamma-1)(1-\gamma)}{4-2\gamma} \\ -\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)} & 1-\left(\frac{4-2\gamma}{8-3\gamma}\right)h\frac{\partial f_{n+2}}{\partial y_{n+2}} \end{bmatrix},$$

$$K = \begin{bmatrix} -y_{n+1}^{(i)} + \left(\frac{(\gamma-1)(1-\gamma)}{4-2\gamma}\right)y_{n+2}^{(i)} + (1-\gamma)\,hf_{n+1}^{(i)} + \psi_1 \\ -y_{n+2}^{(i)} + \left(\frac{(4-2\gamma)(4-2\gamma)}{(1-\gamma)(8-3\gamma)}\right)y_{n+1}^{(i)} + \left(\frac{4-2\gamma}{8-3\gamma}\right)hf_{n+2}^{(i)} + \psi_2 \end{bmatrix}.$$

Step 2: Determine the corrected value for $y_{n+1,n+2}^{(i+1)}$ using the result $e_{n+1,n+2}^{(i+1)}$ obtained in Step 1, as shown below:

$$y_{n+1,n+2}^{(i+1)} = y_{n+1,n+2}^{(i)} + e_{n+1,n+2}^{(i+1)}.$$

Step 3: Repeat the process from Step 1 to solve $e_{n+1,n+2}^{(i+1)} = J^{-1}K$ for the second stage of iteration.

Step 4: Finally, obtain the values for $y_{n+1,n+2}^{(i+1)}$ by executing the calculations from the second stage iteration, $e_{n+1,n+2}^{(i+1)}$.

The computation of matrices in both Step 1 and Step 3 is solved using lower-upper (LU) decomposition for computational efficiency and the potential for speedup in high-performance computing environments.

5. Results and discussions

In this section, six IVPs of linear and non-linear first order ODEs are solved to demonstrate the method's effectiveness. For the analysis of the numerical results, various step sizes are considered, specifically $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$. The independent parameters, $\gamma = 20, 50, 100$ are chosen due to the A-stability characteristic. The numerical outcomes obtained from the CBBDF method are compared against those produced by the conventional 2-point BBDF method. The computation of the tested problems is executed using C programming. For Examples 5.1 to 5.6, the method's performance is evaluated by examining both the maximum error and the average error. To present the results visually, graphs of $\log_{10}$(MAXE) plotted against $\log(h)$ are created for each test problem, as illustrated in Figures 5–10. The following notations are employed:

- h : Step size,
- γ : Independent parameter of CBBDF,
- BBDF : 2-point block backward differentiation formula by [7],
- CBBDF : Composite block backward differentiation formula,
- MAXE : Maximum error,
- AVE : Average error.

Example 5.1. ([6]) Consider the linear differential equation given by

$$y'(x) = -20y + 20\sin(x) + \cos(x), \quad \text{for } 0 \le x \le 2.$$

The initial condition for this problem is $y(0) = 1$. The corresponding eigenvalue for this equation is $\lambda = -20$. The exact solution to the differential equation can be expressed as

$$y(x) = \sin(x) + e^{-20x}.$$

Example 5.2. ([23]) The non-linear differential equation is given by

$$y'(x) = \frac{y(1-y)}{2y-1}, \quad \text{for } 0 \le x \le 1.$$

The initial condition is set as $y(0) = \frac{5}{6}$, and the associated eigenvalue is identified as $\lambda = -1$. The exact solution is expressed as

$$y(x) = \frac{1}{2} + \sqrt{\left(\frac{1}{4} - \frac{5}{36}e^{-x}\right)}.$$

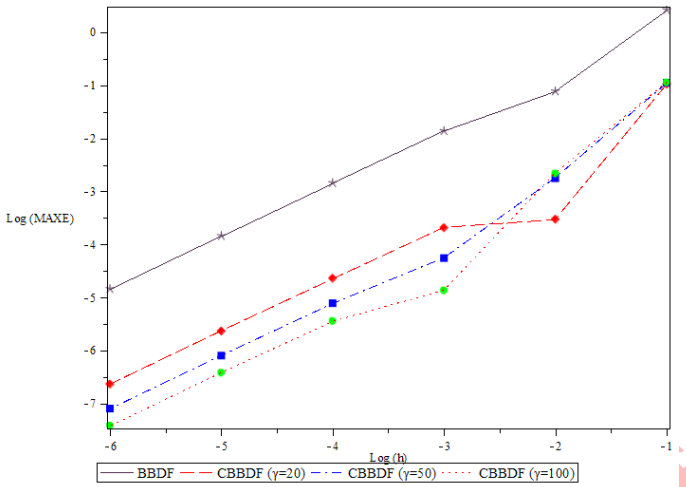

FIGURE 5. Accuracy curves of CBBDF for Example 5.1.

Example 5.3. ([1]) Another non-linear differential equation under consideration is

$$y'(x) = -\frac{y^3}{2}, \text{ for } 0 \le x \le 4.$$

The initial condition is specified as $y(0) = 1$, and the corresponding eigenvalue is determined to be $\lambda = -1$. The exact solution to this equation is given by

$$y(x) = \frac{1}{\sqrt{1+x}}.$$

Example 5.4. ([29]) In this test problem, a system of linear equations is analyzed, characterized as mildly stiff. The equations are defined as follows:

$$y_1'(x) = 198y_1 + 199y_2, \quad y_2'(x) = -398y_1 - 399y_2, \text{ for } 0 \le x \le 5.$$

The initial conditions are specified as $y_1(0) = 1$ and $y_2(0) = -1$. The eigenvalues associated with this system are given by $\lambda_1 = -1$ and $\lambda_2 = -200$. The exact solutions for the variables are expressed as

$$y_1(x) = e^{-x}, \quad y_2(x) = -e^{-x}.$$

Example 5.5. ([8]) This test problem involves a system of linear equations that can be classified as highly stiff. The equations are represented as follows:

$$y_1'(x) = -y_1 + 95y_2, \quad y_2'(x) = -y_1 - 97y_2, \text{ for } 0 \le x \le 10.$$

The initial conditions for this system are given by $y_1(0) = 1$ and $y_2(0) = 1$. The eigenvalues associated with this system are $\lambda_1 = -1$ and $\lambda_2 = -1000$. The exact solutions for the variables can be expressed as:

$$y_1(x) = \frac{1}{47}\left(95e^{-2x} - 48e^{-96x}\right), \quad y_2(x) = \frac{1}{47}\left(48e^{-96x} - e^{-2x}\right).$$

Example 5.6. ([19]) This test problem presents a system of non-linear equations that is also categorized as highly stiff. The equations are formulated as follows:

$$y_1'(x) = -\left(\varepsilon^{-1} + 2\right)y_1 + \varepsilon^{-1}y_2^2, \quad y_2'(x) = y_1 - y_2(1 + y_2), \text{ for } 0 \le x \le 20,$$

where $\varepsilon = 10^{-5}$. The initial conditions are defined as $y_1(0) = 1$ and $y_2(0) = 1$. The eigenvalues associated with this system are given by $\lambda_1 = -1$ and $\lambda_2 = -100002$. The exact solutions for the variables can be expressed as:

$$y_1(x) = e^{-2x}, \quad y_2(x) = e^{-x}.$$

TABLE 1. Numerical results for Example 5.1.

h	Method	γ	MAXE	AVE
10^{-1}	BBDF	-	2.67842e+000	1.01381e+000
	CBBDF	20	1.04535e-001	6.08993e-003
		50	1.11490e-001	6.58036e-003
		100	1.13243e-001	6.70337e-003
10^{-2}	BBDF	-	7.82684e-002	5.01369e-003
	CBBDF	20	2.98302e-004	2.77528e-005
		50	1.76860e-003	1.04303e-004
		100	2.17108e-003	1.25277e-004
10^{-3}	BBDF	-	1.40171e-002	9.91969e-004
	CBBDF	20	2.06383e-004	1.47189e-005
		50	5.53643e-005	4.02406e-006
		100	1.35961e-005	1.06649e-006
10^{-4}	BBDF	-	1.46435e-003	1.05132e-004
	CBBDF	20	2.30157e-005	1.65361e-006
		50	7.81013e-006	5.61821e-007
		100	3.60021e-006	2.59531e-007
10^{-5}	BBDF	-	1.47063e-004	1.05736e-005
	CBBDF	20	2.32541e-006	1.67204e-007
		50	8.03770e-007	5.78004e-008
		100	3.82434e-007	2.75069e-008
10^{-6}	BBDF	-	1.47126e-005	1.05796e-006
	CBBDF	20	2.32780e-007	1.67385e-008
		50	8.06044e-008	5.79663e-009
		100	3.84667e-008	2.76750e-009

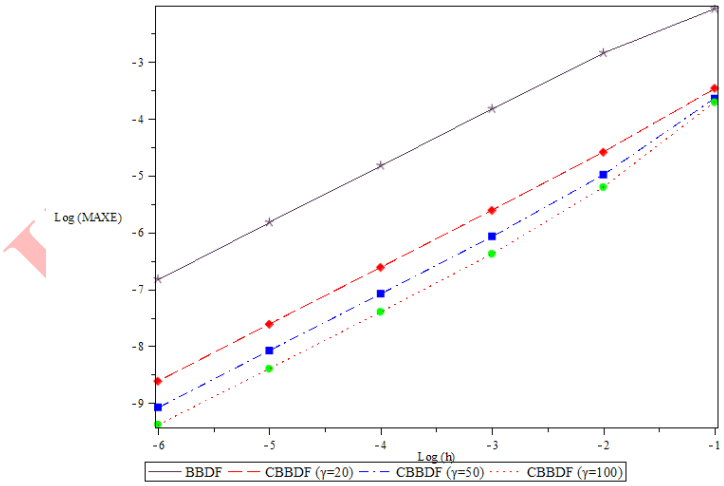

FIGURE 6. Accuracy curves of CBBDF for Example 5.2.

TABLE 2. Numerical results for Example 5.2.

h	Method	γ	MAXE	AVE
10^{-1}	BBDF	-	8.71737e-003	3.19499e-003
	CBBDF	20	3.43593e-004	1.76087e-004
		50	2.27808e-004	1.15068e-004
		100	1.95675e-004	1.11630e-004
10^{-2}	BBDF	-	1.47086e-003	1.11270e-003
	CBBDF	20	2.57766e-005	1.95330e-005
		50	1.04460e-005	7.18930e-006
		100	6.27701e-006	3.77272e-006
10^{-3}	BBDF	-	1.52651e-004	1.22148e-004
	CBBDF	20	2.44049e-006	1.95375e-006
		50	8.58168e-007	6.81865e-007
		100	4.20518e-007	3.29694e-007
10^{-4}	BBDF	-	1.53220e-005	1.23268e-005
	CBBDF	20	2.42688e-007	1.95303e-007
		50	8.41784e-008	6.76942e-008
		100	4.02879e-008	3.23596e-008
10^{-5}	BBDF	-	1.53277e-006	1.23380e-006
	CBBDF	20	2.42542e-008	1.95236e-008
		50	8.40206e-009	6.76277e-009
		100	4.01359e-009	3.22997e-009
10^{-6}	BBDF	-	1.53301e-007	1.23404e-007
	CBBDF	20	2.41680e-009	1.94629e-009
		50	8.43666e-010	6.78621e-010
		100	4.18106e-010	3.34428e-010

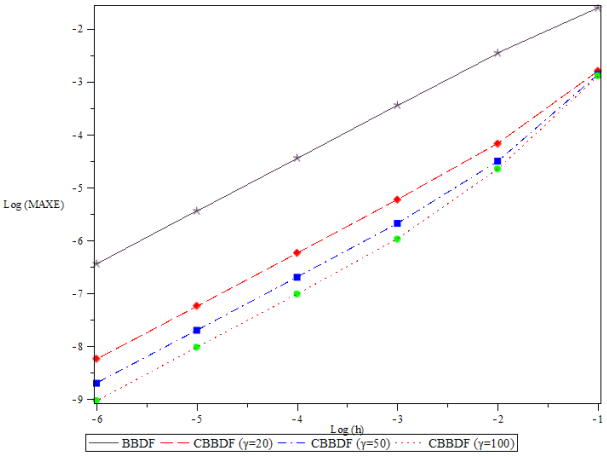

FIGURE 7. Accuracy curves of CBBDF for Example 5.3.

TABLE 3. Numerical results for Example 5.3.

h	Method	γ	MAXE	AVE
10^{-1}	BBDF	-	2.50939e-002	1.86364e-002
	CBBDF	20	1.60282e-003	1.19991e-003
		50	1.36667e-003	9.94416e-004
		100	1.30225e-003	9.38707e-004
10^{-2}	BBDF	-	3.53603e-003	2.77144e-003
	CBBDF	20	6.71866e-005	5.55681e-005
		50	3.16326e-005	2.65473e-005
		100	2.25009e-005	1.85035e-005
10^{-3}	BBDF	-	3.66423e-004	2.88117e-004
	CBBDF	20	5.90195e-006	4.68027e-006
		50	2.10859e-006	1.69591e-006
		100	1.06188e-006	8.69457e-007
10^{-4}	BBDF	-	3.67734e-005	2.89233e-005
	CBBDF	20	5.82889e-007	4.58865e-007
		50	2.02491e-007	1.59674e-007
		100	9.71622e-008	7.68269e-008
10^{-5}	BBDF	-	3.67865e-006	2.89345e-006
	CBBDF	20	5.82177e-008	4.57959e-008
		50	2.01684e-008	1.58674e-008
		100	9.63135e-009	7.57857e-009
10^{-6}	BBDF	-	3.67859e-007	2.89330e-007
	CBBDF	20	5.83388e-009	4.59544e-009
		50	2.01438e-009	1.58250e-009
		100	9.45308e-010	7.32319e-010

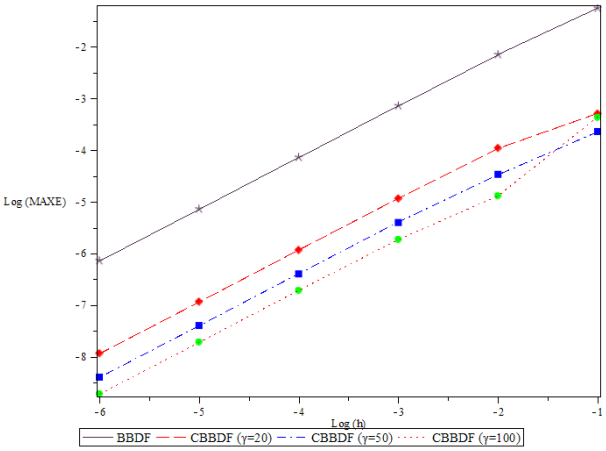

FIGURE 8. Accuracy curves of CBBDF for Example 5.4.

TABLE 4. Numerical results for Example 5.4.

h	Method	γ	MAXE	AVE
10^{-1}	BBDF	-	5.67155e-002	5.30550e-002
	CBBDF	20	5.08253e-004	4.85707e-004
		50	2.29216e-004	2.20454e-004
		100	4.32265e-004	4.15048e-004
10^{-2}	BBDF	-	7.18323e-003	7.41994e-003
	CBBDF	20	1.09804e-004	1.13736e-004
		50	3.40025e-005	3.52230e-005
		100	1.30253e-005	1.34931e-005
10^{-3}	BBDF	-	7.34012e-004	7.65072e-004
	CBBDF	20	1.15755e-005	1.20687e-005
		50	3.96879e-006	4.13792e-006
		100	1.86262e-006	1.94199e-006
10^{-4}	BBDF	-	7.35584e-005	7.67399e-005
	CBBDF	20	1.16351e-006	1.21386e-006
		50	4.02568e-007	4.19988e-007
		100	1.91864e-007	2.00166e-007
10^{-5}	BBDF	-	7.35741e-006	7.67632e-006
	CBBDF	20	1.16411e-007	1.21458e-007
		50	4.03131e-008	4.20608e-008
		100	1.92407e-008	2.00750e-008
10^{-6}	BBDF	-	7.35747e-007	7.67643e-007
	CBBDF	20	1.16542e-008	1.21574e-008
		50	4.03394e-009	4.20679e-009
		100	1.91571e-009	1.99671e-009

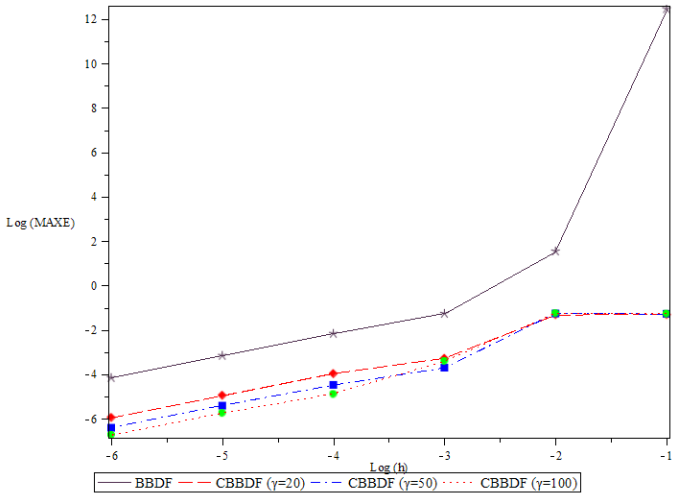

FIGURE 9. Accuracy curves of CBBDF for Example 5.5.

TABLE 5. Numerical results for Example 5.5.

h	Method	γ	MAXE	AVE
10^{-1}	BBDF	-	2.97925e+012	2.60132e+011
	CBBDF	20	4.92960e-002	1.64341e-003
		50	5.35157e-002	2.13669e-003
		100	5.43563e-002	2.25776e-003
10^{-2}	BBDF	-	3.62349e+001	4.41254e+000
	CBBDF	20	4.88913e-002	2.02706e-004
		50	5.53607e-002	1.77429e-004
		100	5.70686e-002	1.70235e-004
10^{-3}	BBDF	-	5.62364e-002	6.83856e-004
	CBBDF	20	5.24748e-004	8.97600e-006
		50	2.00878e-004	3.21477e-006
		100	3.99990e-004	3.09103e-006
10^{-4}	BBDF	-	7.04927e-003	7.92858e-005
	CBBDF	20	1.07912e-004	1.23504e-006
		50	3.35839e-005	4.06160e-007
		100	1.30140e-005	1.76713e-007
10^{-5}	BBDF	-	7.19695e-004	8.04701e-006
	CBBDF	20	1.13513e-005	1.27130e-007
		50	3.89349e-006	4.38128e-008
		100	1.82855e-006	2.07427e-008
10^{-6}	BBDF	-	7.21175e-005	8.05893e-007
	CBBDF	20	1.14074e-006	1.27510e-008
		50	3.94703e-007	4.41350e-009
		100	1.88128e-007	2.10375e-009

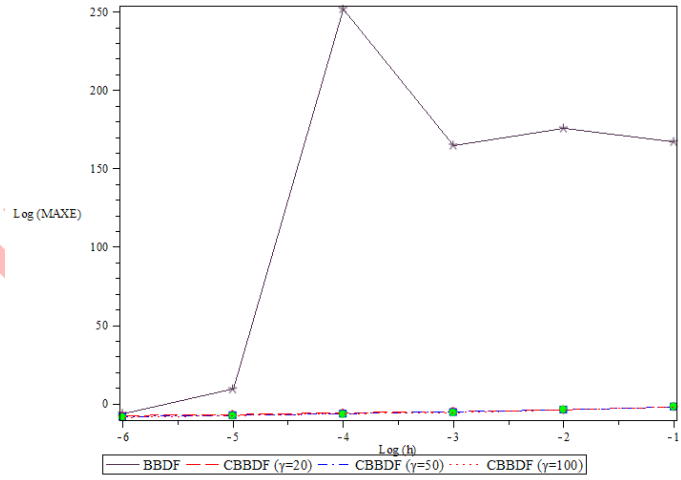

FIGURE 10. Accuracy curves of CBBDF for Example 5.6.

TABLE 6. Numerical results for Example 5.6.

h	Method	γ	MAXE	AVE
10^{-1}	BBDF	-	1.58887e+167	-
	CBBDF	20	1.25834e-002	1.00091e-003
		50	1.30889e-002	1.00962e-003
		100	1.32314e-002	1.04871e-003
10^{-2}	BBDF	-	5.81600e+175	-
	CBBDF	20	1.09807e-004	2.24160e-005
		50	1.59882e-004	1.65663e-005
		100	1.78720e-004	1.51610e-005
10^{-3}	BBDF	-	7.73254e+164	-
	CBBDF	20	1.15757e-005	2.21828e-006
		50	3.96886e-006	6.91611e-007
		100	1.86265e-006	3.19438e-007
10^{-4}	BBDF	-	8.46176e+251	-
	CBBDF	20	1.16353e-006	2.35789e-007
		50	4.02573e-007	8.06667e-008
		100	1.91866e-007	3.77321e-008
10^{-5}	BBDF	-	2.58326e+009	1.06691e+009
	CBBDF	20	1.16413e-007	2.37217e-008
		50	4.03136e-008	8.20719e-009
		100	1.92410e-008	3.91108e-009
10^{-6}	BBDF	-	7.35764e-007	1.49999e-007
	CBBDF	20	1.16543e-008	2.37533e-009
		50	4.03401e-009	8.21764e-010
		100	1.91573e-009	3.89904e-010

From Tables 1–6, the CBBDF demonstrate significant improvements in error reduction over the existing BBDF method. The proposed method can achieve comparable or even better accuracy with larger step sizes, offering less computational cost. It can even improve the maximum error with step sizes as large as 0.1. In contrast, classical BBDF typically requires smaller step sizes to maintain accuracy. For instance, the BBDF is incapable of maintaining accuracy with step sizes larger than 0.01, as shown in Tables 1, 5, and 6.

Remarkably, the CBBDF successfully generates better solutions by using a step size of 0.1 for highly stiff problems, a feat that is considered nearly impossible for standard LMM like the BBDF, which typically require much smaller step sizes to avoid instability or inaccuracies. For instance, in Tables 5 and 6, the BBDF struggles to achieve better accuracy, which shows higher maximum errors when considering a larger step size of 10^{-3} and 10^{-6}, respectively. For the independent parameter, it is observed that as the step size decreases, the parameter $\gamma = 100$ yields superior numerical results compared to $\gamma = 20$ and $\gamma = 50$. This suggests that as the value of γ increases, the accuracy of the results tends to improve. However, while larger γ values may enhance certain aspects of accuracy and stability, they can also increased truncation errors. This trade-off highlights the importance of carefully selecting γ to achieve a balance between high accuracy, optimal stability, and minimal truncation errors in numerical computations.

The advantages of CBBDF are largely due to the incorporation of numerical dissipation in composite schemes, which stabilizes the solution by damping high-frequency oscillations that arise in stiff or oscillatory problems [14, 32]. This controlled dissipation enables the method to maintain accuracy even with larger step sizes by mitigating the numerical instabilities that typically occur at higher frequencies. Moreover, the utilization of sub-stepping mechanisms allows for capturing transient behaviors more accurately without requiring excessively small global time steps [16]. The

flexibility in choosing sub-step sizes enhances the method's ability to adapt to varying dynamics within the same time frame.

6. Conclusion

In summary, a self-starting composite block scheme, namely CBBDF has been proposed for the purpose of solving first-order stiff IVPs. Furthermore, it has been proven that the proposed method is of order three, A-stable, and convergent. The numerical results demonstrate that CBBDF offers better accuracy than the traditional 2-point BBDF method in both maximum and average error metrics. One significant implication of implementing composite block schemes is their exceptional performance in effectively addressing linear and non-linear stiff problems with larger step sizes. Another striking feature of CBBDF excels in solving highly stiff problems, where the existing method struggles. Future research will concentrate on devising a strategy to improve the efficiency of the proposed method by addressing higher-order stiff IVPs through a variable-order and variable-step formula.

Acknowledgments

This research was funded by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme (FRGS/1/2024/STG06/UITM/02/3) and Universiti Teknologi MARA.

References

[1] N. Abd Rasid, Z. B. Ibrahim, Z. Abd Majid, and F. Ismail, *Formulation of a new implicit method for group implicit BBDF in solving related stiff ordinary differential equations*, Statistics, *9*(2) (2021), 144–150.
[2] K. J. Bathe and M. M. I. Baig, *On a composite implicit time integration procedure for nonlinear dynamics*, Comput. Struct., *83*(31–32) (2005), 2513–2524.
[3] X. Gao, C. S. Henriquez, and W. Ying, *Composite backward differentiation formula for the bidomain equations*, Front. Physiol., *11* (2020), 591159.
[4] Z. B. Ibrahim, H. Mohd Ijam, S. Jana Aksah, and N. Abd Rasid, *Enhancing accuracy and efficiency in stiff ODE integration using variable step diagonal BBDF approaches*, Mal. J. Fund. Appl. Sci., *20*(5) (2024), 1083–1100.
[5] Z. B. Ibrahim, N. Mohd Noor, and K. I. Othman, *Fixed coefficient $A(\alpha)$ stable block backward differentiation formulas for stiff ordinary differential equations*, Symmetry, *11*(7) (2019), 846.
[6] Z. B. Ibrahim and A. A. Nasarudin, *A class of hybrid multistep block methods with A–stability for the numerical solution of stiff ordinary differential equations*, Mathematics, *8*(6) (2020), 914.
[7] Z. B. Ibrahim, K. I. Othman, and M. Suleiman, *Implicit r-point block backward differentiation formula for solving first-order stiff ODEs*, Appl. Math. Comput., *186*(1) (2007), 558–565.
[8] S. Jana Aksah, Z. B. Ibrahim, and I. S. Mohd Zawawi, *Stability analysis of singly diagonally implicit block backward differentiation formulas for stiff ordinary differential equations*, Mathematics, *7*(2) (2019), 211.
[9] Y. Ji and Y. Xing, *An optimized three-sub-step composite time integration method with controllable numerical dissipation*, Comput. Struct., *231* (2020), 106210.
[10] Y. Ji, H. Zhang, and Y. Xing, *New insights into a three-sub-step composite method and its performance on multibody systems*, Mathematics, *10*(14) (2022), 2375.
[11] S. A. Junaidi, B. A. Jaafar, and I. S. Mohd Zawawi, *Composite backward differentiation formulas for solving stiff ordinary differential equation*, AIP Conf. Proc., *3189*(1) (2024), 070005.
[12] W. Kim and S. Y. Choi, *An improved implicit time integration algorithm: the generalized composite time integration algorithm*, Comput. Struct., *196* (2018), 341–354.
[13] W. Kim and J. N. Reddy, *An improved time integration algorithm: a collocation time finite element approach*, Int. J. Struct. Stab. Dyn., *17*(2) (2017), 1750024.
[14] W. Kim and J. N. Reddy, *A novel family of two-stage implicit time integration schemes for structural dynamics*, Int. J. Comput. Methods, *18*(8) (2021), 2150021.
[15] J. D. Lambert, *Computational methods in ordinary differential equations*, John Wiley & Sons, New York, 1973.
[16] J. Li, K. Yu, and H. He, *A second-order accurate three sub-step composite algorithm for structural dynamics*, Appl. Math. Model., *77* (2020), 1391–1412.

[17] J. Li, K. Yu, and X. Li, *A novel family of controllably dissipative composite integration algorithms for structural dynamic analysis*, Nonlinear Dyn., *96* (2019), 2475–2507.
[18] H. Mohd Ijam, Z. B. Ibrahim, Z. Abdul Majid, and N. Senu, *Stability analysis of a diagonally implicit scheme of block backward differentiation formula for stiff pharmacokinetics models*, Adv. Differ. Equ., *2020* (2020), 1–22.
[19] H. Mohd Ijam, Z. B. Ibrahim, and I. S. Mohd Zawawi, *Stiffly stable diagonally implicit block backward differentiation formula with adaptive step size strategy for stiff ordinary differential equations*, Matematika, (2024), 27–47.
[20] N. Mohd Husin, I. S. Mohd Zawawi, N. Zainuddin, and Z. B. Ibrahim, *Accuracy improvement of block backward differentiation formulas for solving stiff ordinary differential equations using modified versions of Euler's method*, Math. Stat., *10*(5) (2022), 942–955.
[21] I. S. Mohd Zawawi and Z. B. Ibrahim, *BBDF-Alpha for solving stiff ordinary differential equations with oscillating solutions*, Tamkang J. Math., *51*(2) (2020), 123–136.
[22] I. S. Mohd Zawawi, Z. B. Ibrahim, and K. I. Othman, *Variable step block backward differentiation formula with independent parameter for solving stiff ordinary differential equations*, J. Phys.: Conf. Ser., *1988*(1) (2021), 012031.
[23] H. Musa, M. Suleiman, and N. Senu, *Fully implicit 3-point block extended backward differentiation formula for stiff initial value problems*, Appl. Math. Sci., *6*(85) (2012), 4211–4228.
[24] A. A. Nasarudin, Z. B. Ibrahim, and H. Rosali, *On the integration of stiff ODEs using block backward differentiation formulas of order six*, Symmetry, *12*(6) (2020), 952.
[25] G. Noh and K. J. Bathe, *The Bathe time integration method with controllable spectral radius: the $\rho\infty$-Bathe method*, Comput. Struct., *212* (2019), 299–310.
[26] C. Song and X. Zhang, *High-order composite implicit time integration schemes based on rational approximations for elastodynamics*, Comput. Methods Appl. Mech. Eng., *418* (2024), 116473.
[27] H. Soomro, N. Zainuddin, H. Daud, J. Sunday, N. Jamaludin, A. Abdullah, A. Mulono, and E. A. Kadir, *3-point block backward differentiation formula with an off-step point for the solutions of stiff chemical reaction problems*, J. Math. Chem., *61*(1) (2023), 75–97.
[28] H. Soomro, N. Zainuddin, H. Daud, J. Sunday, N. Jamaludin, A. Abdullah, A. Mulono, and E. A. Kadir, *Variable step block hybrid method for stiff chemical kinetics problems*, Appl. Sci., *12*(9) (2022), 4484.
[29] M. Suleiman, H. Musa, F. Ismail, N. Senu, and Z. B. Ibrahim, *A new superclass of block backward differentiation formula for stiff ordinary differential equations*, Asian-Eur. J. Math., *7*(1) (2014), 1350034.
[30] W. Ying, C. S. Henriquez, and D. J. Rose, *Composite backward differentiation formula: an extension of the TR-BDF2 scheme*, Submitted to Appl. Numer. Math., (2009).
[31] N. Zainuddin, Z. B. Ibrahim, and I. S. Mohd Zawawi, *Diagonal block method for stiff van der Pol equation*, IAENG Int. J. Appl. Math., *53*(1) (2023), 1–8.
[32] L. Zhang, T. Liu, and Q. Li, *A robust and efficient composite time integration algorithm for nonlinear structural dynamic analysis*, Math. Probl. Eng., *2015*(1) (2015), 907023.
[33] H. Zhang and Y. Xing, *Optimization of a class of composite method for structural dynamics*, Comput. Struct., *202* (2018), 60–73.
[34] H. Zhang, R. Zhang, Y. Xing, and P. Masarati, *On the optimization of n-sub-step composite time integration methods*, Nonlinear Dyn., *102*(3) (2020), 1939–1962.