Malaysian Journal of Mathematical Sciences 20(2): 797–825 (2026)
https://doi.org/10.47836/mjms.20.2.18

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

A Convergent and $A-$stable Three-sub-steps Composite Block Backward Differentiation Formula for Solving Stiff Differential Equations

Jaafar, B. A. [1,2] and Mohd Zawawi, I. S.* [1]

[1] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam 40450, Selangor, Malaysia*
[2] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Sarawak, Kampus Samarahan 2, 94300 Kota Samarahan, Sarawak, Malaysia*

E-mail: iskandarshah@uitm.edu.my
**Corresponding author*

Received: 23 June 2025
Accepted: 12 December 2025

Abstract

A novel composite block backward differentiation formula of order four, known as CBBDF(4), has been developed to effectively solve stiff differential equations. This self-starting method approximates solutions by evaluating two points simultaneously at each integration step, incorporating two intermediate points among the interpolating values. The CBBDF(4) scheme is structured through three sub-steps: in the first and second stages, Euler's method computes the intermediate values, setting up the third stage for applying the CBBDF(4) method. The derived method has been established to be convergent and $A-$stable, making it well-suited for solving stiff problems. To validate the method's accuracy and efficiency, several stiff initial value problems are solved using various step sizes. The numerical results are compared with the existing method in terms of maximum error, average error and computational time. Comparative analysis indicates that the new composite block method is not only practical but also successful in delivering reliable results.

Keywords: composite block scheme; linear multi-sub-step; stiff differential equations; three-sub-steps.

1 Introduction

Mathematical models play an essential role in describing and understanding phenomena across various scientific disciplines, including physics [27], chemistry [24], and biology [20]. Many of these models are formulated as initial value problems (IVPs), which involve solving systems of ordinary differential equations (ODEs). Specifically, a system of m first-order ODEs with specified initial conditions can be written in the general form,

$$Y' = F(x, Y), \qquad Y(a) = \eta, \qquad a \leq x \leq b, \tag{1}$$

where

$$Y = [y_1, y_2, \ldots, y_m]^T, \qquad F = [f_1, f_2, \ldots, f_m]^T, \qquad \eta = [\eta_1, \eta_2, \ldots, \eta_m]^T.$$

There is an important type of differential equations, which is the class of stiff differential equations. Stiffness in ODEs is characterized by the presence of rapidly varying solutions that can lead to numerical instability if inappropriate numerical methods or step sizes are used [4]. A system is defined as stiff when the step size required for stability is significantly smaller than that needed for accuracy, often due to components in the system that decay at different rates [1, 18]. Stiff systems typically include terms that induce rapid changes in the solution, which can be mathematically described using the eigenvalues of the system's Jacobian matrix. A large ratio between the largest and smallest eigenvalues indicates greater stiffness [11, 16].

Standard explicit numerical methods often fail to produce stable solutions for stiff differential equations unless extremely small time steps are employed. This instability arises because small errors can grow exponentially, leading to divergent solutions. Therefore, the choice of numerical methods significantly affects the ability to solve stiff ODEs. Implicit methods are generally more suitable for stiff problems because of their $A-$stability properties, which allow for larger step sizes while maintaining stability [17, 23]. One widely used example of such a method is the backward differentiation formula (BDF). A different variation of the BDF method is block backward differentiation formula (BBDF) method introduced by Ibrahim et al. [7]. This method allows the simultaneous computation of multiple solution values at different points, which improves efficiency and accuracy in solving stiff problems. This block method not only reduces the total number of computational steps but also maintains stability and accuracy. Several numerical methods have been recently proposed for solving (1). Notable among these are the BBDF with an off-step point [24], the hybrid BBDF [6], the diagonally implicit BBDF [13], the fixed coefficient BBDF [5], the one-step method with three intermediate points [19], and the hybrid Nyström method [21]. All these methods demonstrate commendable accuracy and computational efficiency.

Another well-known extension of the classical BDF method, widely used in nonlinear structural dynamics, is the class of composite time integration methods, also referred to as multi-sub-step or multi-stage approaches. These methods involve partitioning each time step into multiple sub-steps and applying distinct numerical schemes at each stage [22]. These methods attain up to second-order accuracy by utilizing n sampling points within each time step. However, this increases the dimension of the implicit system by a factor of n compared to the original problem, leading to significantly higher computational costs [28]. Among the earliest contributions is the Bathe method [2], which pioneered the n-sub-step composite framework. This method strategically uses the trapezoidal rule during the initial sub-steps, enabling the effective incorporation of these results into the final sub-steps through a backward difference scheme. Following the introduction of the Bathe method, various types of composite time integration methods have been developed to enhance numerical performance using higher order sub-steps [8, 10].

The TR-BDF2 composite scheme, introduced by Ying et al. [25], combines the trapezoidal rule with the second-order backward differentiation formula (BDF2) to form a singly diagonally implicit Runge-Kutta (SDIRK) method. This scheme offers a self-starting, high-order, one-step alternative to traditional multi-step methods. Building on this, Gao et al. [3] developed two fully implicit methods for solving bidomain equations: the backward Euler method and the composite backward differentiation formula (CBDF2). The CBDF2 method integrates backward Euler as a foundational component with the second-order BDF, ensuring L-stability. This approach enables the use of larger timesteps, delivering improved accuracy and efficiency compared to backward Euler, which requires smaller timesteps to maintain stability. More recently, Junaidi et al. [9] proposed a composite method that combines implicit Euler and second-order BDF with interpolation for solving stiff ODEs, demonstrating improved accuracy relative to the classical BDF. However, their work focused solely on composite schemes within single-step methods and did not extend to multi-step block methods for stiff ODEs. To date, few studies have explored the integration of composite techniques within the frameworks of multi-step block methods, which hold potential for greater accuracy and stability in solving (1). This gap highlights a significant limitation in current numerical methods for stiff ODEs. Motivated by these insights, the present study aims to derive the self-starting composite block backward differentiation formula of order four, CBBDF(4), to solve stiff ODEs with enhanced accuracy and stability. The proposed method not only produces a block of approximations at two points simultaneously but also converges with larger integration steps.

This paper is organized as follows. In Section 2, the governing formulations of the CBBDF(4) is presented. Then, the order of the method and its properties, including convergence and stability, are discussed in Section 3. The detailed implementation of the method is provided in Section 4. Numerical simulations for linear and nonlinear of first order stiff IVPs are conducted in Section 5. Lastly, the conclusions are drawn in Section 6.

2 Formulation of the Method

In this work, the CBBDF(4) scheme is developed through three sequential sub-steps. In the first and second stages, Euler's method is utilized to calculate the intermediate points in the previous block. These calculated points are then used to implement the predictor-corrector method of two-point CBBDF(4), which has an order of four, in the third stage, as illustrated in Figure 1.

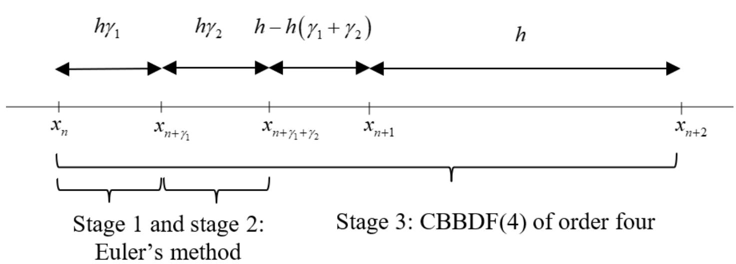

Figure 1: The interpolating points of the CBBDF(4).

The step size for the first point is split into three sub-steps: the intervals, $[x_n, x_{n+\gamma_1}]$, $[x_{n+\gamma_1}, x_{n+\gamma_1+\gamma_2}]$ and $[x_{n+\gamma_1+\gamma_2}, x_{n+1}]$; in contrast, the second point employs a constant step size over the interval $[x_{n+1}, x_{n+2}]$, where h denotes the integration step size, and the parameters γ_1 and γ_2 serve as independent parameters at the intermediate points $x_{n+\gamma_1}$ and $x_{n+\gamma_1+\gamma_2}$. In the context of interpolation, the CBBDF(4) method's strategy involves utilizing intermediate points within the set of interpolating points. This approach enhances the accuracy and stability of the solution by considering additional points between the primary interpolation nodes. By incorporating these intermediate points, the method can better capture the behavior of the underlying function, leading to more precise interpolations. The composite block scheme to be formulated is a k-linear multi-sub-step method (LMM) defined as follows,

$$\sum_{j=0}^{k-3} A_j y_{n+j} + A_{\gamma_1} y_{n+\gamma_1} + A_{\gamma_1+\gamma_2} y_{n+\gamma_1+\gamma_2} = h\left[\sum_{j=0}^{k-3} B_j f_{n+j} + B_{\gamma_1} f_{n+\gamma_1} + B_{\gamma_1+\gamma_2} f_{n+\gamma_1+\gamma_2}\right]. \tag{2}$$

The coefficients A_j, A_{γ_1}, $A_{\gamma_1+\gamma_2}$, B_j, B_{γ_1}, and $B_{\gamma_1+\gamma_2}$ are constants that satisfy the conditions $A_{k-3} \neq 0$ and that not both A_0 and B_0 are zero simultaneously. The LMM described in (2) is classified as explicit when $B_{k-3} = 0$, and it is considered implicit if $B_{k-3} \neq 0$.

The local approximation of CBBDF(4) is obtained using the Lagrange interpolating polynomial, $P_k(x)$ of degree $k = 5$ by considering two intermediate points in the interpolating among the interpolation nodes. The general expression for the Lagrange interpolating polynomial is given by,

$$P_k(x) = \sum_{j=0}^{k-3} L_{k,j}(x)\, y(x_{n+2-j}) + L_{k,\gamma_1}(x)\, y(x_{n+\gamma_1}) + L_{k,\gamma_1+\gamma_2}(x)\, y(x_{n+\gamma_1+\gamma_2}), \tag{3}$$

whereas

$$L_{k,j}(x) = \left(\prod_{i=0, i\neq j}^{k-3} \frac{(x - x_{n+2-i})}{(x_{n+2-j} - x_{n+2-i})} \cdot \frac{(x - x_{n+\gamma_1})}{(x_{n+2-j} - x_{n+\gamma_1})} \cdot \frac{(x - x_{n+\gamma_1+\gamma_2})}{(x_{n+2-j} - x_{n+\gamma_1+\gamma_2})}\right) \quad \text{and}$$

$$L_{k,\gamma_1}(x) = \prod_{i=0}^{k-3} \frac{(x - x_{n+2-i})}{(x_{n+\gamma_1} - x_{n+2-i})}, \quad L_{k,\gamma_1+\gamma_2}(x) = \prod_{i=0}^{k-3} \frac{(x - x_{n+2-i})}{(x_{n+\gamma_1+\gamma_2} - x_{n+2-i})},$$

for $j = 0, \ldots, k-3$.

2.1 Derivation of the method for the first and second stages

To calculate the intermediate values $y_{n+\gamma_1}$ and $y_{n+\gamma_1+\gamma_2}$, the process involves two stages of interpolation based on the principles of Euler's method,

Stage 1: Interpolating between points x_n and $x_{n+\gamma_1}$.
Stage 2: Interpolating between points $x_{n+\gamma_1}$ and $x_{n+\gamma_1+\gamma_2}$.

The resulting polynomial from these interpolations can be expressed as follows,

$$\begin{aligned} G_1(x) &= \frac{(x - x_{n+\gamma_1})}{(x_n - x_{n+\gamma_1})} y_n + \frac{(x - x_n)}{(x_{n+\gamma_1} - x_n)} y_{n+\gamma_1}, \\ G_2(x) &= \frac{(x - x_{n+\gamma_1+\gamma_2})}{(x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2})} y_{n+\gamma_1} + \frac{(x - x_{n+\gamma_1})}{(x_{n+\gamma_1+\gamma_2} - x_{n+\gamma_1})} y_{n+\gamma_1+\gamma_2}, \end{aligned} \tag{4}$$

where $G_1(x)$ and $G_2(x)$ are the interpolations for the first and second stages, respectively. Replacing $x = sh + x_{n+\gamma_1}$ into (4) yields the following results,

$$\begin{aligned} G_1\left(sh + x_{n+\gamma_1}\right) &= \frac{\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1}\right)}{\left(x_n - x_{n+\gamma_1}\right)} y_n + \frac{\left(sh + x_{n+\gamma_1} - x_n\right)}{\left(x_{n+\gamma_1} - x_n\right)} y_{n+\gamma_1}, \\ G_2\left(sh + x_{n+\gamma_1}\right) &= \frac{\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)}{\left(x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)} y_{n+\gamma_1} + \frac{\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1}\right)}{\left(x_{n+\gamma_1+\gamma_2} - x_{n+\gamma_1}\right)} y_{n+\gamma_1+\gamma_2}. \end{aligned} \tag{5}$$

Once the Lagrange basis polynomials in (5) are simplified, differentiating them with respect to s will lead to

$$\begin{aligned} G_1'\left(sh + x_{n+\gamma_1}\right) &= -\left(\frac{1}{\gamma_1}\right) y_n + \left(\frac{1}{\gamma_1}\right) y_{n+\gamma_1}, \\ G_2'\left(sh + x_{n+\gamma_1}\right) &= -\left(\frac{1}{\gamma_2}\right) y_{n+\gamma_1} + \left(\frac{1}{\gamma_2}\right) y_{n+\gamma_1+\gamma_2}. \end{aligned} \tag{6}$$

Evaluating (6) at $s = -\gamma_1$ and $s = 0$, respectively; followed by considering $hf_n = G_1'\left(x_n\right)$ and $hf_{n+\gamma_1} = G_2'\left(x_{n+\gamma_1}\right)$ will produce the Stage 1 and Stage 2 formulas,

$$\begin{aligned} y_{n+\gamma_1} &= y_n + \gamma_1 h f_n, \\ y_{n+\gamma_1+\gamma_2} &= y_{n+\gamma_1} + \gamma_2 h f_{n+\gamma_1}. \end{aligned} \tag{7}$$

2.2 Derivation of the method for the third stagen

The approximation solutions of y_{n+1} and y_{n+2} are determined using the points x_n, $x_{n+\gamma_1}$, $x_{n+\gamma_1+\gamma_2}$, x_{n+1} and x_{n+2}. The Lagrange interpolating polynomial has the form,

$$\begin{aligned} P\left(x\right) = &\frac{\left(x - x_{n+\gamma_1}\right)\left(x - x_{n+\gamma_1+\gamma_2}\right)\left(x - x_{n+1}\right)\left(x - x_{n+2}\right)}{\left(x_n - x_{n+\gamma_1}\right)\left(x_n - x_{n+\gamma_1+\gamma_2}\right)\left(x_n - x_{n+1}\right)\left(x_n - x_{n+2}\right)} y_n \\ &+ \frac{\left(x - x_n\right)\left(x - x_{n+\gamma_1+\gamma_2}\right)\left(x - x_{n+1}\right)\left(x - x_{n+2}\right)}{\left(x_{n+\gamma_1} - x_n\right)\left(x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)\left(x_{n+\gamma_1} - x_{n+1}\right)\left(x_{n+\gamma_1} - x_{n+2}\right)} y_{n+\gamma_1} \\ &+ \frac{\left(x - x_n\right)\left(x - x_{n+\gamma_1}\right)\left(x - x_{n+1}\right)\left(x - x_{n+2}\right)}{\left(x_{n+\gamma_1+\gamma_2} - x_n\right)\left(x_{n+\gamma_1+\gamma_2} - x_{n+\gamma_1}\right)\left(x_{n+\gamma_1+\gamma_2} - x_{n+1}\right)\left(x_{n+\gamma_1+\gamma_2} - x_{n+2}\right)} y_{n+\gamma_1+\gamma_2} \\ &+ \frac{\left(x - x_n\right)\left(x - x_{n+\gamma_1}\right)\left(x - x_{n+\gamma_1+\gamma_2}\right)\left(x - x_{n+2}\right)}{\left(x_{n+1} - x_n\right)\left(x_{n+1} - x_{n+\gamma_1}\right)\left(x_{n+1} - x_{n+\gamma_1+\gamma_2}\right)\left(x_{n+1} - x_{n+2}\right)} y_{n+1} \\ &+ \frac{\left(x - x_n\right)\left(x - x_{n+\gamma_1}\right)\left(x - x_{n+\gamma_1+\gamma_2}\right)\left(x - x_{n+1}\right)}{\left(x_{n+2} - x_n\right)\left(x_{n+2} - x_{n+\gamma_1}\right)\left(x_{n+2} - x_{n+\gamma_1+\gamma_2}\right)\left(x_{n+2} - x_{n+1}\right)} y_{n+2}. \end{aligned} \tag{8}$$

Substitute $x = sh + x_{n+\gamma_1}$ into (8) yields,

$$
\begin{aligned}
&P\left(sh + x_{n+\gamma_1}\right) \\
&= \frac{\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1}\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)\left(sh + x_{n+\gamma_1} - x_{n+1}\right)\left(sh + x_{n+\gamma_1} - x_{n+2}\right)}{\left(x_n - x_{n+\gamma_1}\right)\left(x_n - x_{n+\gamma_1+\gamma_2}\right)\left(x_n - x_{n+1}\right)\left(x_n - x_{n+2}\right)} y_n \\
&\quad + \frac{\left(sh + x_{n+\gamma_1} - x_n\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)\left(sh + x_{n+\gamma_1} - x_{n+1}\right)\left(sh + x_{n+\gamma_1} - x_{n+2}\right)}{\left(x_{n+\gamma_1} - x_n\right)\left(x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)\left(x_{n+\gamma_1} - x_{n+1}\right)\left(x_{n+\gamma_1} - x_{n+2}\right)} y_{n+\gamma_1} \\
&\quad + \frac{\left(sh + x_{n+\gamma_1} - x_n\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1}\right)\left(sh + x_{n+\gamma_1} - x_{n+1}\right)\left(sh + x_{n+\gamma_1} - x_{n+2}\right)}{\left(x_{n+\gamma_1+\gamma_2} - x_n\right)\left(x_{n+\gamma_1+\gamma_2} - x_{n+\gamma_1}\right)\left(x_{n+\gamma_1+\gamma_2} - x_{n+1}\right)\left(x_{n+\gamma_1+\gamma_2} - x_{n+2}\right)} y_{n+\gamma_1+\gamma_2} \\
&\quad + \frac{\left(sh + x_{n+\gamma_1} - x_n\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1}\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)\left(sh + x_{n+\gamma_1} - x_{n+2}\right)}{\left(x_{n+1} - x_n\right)\left(x_{n+1} - x_{n+\gamma_1}\right)\left(x_{n+1} - x_{n+\gamma_1+\gamma_2}\right)\left(x_{n+1} - x_{n+2}\right)} y_{n+1} \\
&\quad + \frac{\left(sh + x_{n+\gamma_1} - x_n\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1}\right)\left(sh + x_{n+\gamma_1} - x_{n+\gamma_1+\gamma_2}\right)\left(sh + x_{n+\gamma_1} - x_{n+1}\right)}{\left(x_{n+2} - x_n\right)\left(x_{n+2} - x_{n+\gamma_1}\right)\left(x_{n+2} - x_{n+\gamma_1+\gamma_2}\right)\left(x_{n+2} - x_{n+1}\right)} y_{n+2}.
\end{aligned} \tag{9}
$$

Taking the first derivative of (9) with respect to s, and subsequently replacing s by $1 - \gamma_1$ and $2 - \gamma_1$, respectively, leads to the following equations,

$$
\begin{aligned}
P'\left(x_{n+1}\right) &= \left(\frac{2\gamma_1 + \gamma_2 - \gamma_1^2 - \gamma_1\gamma_2 - 1}{2\gamma_1\left(\gamma_1 + \gamma_2\right)}\right) y_n + \left(\frac{1 - \gamma_1 - \gamma_2}{\gamma_1\gamma_2\left(\gamma_1 - 1\right)\left(\gamma_1 - 2\right)}\right) y_{n+\gamma_1} \\
&\quad + \left(\frac{\gamma_1 - 1}{\left(\gamma_1 + \gamma_2\right)\gamma_2\left(\gamma_1 + \gamma_2 - 1\right)\left(\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+\gamma_1+\gamma_2} \\
&\quad + \left(\frac{2 - 2\gamma_1 - \gamma_2}{\left(\gamma_1 - 1\right)\left(\gamma_1 + \gamma_2 - 1\right)}\right) y_{n+1} + \left(\frac{1 - 2\gamma_1 + \gamma_1^2 - \gamma_2 + \gamma_1\gamma_2}{2\left(\gamma_1 - 2\right)\left(\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+2},
\end{aligned} \tag{10}
$$

and

$$
\begin{aligned}
P'\left(x_{n+2}\right) &= \left(\frac{4 - 4\gamma_1 - 2\gamma_2 + \gamma_1^2 + \gamma_1\gamma_2}{2\gamma_1\left(\gamma_1 + \gamma_2\right)}\right) y_n + \left(\frac{2\left(\gamma_1 + \gamma_2 - 2\right)}{\gamma_1\gamma_2\left(\gamma_1 - 1\right)\left(\gamma_1 - 2\right)}\right) y_{n+\gamma_1} \\
&\quad + \left(\frac{2\left(2 - \gamma_1\right)}{\left(\gamma_1 + \gamma_2\right)\gamma_2\left(\gamma_1 + \gamma_2 - 1\right)\left(\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+\gamma_1+\gamma_2} \\
&\quad + \left(\frac{2\left(4\gamma_1 + 2\gamma_2 - \gamma_1^2 - \gamma_1\gamma_2 - 4\right)}{\left(\gamma_1 - 1\right)\left(\gamma_1 + \gamma_2 - 1\right)}\right) y_{n+1} \\
&\quad + \left(\frac{20 - 16\gamma_1 + 3\gamma_1^2 - 8\gamma_2 + 3\gamma_1\gamma_2}{2\left(\gamma_1 - 2\right)\left(\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+2}.
\end{aligned} \tag{11}
$$

Consider $hf_{n+1,n+2} = P'\left(x_{n+1,n+2}\right)$ in (10) and (11) gives,

$$
\begin{aligned}
y_{n+1}^{(c)} &= \left(\frac{\left(2\gamma_1 + \gamma_2 - \gamma_1^2 - \gamma_1\gamma_2 - 1\right)\left(\gamma_1 - 1\right)\left(\gamma_1 + \gamma_2 - 1\right)}{2\gamma_1\left(\gamma_1 + \gamma_2\right)\left(2\gamma_1 + \gamma_2 - 2\right)}\right) y_n \\
&\quad + \left(\frac{\left(1 - \gamma_1 - \gamma_2\right)\left(\gamma_1 + \gamma_2 - 1\right)}{\gamma_1\gamma_2\left(\gamma_1 - 2\right)\left(2\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+\gamma_1} \\
&\quad + \left(\frac{\left(\gamma_1 - 1\right)\left(\gamma_1 - 1\right)}{\left(\gamma_1 + \gamma_2\right)\gamma_2\left(\gamma_1 + \gamma_2 - 2\right)\left(2\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+\gamma_1+\gamma_2} \\
&\quad + \left(\frac{\left(1 - 2\gamma_1 + \gamma_1^2 - \gamma_2 + \gamma_1\gamma_2\right)\left(\gamma_1 - 1\right)\left(\gamma_1 + \gamma_2 - 1\right)}{2\left(\gamma_1 - 2\right)\left(\gamma_1 + \gamma_2 - 2\right)\left(2\gamma_1 + \gamma_2 - 2\right)}\right) y_{n+2} \\
&\quad + \left(\frac{\left(\gamma_1 - 1\right)\left(\gamma_1 + \gamma_2 - 1\right)}{2 - 2\gamma_1 - \gamma_2}\right) hf_{n+1},
\end{aligned} \tag{12}
$$

and

$$
\begin{aligned}
y_{n+2}^{(c)} = &\left(\frac{\left(4\gamma_1+2\gamma_2-\gamma_1^2-\gamma_1\gamma_2-4\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{\gamma_1(\gamma_1+\gamma_2)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_n\\
&+\left(\frac{4(2-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-2)}{\gamma_1\gamma_2(\gamma_1-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_{n+\gamma_1}\\
&+\left(\frac{4(\gamma_1-2)(\gamma_1-2)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_{n+\gamma_1+\gamma_2}\\
&+\left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_{n+1}\\
&+\left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right)hf_{n+2}.
\end{aligned}
\tag{13}
$$

The CBBDF(4) method is implemented through a predictor-corrector approach. To determine the solutions for $y_{n+1}^{(c)}$ and $y_{n+2}^{(c)}$, as derived in (12) and (13), it is essential to first obtain the predicted values. The predictor formulas are constructed from the points x_n, $x_{n+\gamma_1}$ and $x_{n+\gamma_1+\gamma_2}$. The polynomial can be expressed as,

$$
\begin{aligned}
Q(x) = &\frac{(x-x_{n+\gamma_1})(x-x_{n+\gamma_1+\gamma_2})}{(x_n-x_{n+\gamma_1})(x_n-x_{n+\gamma_1+\gamma_2})}y_n + \frac{(x-x_n)(x-x_{n+\gamma_1+\gamma_2})}{(x_{n+\gamma_1}-x_n)(x_{n+\gamma_1}-x_{n+\gamma_1+\gamma_2})}y_{n+\gamma_1}\\
&+\frac{(x-x_n)(x-x_{n+\gamma_1})}{(x_{n+\gamma_1+\gamma_2}-x_n)(x_{n+\gamma_1+\gamma_2}-x_{n+\gamma_1})}y_{n+\gamma_1+\gamma_2}.
\end{aligned}
\tag{14}
$$

The formulation of (14) resembles the derivation of (12) and (13), but it excludes the differentiation step. As a result, the predictor formulas for $y_{n+1}^{(p)}$ and $y_{n+2}^{(p)}$ take the form,

$$
\begin{aligned}
y_{n+1}^{(p)} &= \left(\frac{1-2\gamma_1-\gamma_2+\gamma_1^2+\gamma_1\gamma_2}{\gamma_1(\gamma_1+\gamma_2)}\right)y_n + \left(\frac{\gamma_1+\gamma_2-1}{\gamma_1\gamma_2}\right)y_{n+\gamma_1}\\
&\quad+\left(\frac{\gamma_1-1}{\gamma_2(\gamma_1+\gamma_2)}\right)y_{n+\gamma_1+\gamma_2},\\
y_{n+2}^{(p)} &= \left(\frac{4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2}{\gamma_1(\gamma_1+\gamma_2)}\right)y_n + \left(\frac{2(\gamma_1+\gamma_2-2)}{\gamma_1\gamma_2}\right)y_{n+\gamma_1}\\
&\quad+\left(\frac{2(\gamma_1-2)}{\gamma_2(\gamma_1+\gamma_2)}\right)y_{n+\gamma_1+\gamma_2}.
\end{aligned}
\tag{15}
$$

3 Properties of the Method

This section focuses on discussing the properties of the CBBDF(4) method. Key aspects include the order of the developed method, as well as its zero-stability and consistency: the essential conditions for the method's convergence. Additionally, the region of absolute stability of the CBBDF(4) method will be examined to assess its effectiveness in solving stiff ODEs.

3.1 Order and error constant

Definition 3.1. *The LMM associated with the linear difference operator is said to be of order q if $C_0 = C_1 = \cdots = C_q = 0$ and the error constant, $C_{q+1} \neq 0$.*

To assess the consistency of the method, the order of the method will first be determined. By rearranging (12) and (13), the following equation is obtained,

$$\begin{aligned}
y_{n+1} &- \left(\frac{\left(2\gamma_1+\gamma_2-\gamma_1^2-\gamma_1\gamma_2-1\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2\gamma_1(\gamma_1+\gamma_2)(2\gamma_1+\gamma_2-2)}\right) y_n \\
&- \left(\frac{(1-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-1)}{\gamma_1\gamma_2(\gamma_1-2)(2\gamma_1+\gamma_2-2)}\right) y_{n+\gamma_1} \\
&- \left(\frac{(\gamma_1-1)(\gamma_1-1)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right) y_{n+\gamma_1+\gamma_2} \\
&- \left(\frac{\left(1-2\gamma_1+\gamma_1^2-\gamma_2+\gamma_1\gamma_2\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2(\gamma_1-2)(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right) y_{n+2} \\
&= \left(\frac{(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2-2\gamma_1-\gamma_2}\right) hf_{n+1},
\end{aligned} \tag{16}$$

$$\begin{aligned}
y_{n+2} &- \left(\frac{\left(4\gamma_1+2\gamma_2-\gamma_1^2-\gamma_1\gamma_2-4\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{\gamma_1(\gamma_1+\gamma_2)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right) y_n \\
&- \left(\frac{4(2-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-2)}{\gamma_1\gamma_2(\gamma_1-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right) y_{n+\gamma_1} \\
&- \left(\frac{4(\gamma_1-2)(\gamma_1-2)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right) y_{n+\gamma_1+\gamma_2} \\
&- \left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right) y_{n+1} \\
&= \left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right) hf_{n+2}.
\end{aligned} \tag{17}$$

The matrix form of (16) and (17) is given by,

$$\begin{aligned}
&\begin{bmatrix} 0 & A_{0,1} \\ 0 & A_{0,2} \end{bmatrix}\begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} + \begin{bmatrix} A_{\gamma_1,1} & A_{\gamma_1+\gamma_2,1} \\ A_{\gamma_1,2} & A_{\gamma_1+\gamma_2,2} \end{bmatrix}\begin{bmatrix} y_{n+\gamma_1} \\ y_{n+\gamma_1+\gamma_2} \end{bmatrix} + \begin{bmatrix} A_{1,1} & A_{2,1} \\ A_{1,2} & A_{2,2} \end{bmatrix}\begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} \\
&= h\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix} + h\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} f_{n+\gamma_1} \\ f_{n+\gamma_1+\gamma_2} \end{bmatrix} + h\begin{bmatrix} B_{1,1} & B_{2,1} \\ B_{1,2} & B_{2,2} \end{bmatrix}\begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix},
\end{aligned} \tag{18}$$

where

$$A_0 = \begin{bmatrix} A_{0,1} \\ A_{0,2} \end{bmatrix} = \begin{bmatrix} -\left(\frac{\left(2\gamma_1+\gamma_2-\gamma_1^2-\gamma_1\gamma_2-1\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2\gamma_1(\gamma_1+\gamma_2)(2\gamma_1+\gamma_2-2)}\right) \\ -\left(\frac{\left(4\gamma_1+2\gamma_2-\gamma_1^2-\gamma_1\gamma_2-4\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{\gamma_1(\gamma_1+\gamma_2)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) \end{bmatrix},$$

$$A_{\gamma_1} = \begin{bmatrix} A_{\gamma_1,1} \\ A_{\gamma_1,2} \end{bmatrix} = \begin{bmatrix} -\left(\frac{(1-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-1)}{\gamma_1\gamma_2(\gamma_1-2)(2\gamma_1+\gamma_2-2)}\right) \\ -\left(\frac{4(2-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-2)}{\gamma_1\gamma_2(\gamma_1-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) \end{bmatrix},$$

$$
\begin{aligned}
A_{\gamma_1+\gamma_2} &= \begin{bmatrix} A_{\gamma_1+\gamma_2,1} \\ A_{\gamma_1+\gamma_2,2} \end{bmatrix} = \begin{bmatrix} -\left(\frac{(\gamma_1-1)(\gamma_1-1)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right) \\ -\left(\frac{4(\gamma_1-2)(\gamma_1-2)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) \end{bmatrix}, \\
A_1 &= \begin{bmatrix} A_{1,1} \\ A_{1,2} \end{bmatrix} = \begin{bmatrix} 1 \\ -\left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) \end{bmatrix}, \\
A_2 &= \begin{bmatrix} A_{2,1} \\ A_{2,2} \end{bmatrix} = \begin{bmatrix} -\left(\frac{\left(1-2\gamma_1+\gamma_1^2-\gamma_2+\gamma_1\gamma_2\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2(\gamma_1-2)(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right) \\ 1 \end{bmatrix}, \\
B_0 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad B_{\gamma_1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad B_{\gamma_1+\gamma_2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
B_1 &= \begin{bmatrix} B_{1,1} \\ B_{1,2} \end{bmatrix} = \begin{bmatrix} \left(\frac{(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2-2\gamma_1-\gamma_2}\right) \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} B_{2,1} \\ B_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \\ \left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right) \end{bmatrix}.
\end{aligned}
$$

The linear difference, L_i corresponding to the LMM in (2) can be expressed as,

$$
L\left[y\left(x\right);h\right] = \sum_j \left[A_j y\left(x+jh\right) - hB_j y'\left(x+jh\right)\right], \tag{19}
$$

where $j = 0, \gamma_1, \gamma_1+\gamma_2, 1, 2$. By expanding the function $y\left(x+jh\right)$ and its derivative $y'\left(x+jh\right)$ in (19) around x using Taylor's method, the following series is obtained,

$$
\begin{aligned}
L\left[y\left(x\right),h\right] &= \sum_j A_j \left[y\left(x\right) + \frac{jh}{1!}y'\left(x\right) + \frac{\left(jh\right)^2}{2!}y''\left(x\right) + \frac{\left(jh\right)^3}{3!}y'''\left(x\right) + \frac{\left(jh\right)^4}{4!}y^{(4)}\left(x\right) + \ldots\right] \\
&\quad - \sum_j hB_j \left[y'\left(x\right) + \frac{jh}{1!}y''\left(x\right) + \frac{\left(jh\right)^2}{2!}y'''\left(x\right) + \frac{\left(jh\right)^3}{3!}y^{(4)}\left(x\right) + \ldots\right] \\
&= \sum_j \left[A_j\right] y\left(x\right) + \sum_j \left[jA_j - B_j\right] hy'\left(x\right) + \sum_j \left[\frac{1}{2!}j^2A_j - jB_j\right] h^2y''\left(x\right) \\
&\quad + \sum_j \left[\frac{1}{3!}j^3A_j - \frac{1}{2!}j^2B_j\right] h^3y'''\left(x\right) + \sum_j \left[\frac{1}{4!}j^4A_j - \frac{1}{3!}j^3B_j\right] h^4y^{(4)}\left(x\right) + \ldots.
\end{aligned} \tag{20}
$$

By denoting the collected terms of $y\left(x\right)$ and its derivative in (20) as C_q, where $q = 0, 1, 2, \ldots$, the order of the method and the error constant can be determined by substituting the associated set of coefficients from (18) to achieve as follows,

$$
\begin{aligned}
C_0 &= A_0 + A_{\gamma_1} + A_{\gamma_1+\gamma_2} + A_1 + A_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_1 &= \left[(0\cdot(A_0)) + (\gamma_1\cdot(A_{\gamma_1})) + ((\gamma_1+\gamma_2)\cdot(A_{\gamma_1+\gamma_2})) + (1\cdot(A_1)) + (2\cdot(A_2))\right] \\
&\quad - \left[B_0 + B_{\gamma_1} + B_{\gamma_1+\gamma_2} + B_1 + B_2\right] \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_2 &= \frac{1}{2}\left[\left(0^2\cdot(A_0)\right) + \left({\gamma_1}^2\cdot(A_{\gamma_1})\right) + \left((\gamma_1+\gamma_2)^2\cdot(A_{\gamma_1+\gamma_2})\right) + \left(1^2\cdot(A_1)\right) + \left(2^2\cdot(A_2)\right)\right] \\
&\quad - \left[(0\cdot(B_0)) + (\gamma_1\cdot(B_{\gamma_1})) + ((\gamma_1+\gamma_2)\cdot(B_{\gamma_1+\gamma_2})) + (1\cdot(B_1)) + (2\cdot(B_2))\right] \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
C_3 &= \frac{1}{6}\Big[\big(0^3 \cdot (A_0)\big) + \big({\gamma_1}^3 \cdot (A_{\gamma_1})\big) + \Big((\gamma_1+\gamma_2)^3 \cdot (A_{\gamma_1+\gamma_2})\Big) + \big(1^3 \cdot (A_1)\big) + \big(2^3 \cdot (A_2)\big)\Big] \\
&\quad - \frac{1}{2}\Big[\big(0^2 \cdot (B_0)\big) + \big({\gamma_1}^2 \cdot (B_{\gamma_1})\big) + \Big((\gamma_1+\gamma_2)^2 \cdot (B_{\gamma_1+\gamma_2})\Big) + \big(1^2 \cdot (B_1)\big) + \big(2^2 \cdot (B_2)\big)\Big] \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
C_4 &= \frac{1}{24}\Big[\big(0^4 \cdot (A_0)\big) + \big({\gamma_1}^4 \cdot (A_{\gamma_1})\big) + \Big((\gamma_1+\gamma_2)^4 \cdot (A_{\gamma_1+\gamma_2})\Big) + \big(1^4 \cdot (A_1)\big) + \big(2^4 \cdot (A_2)\big)\Big] \\
&\quad - \frac{1}{6}\Big[\big(0^3 \cdot (B_0)\big) + \big({\gamma_1}^3 \cdot (B_{\gamma_1})\big) + \Big((\gamma_1+\gamma_2)^3 \cdot (B_{\gamma_1+\gamma_2})\Big) + \big(1^3 \cdot (B_1)\big) + \big(2^3 \cdot (B_2)\big)\Big] \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
C_5 &= \frac{1}{120}\Big[\big(0^5 \cdot (A_0)\big) + \big({\gamma_1}^5 \cdot (A_{\gamma_1})\big) + \Big((\gamma_1+\gamma_2)^5 \cdot (A_{\gamma_1+\gamma_2})\Big) + \big(1^5 \cdot (A_1)\big) + \big(2^5 \cdot (A_2)\big)\Big] \\
&\quad - \frac{1}{24}\Big[\big(0^4 \cdot (B_0)\big) + \big({\gamma_1}^4 \cdot (B_{\gamma_1})\big) + \Big((\gamma_1+\gamma_2)^4 \cdot (B_{\gamma_1+\gamma_2})\Big) + \big(1^4 \cdot (B_1)\big) + \big(2^4 \cdot (B_2)\big)\Big] \\
&= \begin{bmatrix} -\frac{1}{120}\left(\frac{6\gamma_1^2+6\gamma_1\gamma_2-4\gamma_1-2\gamma_2+1-4\gamma_1^3-6\gamma_1^2\gamma_2+\gamma_1^4+2\gamma_1^3\gamma_2-2\gamma_1\gamma_2^2+\gamma_2^2+\gamma_1^2\gamma_2^2}{2\gamma_1+\gamma_2-2}\right) \\ -\frac{1}{30}\left(\frac{24\gamma_1^2+24\gamma_1\gamma_2-32\gamma_1-16\gamma_2+16-8\gamma_1^3-12\gamma_1^2\gamma_2-4\gamma_1\gamma_2^2+4\gamma_2^2+\gamma_1^4+2\gamma_1^3\gamma_2+\gamma_1^2\gamma_2^2}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right) \end{bmatrix}.
\end{aligned}
$$

Based on Definition 3.1, it is found that $C_0 = C_1 = C_2 = C_3 = C_4 = [0,0,0,0,0]^T$ and $C_5 \neq 0$, indicating that the CBBDF(4) method has an order of four with error constant as stated in C_5.

3.2 Convergence properties

In this section, the convergence of the CBBDF(4) is investigated. To establish the convergence of the method, it is necessary for it to satisfy the criteria of consistency and zero-stability, as outlined in the following definitions.

Definition 3.2. *The LMM is considered consistent if it has an order of $q \geq 1$.*

Since the CBBDF(4) method is of order four with $q = 4 \geq 1$, therefore it is consistent by Definition 3.2. Next, the zero-stability associated with CBBDF(4) is discussed.

Definition 3.3. *The LMM is considered zero-stable if all roots of the first stability polynomial lie within or on the unit circle, with no root having a modulus exceeding one, and any root located exactly on the unit circle is simple.*

The stability polynomial for the CBBDF(4) method is derived using a linear differential equation test represented as,

$$y' = \lambda y, \tag{21}$$

where $\lambda < 0$ is complex. By substituting (21) into (12) and (13) will lead to

$$
\begin{aligned}
&\left(1-\left(\frac{(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2-2\gamma_1-\gamma_2}\right)h\lambda\right)y_{n+1}\\
&-\left(\frac{\left(1-2\gamma_1+\gamma_1^2-\gamma_2+\gamma_1\gamma_2\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2(\gamma_1-2)(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right)y_{n+2}\\
&=\left(\frac{\left(2\gamma_1+\gamma_2-\gamma_1^2-\gamma_1\gamma_2-1\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2\gamma_1(\gamma_1+\gamma_2)(2\gamma_1+\gamma_2-2)}\right)y_n\\
&\quad+\left(\frac{(1-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-1)}{\gamma_1\gamma_2(\gamma_1-2)(2\gamma_1+\gamma_2-2)}\right)y_{n+\gamma_1}\\
&\quad+\left(\frac{(\gamma_1-1)(\gamma_1-1)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right)y_{n+\gamma_1+\gamma_2},
\end{aligned}
\tag{22}
$$

$$
\begin{aligned}
&\left(1-\left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right)h\lambda\right)y_{n+2}\\
&-\left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_{n+1}\\
&=\left(\frac{\left(4\gamma_1+2\gamma_2-\gamma_1^2-\gamma_1\gamma_2-4\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{\gamma_1(\gamma_1+\gamma_2)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_n\\
&\quad+\left(\frac{4(2-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-2)}{\gamma_1\gamma_2(\gamma_1-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_{n+\gamma_1}\\
&\quad+\left(\frac{4(\gamma_1-2)(\gamma_1-2)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-1)(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2)}\right)y_{n+\gamma_1+\gamma_2}.
\end{aligned}
\tag{23}
$$

Equations (22) and (23) can be expressed in matrix form, and by letting $\bar{h}=\lambda h$ generates,

$$
\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}y_{n+1}\\ y_{n+2}\end{bmatrix}=\begin{bmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{bmatrix}\begin{bmatrix}y_{n+\gamma_1}\\ y_{n+\gamma_1+\gamma_2}\end{bmatrix}+\begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix}\begin{bmatrix}y_{n-1}\\ y_n\end{bmatrix},
\tag{24}
$$

where

$$
\begin{aligned}
a&=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\\
&=\begin{bmatrix}1-\left(\frac{(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2-2\gamma_1-\gamma_2}\right)\bar{h} & -\left(\frac{\left(1-2\gamma_1+\gamma_1^2-\gamma_2+\gamma_1\gamma_2\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2(\gamma_1-2)(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right)\\ -\left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) & 1-\left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right)\bar{h}\end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
b&=\begin{bmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{bmatrix}\\
&=\begin{bmatrix}\left(\frac{(1-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-1)}{\gamma_1\gamma_2(\gamma_1-2)(2\gamma_1+\gamma_2-2)}\right) & \left(\frac{(\gamma_1-1)(\gamma_1-1)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right)\\ \left(\frac{4(2-\gamma_1-\gamma_2)(\gamma_1+\gamma_2-2)}{\gamma_1\gamma_2(\gamma_1-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) & \left(\frac{4(\gamma_1-2)(\gamma_1-2)}{(\gamma_1+\gamma_2)\gamma_2(\gamma_1+\gamma_2-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right)\end{bmatrix},
\end{aligned}
$$

$$
c=\begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix}=\begin{bmatrix}0 & \left(\frac{\left(2\gamma_1+\gamma_2-\gamma_1^2-\gamma_1\gamma_2-1\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2\gamma_1(\gamma_1+\gamma_2)(2\gamma_1+\gamma_2-2)}\right)\\ 0 & \left(\frac{\left(4\gamma_1+2\gamma_2-\gamma_1^2-\gamma_1\gamma_2-4\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{\gamma_1(\gamma_1+\gamma_2)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right)\end{bmatrix}.
$$

Solve $R\left(t, \bar{h}, \gamma_1, \gamma_2\right) = \det\left(at^2 - bt - c\right)$ and set $\bar{h} = 0$ to obtain the stability polynomial for (24). Then, solving for t gives,

$$
\begin{aligned}
&t_1 = 0, \qquad t_2 = 1, \\
&t_{3,4} = \pm \frac{\left(\begin{array}{l}
\pm\,(-12\gamma_1 - 12\gamma_2 + 58\gamma_1\gamma_2 + 13\gamma_1^2 + 29\gamma_2^2 - 114\gamma_1^2\gamma_2 - 114\gamma_1\gamma_2^2 + 116\gamma_1^3\gamma_2 + 174\gamma_1^2\gamma_2^2 \\
+ 68\gamma_1\gamma_2^3 - 64\gamma_1^4\gamma_2 - 128\gamma_1^3\gamma_2^2 - 76\gamma_1^2\gamma_2^3 - 12\gamma_1\gamma_2^4 + 45\gamma_1^4\gamma_2^2 + 36\gamma_1^3\gamma_2^3 + 9\gamma_1^2\gamma_2^4 + 18\gamma_2\gamma_1^5 \\
- 2\gamma_1^6\gamma_2 - 6\gamma_1^5\gamma_2^2 - 6\gamma_1^4\gamma_2^3 - 2\gamma_1^3\gamma_2^4 - 6\gamma_1^3 + \gamma_1^4 - 22\gamma_2^3 + 5\gamma_2^4) + (288\gamma_1\gamma_2 + 144\gamma_1^2 + 144\gamma_2^2 \\
- 936\gamma_1^2\gamma_2 - 1320\gamma_1\gamma_2^2 + 548\gamma_1^3\gamma_2 + 2678\gamma_1^2\gamma_2^2 + 3556\gamma_1\gamma_2^3 + 1660\gamma_1^4\gamma_2 - 2440\gamma_1^3\gamma_2^2 \\
- 8296\gamma_1^2\gamma_2^3 - 5924\gamma_1\gamma_2^4 + 4994\gamma_1^4\gamma_2^2 + 20088\gamma_1^3\gamma_2^3 + 16778\gamma_1^2\gamma_2^4 - 3468\gamma_1^5\gamma_2 + 3004\gamma_1^6\gamma_2 \\
- 13268\gamma_1^5\gamma_2^2 - 44532\gamma_1^4\gamma_2^3 - 42548\gamma_1^3\gamma_2^4 + 18994\gamma_1^6\gamma_2^2 + 64168\gamma_1^5\gamma_2^3 + 73538\gamma_1^4\gamma_2^4 \\
- 1436\gamma_1^7\gamma_2 - 16812\gamma_1^2\gamma_2^5 + 37548\gamma_1^3\gamma_2^5 + 5604\gamma_1\gamma_2^5 - 2860\gamma_1\gamma_2^6 + 8466\gamma_1^2\gamma_2^6 - 312\gamma_1^3 \\
+ 313\gamma_1^4 - 696\gamma_2^3 - 180\gamma_1^5 + 1369\gamma_2^4 - 1396\gamma_2^5 + 774\gamma_2^6 + 62\gamma_1^6 - 12\gamma_1^7 + \gamma_1^8 + 396\gamma_1^8\gamma_2 \\
- 57556\gamma_1^6\gamma_2^3 - 15704\gamma_1^7\gamma_2^2 - 78488\gamma_1^5\gamma_2^4 - 50940\gamma_1^4\gamma_2^5 - 15912\gamma_1^3\gamma_2^6 - 60\gamma_1^9\gamma_2 \\
+ 33040\gamma_1^7\gamma_2^3 + 8134\gamma_1^8\gamma_2^2 + 52498\gamma_1^6\gamma_2^4 + 41476\gamma_1^5\gamma_2^5 + 728\gamma_1\gamma_2^7 - 2076\gamma_1^2\gamma_2^7 + 16786\gamma_1^4\gamma_2^6 \\
+ 3160\gamma_1^3\gamma_2^7 - 22340\gamma_1^7\gamma_2^4 - 10260\gamma_1^5\gamma_2^6 - 20748\gamma_1^6\gamma_2^5 - 2508\gamma_1^4\gamma_2^7 - 12300\gamma_1^8\gamma_2^3 - 2724\gamma_1^9\gamma_2^2 \\
- 220\gamma_2^7 + 25\gamma_2^8 + 1096\gamma_1^5\gamma_2^7 + 198\gamma_1^2\gamma_2^8 - 228\gamma_1^3\gamma_2^8 + 129\gamma_1^4\gamma_2^8 - 1080\gamma_1^8\gamma_2^5 - 252\gamma_1^6\gamma_2^7 \\
- 720\gamma_1^7\gamma_2^6 - 36\gamma_1^5\gamma_2^8 - 900\gamma_1^9\gamma_2^4 - 396\gamma_1^{10}\gamma_2^3 - 72\gamma_1\gamma_2^8 + 4\gamma_1^6\gamma_2^8 + 4\gamma_1^{12}\gamma_2^2 + 60\gamma_1^{10}\gamma_2^4 \\
+ 24\gamma_1^{11}\gamma_2^3 + 80\gamma_1^9\gamma_2^5 - 72\gamma_1^{11}\gamma_2^2 + 60\gamma_1^8\gamma_2^6 + 24\gamma_1^7\gamma_2^7 + 2900\gamma_1^9\gamma_2^3 + 580\gamma_1^{10}\gamma_2^2 + 5929\gamma_1^8\gamma_2^4 \\
+ 4\gamma_1^{10}\gamma_2 + 6316\gamma_1^7\gamma_2^5 + 3674\gamma_1^6\gamma_2^6)^{\frac{1}{2}}
\end{array}\right).}{2\gamma_1\gamma_2\left(\begin{array}{l}
76\gamma_1\gamma_2^2 - 45\gamma_1^3\gamma_2 + 64\gamma_1^3 - 9\gamma_1\gamma_2^3 + 2\gamma_1^5 + 6\gamma_1^3\gamma_2^2 + 128\gamma_1^2\gamma_2 - 96\gamma_1^2 \\
- 144\gamma_1\gamma_2 + 6\gamma_1^4\gamma_2 - 48\gamma_2^2 + 12\gamma_2^3 - 36\gamma_1^2\gamma_2^2 + 2\gamma_1^2\gamma_2^3 + 48\gamma_1 + 48\gamma_2 - 18\gamma_1^4
\end{array}\right)}.
\end{aligned}
$$

From the obtained roots, it is observed that roots t_3 and t_4 are associated with the parameters γ_1 and γ_2. Therefore, an analysis of the relationship between these parameters and the roots is deemed essential to satisfy the criteria for zero-stability as outlined in Definition 3.3. To facilitate implementation, one of the parameters can be fixed. In this study, a specific value for γ_1 will be chosen. By setting γ_1 to a constant value, the corresponding range of γ_2 values that ensure zero-stability for both roots t_3 and t_4 can be identified. Fixed values of γ_1 at 25, 50, and 100 will be considered. This approach allows for the examination of how the range of γ_2 values behaves graphically. Specifically, values of γ_2 within the range of $[-200, 200]$ will be analyzed to observe their impact on the stability of the roots.

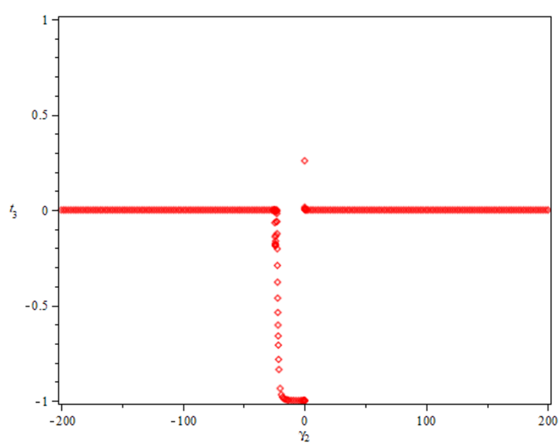

Figure 2: Graph of t_3 versus γ_2 for CBBDF(4) when $\gamma_1 = 25$.

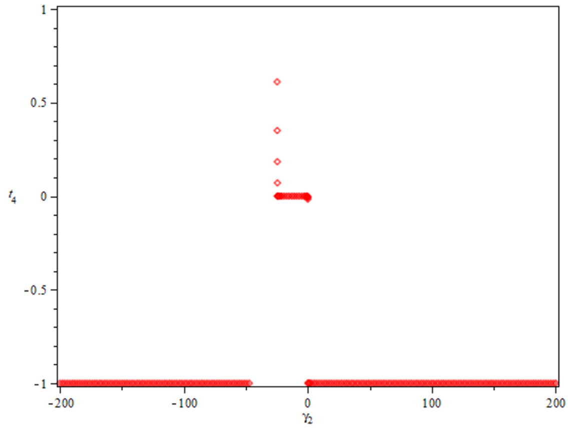

Figure 3: Graph of t_4 versus γ_2 for CBBDF(4) when $\gamma_1 = 25$.

Figures 2–3 illustrate that for $\gamma_1 = 25$, the requirement $|t_3| \leq 1$ is fulfilled for all values of γ_2 within the range $(-\infty, \infty)$. Conversely, the condition $|t_4| \leq 1$ is met when γ_2 is confined to the intervals $(-\infty, -47.015) \cup (-24.862, \infty)$. By combining these criteria, the values of γ_2 that guarantee zero-stability for both roots t_3 and t_4 are expressed as $\gamma_2 \in (-\infty, -47.015) \cup (-24.862, \infty)$.

Similarly, Figures 4–5 reveal that when $\gamma_1 = 50$, the condition $|t_3| \leq 1$ remains valid for all values of γ_2 across the entire range of $(-\infty, \infty)$. In this case, the condition for zero-stability of root t_4 is satisfied when γ_2 falls within the intervals $(-\infty, -85.083) \cup (-49.916, \infty)$. Thus, the corresponding values $\gamma_2 \in (-\infty, -85.083) \cup (-49.916, \infty)$ guarantee stability for both roots.

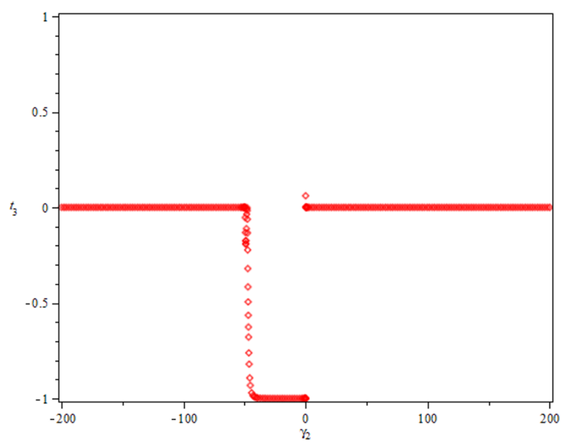

Figure 4: Graph of t_3 versus γ_2 for CBBDF(4) when $\gamma_1 = 50$.

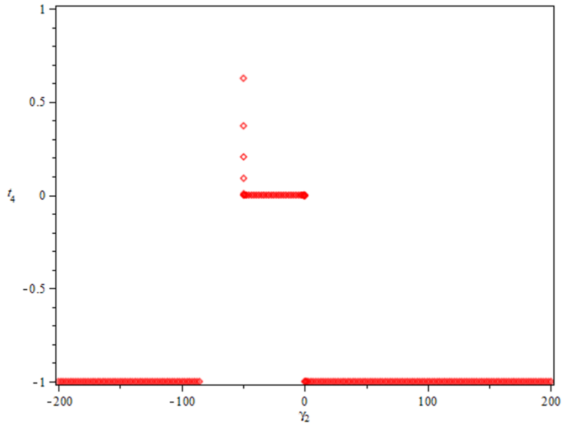

Figure 5: Graph of t_4 versus γ_2 for CBBDF(4) when $\gamma_1 = 50$.

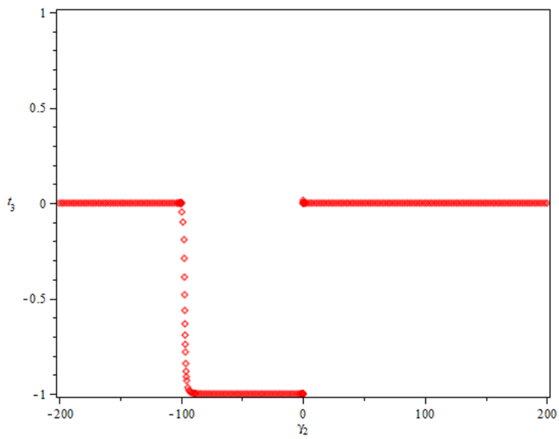

Figure 6: Graph of t_3 versus γ_2 for CBBDF(4) when $\gamma_1 = 100$.

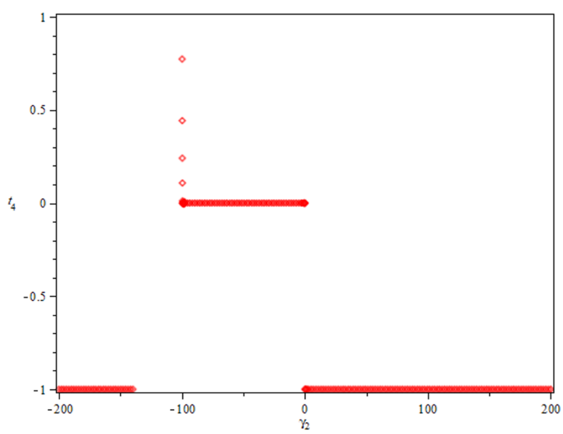

Figure 7: Graph of t_4 versus γ_2 for CBBDF(4) when $\gamma_1 = 100$.

Finally, Figures 6−7 depict that for a value of $\gamma_1 = 100$, the condition $|t_3| \leq 1$ continues to hold for all values of γ_2 in the range of $(-\infty, \infty)$. Meanwhile, the stability condition for root t_4 is satisfied when γ_2 lies within the intervals of $(-\infty, -141.216) \cup (-99.865, \infty)$. Therefore, after merging these conditions, the values of γ_2 that ensure zero-stability for both roots are expressed as $\gamma_2 \in (-\infty, -141.216) \cup (-99.865, \infty)$.

Theorem 3.1. *The conditions required for the LMM to achieve convergence are that it must be both consistent and zero-stable.*

Proof. Since both the conditions of consistency and zero-stability have been satisfied in the previous analysis, it is concluded that the CBBDF(4) method converges and is appropriate for the numerical integration of first-order stiff ODEs. □

3.3 Absolute stability region

The stability of implicit numerical methods for stiff IVPs is crucial for avoiding numerical instabilities and ensuring accurate, reliable solutions. One of the key criteria for evaluating the robustness of these methods is $A-$stability, which facilitates efficient simulations of problems involving disparate timescales. The definition for $A-$stability is given as follows;

Definition 3.4. *The LMM is said to be $A-$stable (also known as stiffly stable) if the stable region covers the entire negative complex plane.*

According to the literature, the boundary locus technique is regarded as the most widely used and convenient method for identifying the region of absolute stability. The boundary can be established by substituting the set of points $t = e^{i\theta}$, where $|t| \leq 1$ and θ ranges from 0 to 2π. Consequently, the stability region graphs are plotted using Maple software. Based on the previous analysis of zero-stability, a broad range of intervals for γ_1 and γ_2 are identified that satisfy the properties of zero-stability. Consequently, specific values of γ_1 and γ_2 will be emphasized in this study to illustrate the $A-$stability region. The chosen values for γ_1 and γ_2 are 25, 50 and 100, with both parameters set equal. These values will be utilized to analyze their $A-$stability behavior. It should be noted that the stability regions for other values of γ_1 and γ_2 are not presented in this study, as no significant differences are observed.

Figure 8 illustrates the stability regions for the BBDF and CBBDF(4) methods with parameters $\gamma_1 = \gamma_2 = 25$, $\gamma_1 = \gamma_2 = 50$ and $\gamma_1 = \gamma_2 = 100$. The graphs indicate that the stable region lies outside the boundary defined by the diamond-shaped line, while the unstable region is contained within it. Notably, the CBBDF(4) method demonstrates a smaller unstable region compared to the BBDF method, indicating enhanced stability characteristics. According to Definition 3.4, the CBBDF(4) method is classified as $A-$stable, as its stability region encompasses the entire negative half-plane.

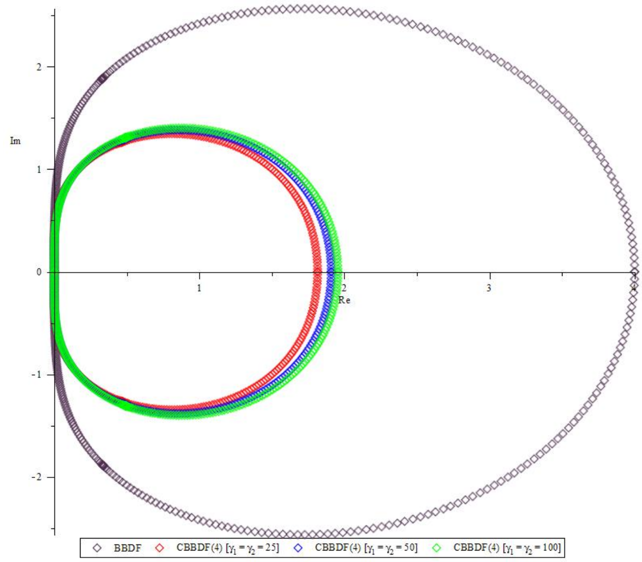

Figure 8: The graphs of stability regions for BBDF and CBBDF(4).

4 Implementation of the Method

The CBBDF(4) method in the third stage is implemented within a predictor-corrector computational framework. Before advancing to this stage, it is necessary to compute the intermediate values, $y_{n+\gamma_1}$ and $y_{n+\gamma_1+\gamma_2}$, which serve as crucial inputs for determining the future solutions y_{n+1} and y_{n+2}. To obtain these intermediate values, Euler's method is employed during the first and second stages. This approach provides a solid foundation for the subsequent calculations. Once the intermediate values are established, they are used to predict the future solutions, as outlined in (15). The predicted values can then be refined using the corrector formula associated with the CBBDF(4) method in (12) and (13). This refinement process is conducted through the application of Newton's method, which is particularly effective for integrating stiff ODEs. The general formulation of Newton's iteration for the CBBDF(4) is written as,

$$y^{(i+1)}_{n+1,n+2} = y^{(i)}_{n+1,n+2} - \left[\frac{\hat{F}_{1,2}\left(y^{(i)}_{n+1,n+2}\right)}{\hat{F}'_{1,2}\left(y^{(i)}_{n+1,n+2}\right)} \right], \tag{25}$$

where the notation (i) and $(i+1)$ represent the previous and current iterations, respectively and

$$\begin{aligned}
\hat{F}_1 = y_{n+1} &- \left(\frac{\left(1 - 2\gamma_1 + \gamma_1^2 - \gamma_2 + \gamma_1\gamma_2\right)(\gamma_1 - 1)(\gamma_1 + \gamma_2 - 1)}{2(\gamma_1 - 2)(\gamma_1 + \gamma_2 - 2)(2\gamma_1 + \gamma_2 - 2)} \right) y_{n+2} \\
&- \left(\frac{(\gamma_1 - 1)(\gamma_1 + \gamma_2 - 1)}{2 - 2\gamma_1 - \gamma_2} \right) hf_{n+1} - \xi_1, \\
\hat{F}_2 = y_{n+2} &- \left(\frac{4\left(4 - 4\gamma_1 - 2\gamma_2 + \gamma_1^2 + \gamma_1\gamma_2\right)(\gamma_1 - 2)(\gamma_1 + \gamma_2 - 2)}{(\gamma_1 - 1)(\gamma_1 + \gamma_2 - 1)\left(20 - 16\gamma_1 + 3\gamma_1^2 - 8\gamma_2 + 3\gamma_1\gamma_2\right)} \right) y_{n+1} \\
&- \left(\frac{2(\gamma_1 - 2)(\gamma_1 + \gamma_2 - 2)}{20 - 16\gamma_1 + 3\gamma_1^2 - 8\gamma_2 + 3\gamma_1\gamma_2} \right) hf_{n+2} - \xi_2,
\end{aligned}$$

with ξ_1 and ξ_2 are the intermediate values in the previous block. To clarify the notation, we define,

$$e^{(i+1)}_{n+1,n+2} = y^{(i+1)}_{n+1,n+2} - y^{(i)}_{n+1,n+2}, \tag{26}$$

where $e^{(i+1)}_{n+1,n+2}$ represents the difference between the values of $y_{n+1,n+2}$ at the $(i)^{th}$ and $(i+1)^{th}$ iterations. Substituting (26) into (25) allows Newton's iteration formula to be rearranged and expressed as,

$$\hat{F}'_{1,2}\left(y^{(i)}_{n+1,n+2}\right) \cdot \left[e^{(i+1)}_{n+1,n+2}\right] = -\hat{F}_{1,2}\left(y^{(i)}_{n+1,n+2}\right). \tag{27}$$

The matrix form of (27) becomes,

$$\begin{aligned}
&\begin{bmatrix} 1-\left(\frac{(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2-2\gamma_1-\gamma_2}\right)h\frac{\partial f_{n+1}}{\partial y_{n+1}} & -\left(\frac{\left(1-2\gamma_1+\gamma_1^2-\gamma_2+\gamma_1\gamma_2\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2(\gamma_1-2)(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right) \\ -\left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) & 1-\left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right)h\frac{\partial f_{n+2}}{\partial y_{n+2}} \end{bmatrix}\begin{bmatrix} e^{(i+1)}_{n+1} \\ e^{(i+1)}_{n+2} \end{bmatrix} \\
&= \begin{bmatrix} -1 & \left(\frac{\left(1-2\gamma_1+\gamma_1^2-\gamma_2+\gamma_1\gamma_2\right)(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2(\gamma_1-2)(\gamma_1+\gamma_2-2)(2\gamma_1+\gamma_2-2)}\right) \\ \left(\frac{4\left(4-4\gamma_1-2\gamma_2+\gamma_1^2+\gamma_1\gamma_2\right)(\gamma_1-2)(\gamma_1+\gamma_2-2)}{(\gamma_1-1)(\gamma_1+\gamma_2-1)\left(20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2\right)}\right) & -1 \end{bmatrix}\begin{bmatrix} y^{(i)}_{n+1} \\ y^{(i)}_{n+2} \end{bmatrix} \\
&\quad + h\begin{bmatrix} \left(\frac{(\gamma_1-1)(\gamma_1+\gamma_2-1)}{2-2\gamma_1-\gamma_2}\right) & 0 \\ 0 & \left(\frac{2(\gamma_1-2)(\gamma_1+\gamma_2-2)}{20-16\gamma_1+3\gamma_1^2-8\gamma_2+3\gamma_1\gamma_2}\right) \end{bmatrix}\begin{bmatrix} f^{(i)}_{n+1} \\ f^{(i)}_{n+2} \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}.
\end{aligned} \tag{28}$$

The algorithm developed is conducting using $PECE$ (predict-evaluate-correct-evaluate) mode. 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.
$E: y'_{n+1,n+2} = f\left(x_{n+1,n+2}, y^{(c)}_{n+1,n+2}\right)$.

5 Numerical Results and Discussions

In this study, the newly developed method, CBBDF(4), will be employed to approximate the solutions for both single equations and systems of linear and non-linear stiff IVPs using step sizes of 10^{-1}, 10^{-3} and 10^{-5}. The computed numerical results will be compared with those obtained from the existing 2-point BBDF method to evaluate the performance of the new composite block scheme in terms of maximum error, average error and computational time. The 2-point BBDF method is chosen because it is a pioneer and well-known method in the family of multistep block methods, making it an ideal benchmark for comparing the performance of the new composite block scheme. The selected values of the independent parameters, γ_1 and γ_2, are 25, 50 and 100, which have been shown to have $A-$stability characteristics. Tables 1−4 present the numerical approximations for each test problem. To visually illustrate the results, graphs of $\log(h)$ against $\log(\text{MAXE})$ and $\log(\text{TIME})$ against $\log(\text{MAXE})$ are shown in Figures 9−16. The tables use the following notations:

h	:	Step size.
γ_1	:	Independent parameter in the first stage of CBBDF(4).
γ_2	:	Independent parameter in the second stage of CBBDF(4).
BBDF	:	2-point block backward differentiation formula by [7].
CBBDF(4)	:	Three-sub-steps composite block backward differentiation formula of order four.
MAXE	:	Maximum error.
AVE	:	Average error.
TIME	:	Computational time (in seconds).

Example 5.1. *The cosine problem is formulated as follows,*

$$y'(x) = -2\pi \sin(2\pi x) - \frac{1}{\epsilon}(y - \cos(2\pi x)),$$

where $\epsilon = 10^{-3}$. The initial condition is given by $y(0) = 1$, and the problem is defined over the range $x \in [0, 10]$. The exact solution to this equation is $y(x) = \cos(2\pi x)$. The eigenvalue of the Jacobian matrix associated with this equation is $\lambda = -1000$. As ϵ approaches zero, this cosine problem becomes increasingly stiff, thus classifying it as a highly stiff ODEs.
Source: [12]

Example 5.2. *This test problem involves a non-linear mildly stiff ODEs with oscillating solutions over the range $x \in [0, 3]$, defined as follows,*

$$y'(x) = 100(\sin(x) - y).$$

The initial condition is specified as $y(0) = 0$. The eigenvalue associated with this equation is $\lambda = -100$. The theoretical solution to this differential equation is expressed as,

$$y(x) = \frac{\sin(x) - 0.01\cos(x) + 0.01e^{-100x}}{1.0001}.$$

Source: [14]

Example 5.3. *Consider a system of linear equations which is classified as mildly stiff ODEs defined over the interval $x \in [0, 10]$. The equations are expressed as follows,*

$$\begin{aligned} {y_1}'(x) &= 198y_1 + 199y_2, \\ {y_2}'(x) &= -398y_1 - 399y_2. \end{aligned}$$

The exact solutions to this system are given by,

$$\begin{aligned} y_1(x) &= e^{-x}, \\ y_2(x) &= -e^{-x}. \end{aligned}$$

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 $\lambda_2 = -200$.
Source: [26]

Example 5.4. *This test problem also consists of a system of linear mildly stiff ODEs defined over the interval* $x \in [0, 10]$. *The equations are as follows,*

$$y_1{}'(x) = -100y_1 + 9.901y_2,$$
$$y_2{}'(x) = 0.1y_1 - y_2.$$

The theoretical solutions for this system are given by,

$$y_1(x) = e^{-0.99x},$$
$$y_2(x) = 10e^{-0.99x}.$$

The initial conditions for this problem are $y_1(0) = 1$ *and* $y_2(0) = 10$. *The eigenvalues associated with this system are* $\lambda_1 = -0.99$ *and* $\lambda_2 = -100.01$.
Source: [15]

Table 1: Numerical results for Example 5.1.

h	Method	γ_1	γ_2	MAXE	AVE	TIME
10^{-1}	BBDF	-	-	3.80769e+014	1.46501e+013	0.001959
	CBBDF(4)	25	25	9.86614e-003	3.20069e-003	0.001161
		25	50	2.54025e-003	1.27628e-003	0.000929
		25	100	1.11552e-003	5.58448e-004	0.000396
		50	25	6.07163e+000	6.97518e-001	0.000348
		50	50	3.83952e-003	1.89955e-003	0.000411
		50	100	1.64141e-003	8.07837e-004	0.000374
		100	25	1.08516e-002	3.74793e-003	0.000411
		100	50	4.63403e-003	2.11430e-003	0.000338
		100	100	2.28738e-003	1.09418e-003	0.000367
10^{-3}	BBDF	-	-	1.01029e+064	4.05835e+061	0.002541
	CBBDF(4)	25	25	4.32939e-006	1.40646e-006	0.018117
		25	50	2.09464e-006	6.76399e-007	0.011730
		25	100	1.30234e-006	4.17693e-007	0.011886
		50	25	7.13327e-006	2.29832e-006	0.011706
		50	50	3.31705e-006	1.06810e-006	0.015586
		50	100	1.35533e-006	4.35226e-007	0.011673
		100	25	1.09042e-005	3.49370e-006	0.011741
		100	50	6.40748e-006	2.05214e-006	0.012977
		100	100	2.85186e-006	9.13394e-007	0.011668
10^{-5}	BBDF	-	-	7.89552e-007	5.02563e-007	0.282675
	CBBDF(4)	25	25	1.72878e-008	1.09229e-008	1.172617
		25	50	1.36535e-008	8.62639e-009	1.165852
		25	100	1.16733e-008	7.37511e-009	1.170962
		50	25	1.05536e-008	6.66786e-009	1.164631
		50	50	7.91294e-009	4.99938e-009	1.179072
		50	100	6.20632e-009	3.92109e-009	1.153965
		100	25	6.94529e-009	4.38798e-009	1.168986
		100	50	5.30804e-009	3.35357e-009	1.169792
		100	100	3.88745e-009	2.45603e-009	1.170312

From Table 1, it can be observed that for a step size of $h = 10^{-1}$, the maximum error is minimized when using parameters $\gamma_1 = 25$ and $\gamma_2 = 100$, while the larger error occurs with $\gamma_1 = 50$ and $\gamma_2 = 25$. When the step size is reduced to $h = 10^{-3}$, CBBDF(4) demonstrates improved accuracy with parameters $\gamma_1 = 25$ and $\gamma_2 = 100$; however, the less accurate results are observed with both parameters set to $\gamma_1 = 100$ and $\gamma_2 = 25$. For an even smaller step size of $h = 10^{-5}$, the proposed method yields stronger results with parameters $\gamma_1 = 100$ and $\gamma_2 = 100$, whereas the weaker performance is noted with $\gamma_1 = 25$ and $\gamma_2 = 25$.

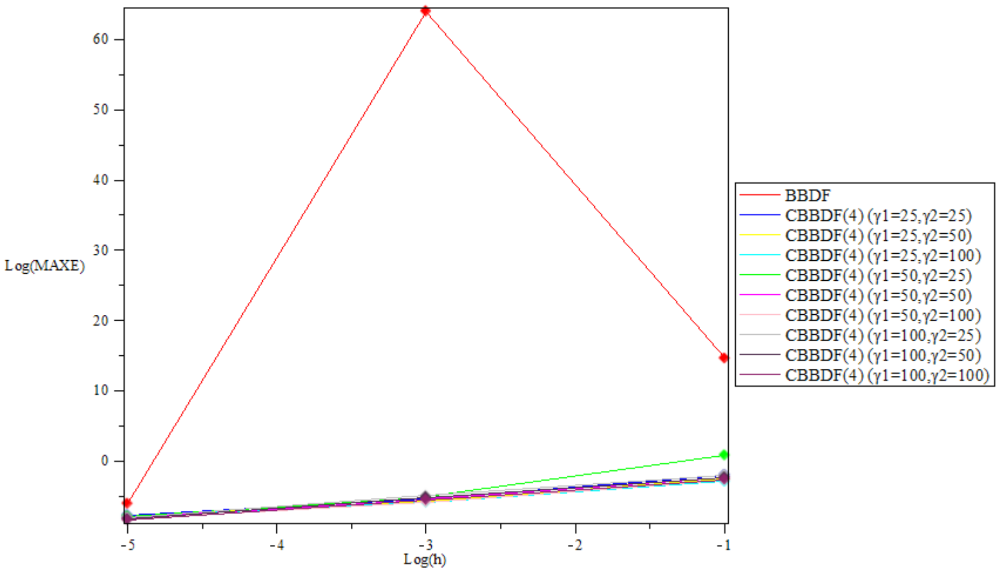

Figure 9: Accuracy graph for Example 5.1.

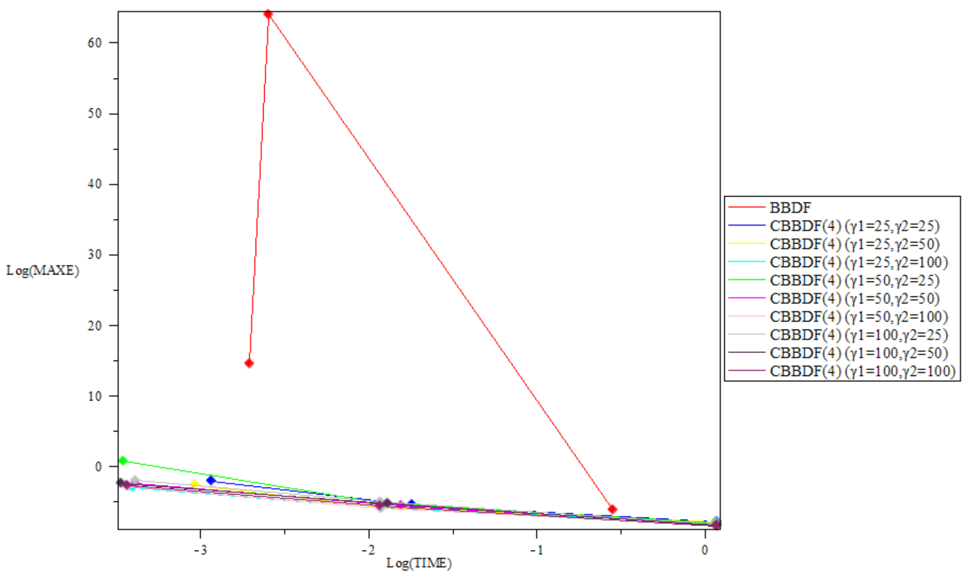

Figure 10: Efficiency graph for Example 5.1.

Table 2: Numerical results for Example 5.2.

h	Method	γ_1	γ_2	MAXE	AVE	TIME
10^{-1}	BBDF	-	-	2.20355e+003	3.14415e+002	0.000075
	CBBDF(4)	25	25	9.07150e-004	4.48382e-004	0.000834
		25	50	4.46571e-004	1.59219e-004	0.000817
		25	100	4.82882e-004	7.36945e-005	0.000616
		50	25	1.17552e-003	5.45056e-004	0.000472
		50	50	5.66790e-004	2.41237e-004	0.000321
		50	100	5.32487e-004	1.05782e-004	0.000364
		100	25	9.08550e-004	4.63750e-004	0.000538
		100	50	5.43023e-004	2.77501e-004	0.000327
		100	100	5.33897e-004	1.38062e-004	0.000341
10^{-3}	BBDF	-	-	5.67110e-004	1.78110e-005	0.015676
	CBBDF(4)	25	25	9.43078e-006	3.84706e-007	0.006373
		25	50	6.12461e-006	2.73561e-007	0.010773
		25	100	4.33133e-006	2.19250e-007	0.019829
		50	25	3.07588e-006	2.76994e-007	0.005132
		50	50	8.34234e-007	1.66250e-007	0.014490
		50	100	6.33008e-007	1.12652e-007	0.012525
		100	25	6.06589e-007	2.61113e-007	0.008763
		100	50	1.79507e-006	1.82796e-007	0.012728
		100	100	2.84885e-006	1.20044e-007	0.012460
10^{-5}	BBDF	-	-	7.33991e-006	1.98408e-007	0.261936
	CBBDF(4)	25	25	1.57952e-007	4.27728e-009	0.574851
		25	50	1.25883e-007	3.40910e-009	0.484129
		25	100	1.08002e-007	2.92564e-009	0.485253
		50	25	9.19015e-008	2.50090e-009	0.536899
		50	50	7.06918e-008	1.92285e-009	0.669888
		50	100	5.65001e-008	1.53690e-009	0.584927
		100	25	5.43060e-008	1.49355e-009	0.602972
		100	50	4.32073e-008	1.18684e-009	0.663891
		100	100	3.32433e-008	9.12181e-010	0.594253

From Table 2, when the step size is set to $h = 10^{-1}$, the parameters that result in better minimization of the maximum error are identified as $\gamma_1 = 25$ and $\gamma_2 = 50$. Conversely, the more significant maximum error arises with the parameters $\gamma_1 = 50$ and $\gamma_2 = 25$. As the step size is refined to $h = 10^{-3}$, the configuration yielding a more favorable maximum error shifts to $\gamma_1 = 100$ and $\gamma_2 = 25$, while the less favorable outcome is observed with both parameters set to $\gamma_1 = 25$ and $\gamma_2 = 25$. Further reducing the step size to $h = 10^{-5}$ reveals that improved performance is achieved with $\gamma_1 = 100$ and $\gamma_2 = 100$, whereas the less effective results occur with both parameters at $\gamma_1 = 25$ and $\gamma_2 = 25$.

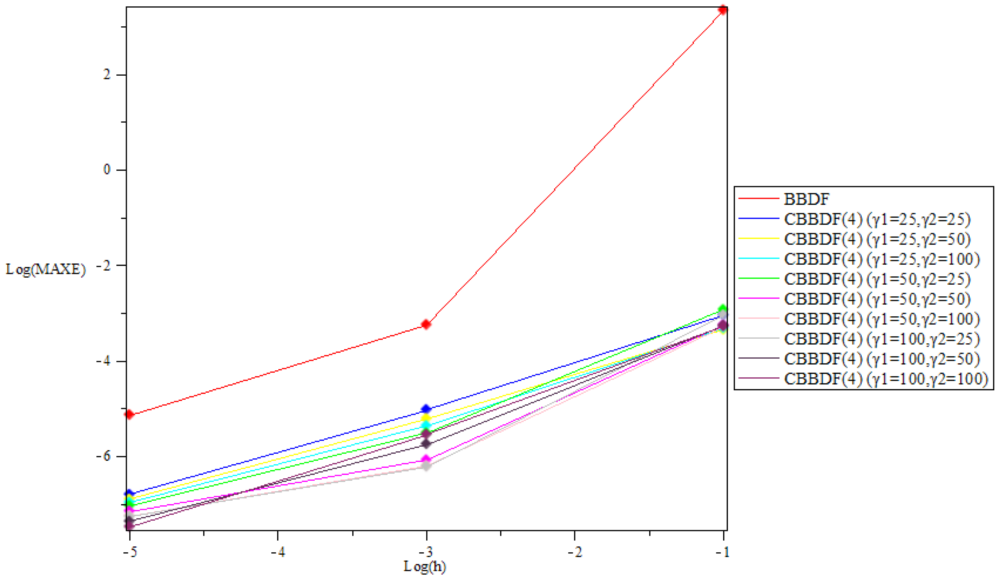

Figure 11: Accuracy graph for Example 5.2.

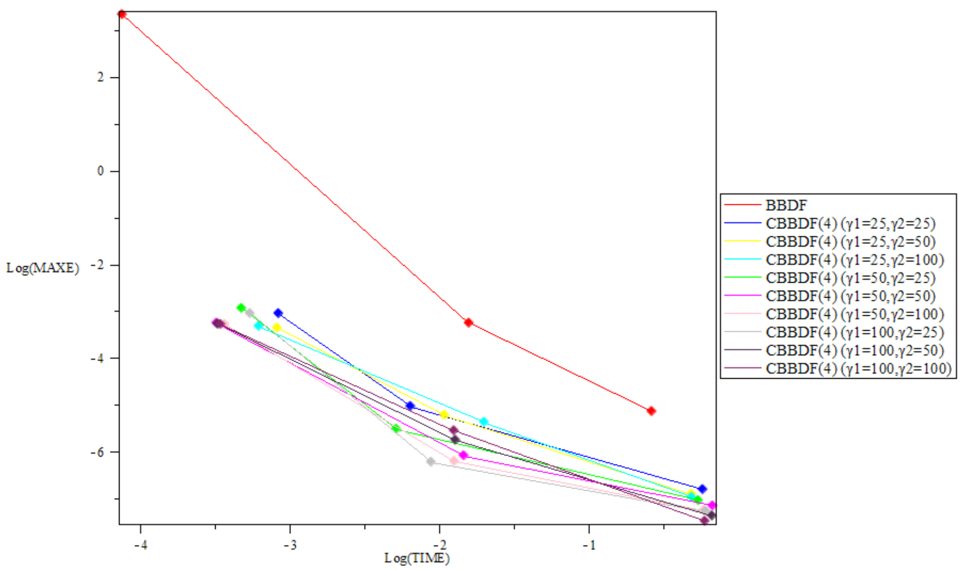

Figure 12: Efficiency graph for Example 5.2.

Table 3: Numerical results for Example 5.3.

h	Method	γ_1	γ_2	MAXE	AVE	TIME
10^{-1}	BBDF	-	-	5.67155e-002	2.76304e-002	0.000050
	CBBDF(4)	25	25	9.43237e-004	4.71699e-004	0.000819
		25	50	6.12568e-004	3.06182e-004	0.000790
		25	100	4.33220e-004	2.16359e-004	0.000536
		50	25	3.07887e-004	1.53812e-004	0.000361
		50	50	8.35866e-005	4.15285e-005	0.000735
		50	100	6.32076e-005	3.19863e-005	0.000901
		100	25	5.99303e-005	3.03339e-005	0.000465
		100	50	1.79064e-004	8.99767e-005	0.001121
		100	100	2.84666e-004	1.42859e-004	0.000627
10^{-3}	BBDF	-	-	7.34012e-004	3.98453e-004	0.008055
	CBBDF(4)	25	25	1.57957e-005	8.57659e-006	0.016035
		25	50	1.25887e-005	6.83530e-006	0.014053
		25	100	1.08005e-005	5.86435e-006	0.037922
		50	25	9.19042e-006	4.99015e-006	0.013366
		50	50	7.06939e-006	3.83850e-006	0.022335
		50	100	5.65018e-006	3.06790e-006	0.039168
		100	25	5.43077e-006	2.94877e-006	0.024846
		100	50	4.32086e-006	2.34612e-006	0.040475
		100	100	3.32443e-006	1.80509e-006	0.035614
10^{-5}	BBDF	-	-	7.35741e-006	3.99787e-006	0.459211
	CBBDF(4)	25	25	1.58607e-007	8.61837e-008	1.856961
		25	50	1.26542e-007	6.87598e-008	1.517058
		25	100	1.08656e-007	5.90415e-008	1.770600
		50	25	9.25212e-008	5.02737e-008	1.948932
		50	50	7.13211e-008	3.87537e-008	1.701128
		50	100	5.71325e-008	3.10447e-008	1.554695
		100	25	5.49136e-008	2.98388e-008	1.748470
		100	50	4.38209e-008	2.38114e-008	2.089290
		100	100	3.38616e-008	1.84001e-008	1.688755

In Table 3, for a step size of $h = 10^{-1}$, the parameters that yield a reduced maximum error are found to be $\gamma_1 = 100$ and $\gamma_2 = 25$. The larger maximum error is recorded with both parameters at their lower values of $\gamma_1 = 25$ and $\gamma_2 = 25$. When analyzing a smaller step size of $h = 10^{-3}$, the more accurate results emerge with both parameters set to $\gamma_1 = 100$ and $\gamma_2 = 100$. Again, the less accurate configuration is seen with both parameters at their lower values of $\gamma_1 = 25$ and $\gamma_2 = 25$. At an even finer resolution of $h = 10^{-5}$, leading performance is maintained at the same parameter values of $\gamma_1 = 100$ and $\gamma_2 = 100$, contrasting with the weaker performance noted at lower parameter values.

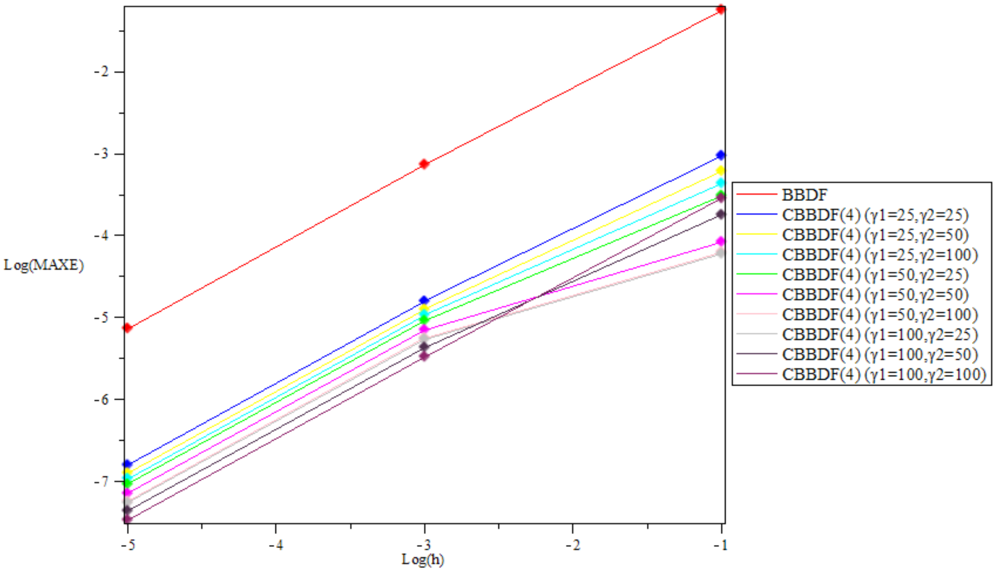

Figure 13: Accuracy graph for Example 5.3.

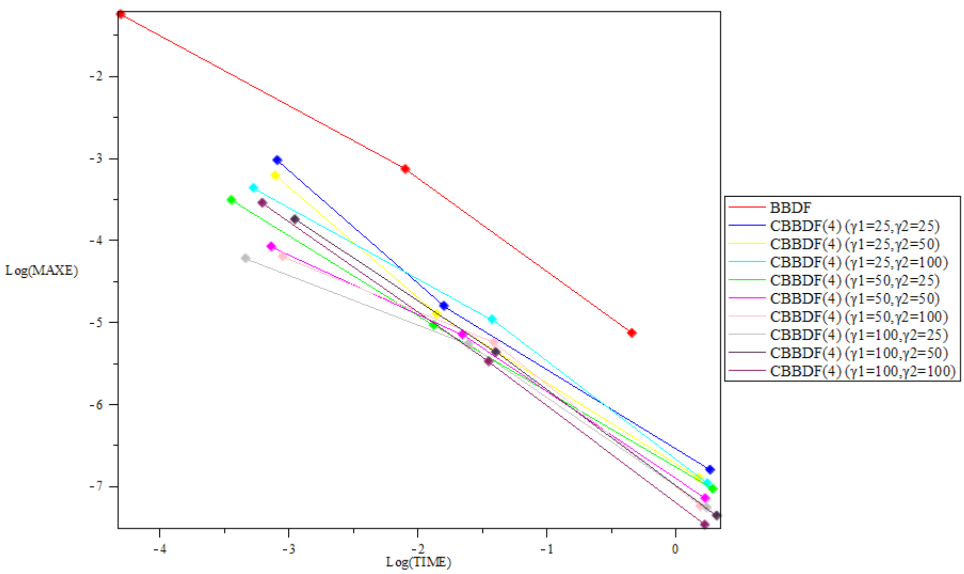

Figure 14: Efficiency graph for Example 5.3.

Table 4: Numerical results for Example 5.4.

h	Method	γ_1	γ_2	MAXE	AVE	TIME
10^{-1}	BBDF	-	-	5.63179e-001	1.52583e-001	0.000029
	CBBDF(4)	25	25	9.40590e-003	2.61411e-003	0.000285
		25	50	6.13197e-003	1.70338e-003	0.000392
		25	100	4.35578e-003	1.20900e-003	0.000381
		50	25	3.11076e-003	8.63712e-004	0.000293
		50	50	8.90524e-004	2.46040e-004	0.000399
		50	100	5.62826e-004	1.58451e-004	0.000376
		100	25	5.32950e-004	1.50074e-004	0.000431
		100	50	1.71202e-003	4.78121e-004	0.000418
		100	100	2.75730e-003	7.69018e-004	0.000474
10^{-3}	BBDF	-	-	7.26689e-003	2.19146e-003	0.002554
	CBBDF(4)	25	25	1.56384e-004	4.71713e-005	0.009169
		25	50	1.24635e-004	3.75946e-005	0.010456
		25	100	1.06931e-004	3.22546e-005	0.009204
		50	25	9.09913e-005	2.74466e-005	0.009283
		50	50	6.99932e-005	2.11128e-005	0.009246
		50	100	5.59431e-005	1.68747e-005	0.009089
		100	25	5.37707e-005	1.62194e-005	0.009191
		100	50	4.27827e-005	1.29050e-005	0.009114
		100	100	3.29180e-005	9.92943e-006	0.009223
10^{-5}	BBDF	-	-	7.28384e-005	2.19872e-005	0.245668
	CBBDF(4)	25	25	1.57021e-006	4.73988e-007	0.911100
		25	50	1.25276e-006	3.78161e-007	0.908632
		25	100	1.07570e-006	3.24713e-007	0.924102
		50	25	9.15961e-007	2.76492e-007	0.950025
		50	50	7.06079e-007	2.13135e-007	0.935472
		50	100	5.65613e-007	1.70738e-007	0.926682
		100	25	5.43645e-007	1.64106e-007	0.925899
		100	50	4.33828e-007	1.30957e-007	0.967707
		100	100	3.35231e-007	1.01196e-007	0.920733

Turning to Table 4, consistent results are observed. With a step size of $h = 10^{-1}$, the configuration that minimizes maximum error is again identified as $\gamma_1 = 100$ and $\gamma_2 = 25$. The larger maximum error is noted to persist at the same parameter settings of both values at their lower levels: $\gamma_1 = 25$ and $\gamma_2 = 25$. As the step size is decreased to $h = 10^{-3}$, improved accuracy is achieved with parameters set to both values at $\gamma_1 = 100$ and $\gamma_2 = 100$. Once more, the less favorable results are observed with both parameters at their lower values of $\gamma_1 = 25$ and $\gamma_2 = 25$. This trend continues at a step size of $h = 10^{-5}$, where better performance is noted to remain unchanged at parameters of $\gamma_1 = 100$ and $\gamma_2 = 100$, contrasting with the less favorable performance observed at lower parameter values.

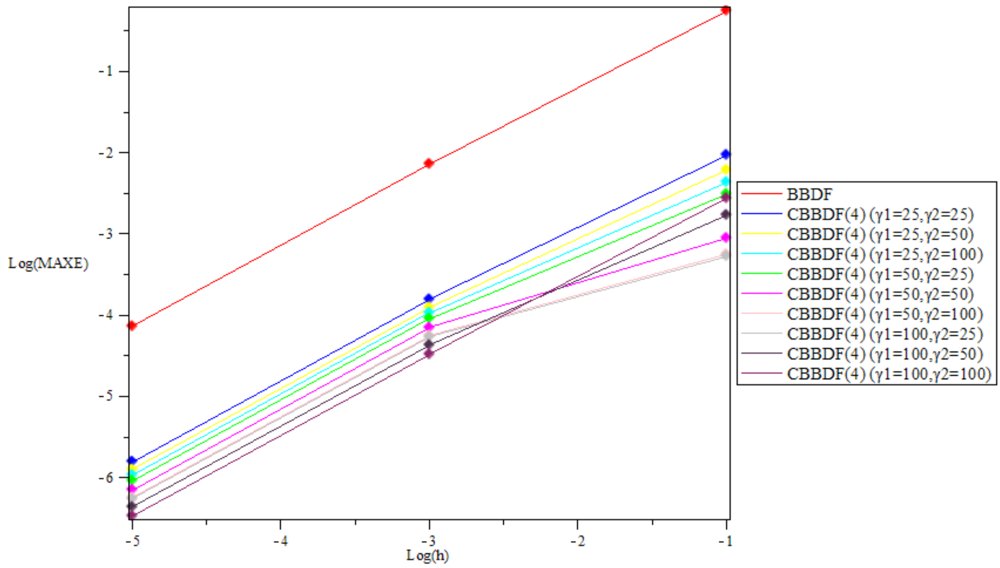

Figure 15: Accuracy graph for Example 5.4.

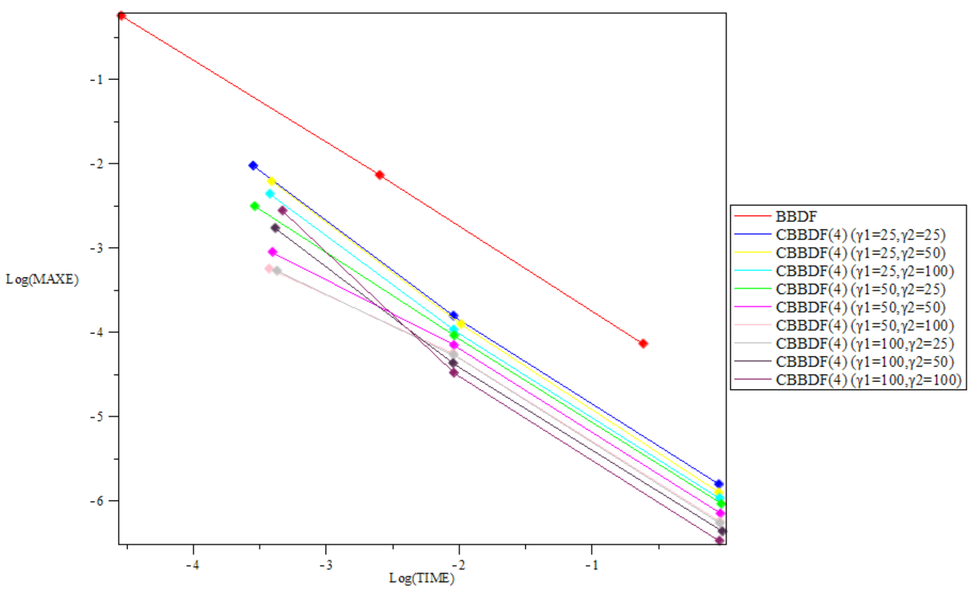

Figure 16: Efficiency graph for Example 5.4.

Overall, CBBDF(4) consistently outperforms the existing method, BBDF, across various step sizes, indicating its robustness and reliability in minimizing maximum error. For larger step sizes, $h = 10^{-1}$, parameter values that better minimize maximum error are often found to be $\gamma_1 = 25$ and $\gamma_2 = 100$. As the step size decreases to $h = 10^{-3}$ and $h = 10^{-5}$, the configurations tend to shift, with $\gamma_1 = 100$ and $\gamma_2 = 100$ emerging as the more effective settings for achieving improved accuracy.

From the analysis of Examples 5.1 and 5.2, it is evident that the proposed method, CBBDF(4), achieves better numerical results with reduced maximum errors, even when employing larger step sizes. In contrast, the traditional BBDF method requires smaller step sizes to ensure comparable accuracy. Specifically, for highly stiff problems, the BBDF method struggles to maintain acceptable accuracy levels, necessitating significantly reduced step sizes compared to those used by CBBDF(4). Notably, CBBDF(4) is capable of effectively addressing both mildly and highly stiff problems, a task that the conventional BBDF method often fails to accomplish.

In reviewing Tables 1 to 4, it is clear that the newly developed method, CBBDF(4), requires a slightly longer computation time compared to the existing BBDF method, particularly as the step size decreases. However, there are instances where the computational time for CBBDF(4) is less than or comparable to that of BBDF. For example, in Table 1, using a step size of $h = 10^{-1}$, the computational time for CBBDF(4) across all parameter values is less than that for BBDF. This advantage may arise because BBDF struggles to obtain numerical results for highly stiff problems when using larger step sizes. The overall increase in computational time for CBBDF(4) can be attributed to two main factors. First, the CBBDF(4) method involves additional calculations during the intermediate steps of Stage 1 before progressing to the next stage, significantly contributing to the overall computation time. Second, each iteration of the CBBDF(4) method necessitates computations for both Stage 1 and Stage 2. In contrast, the BBDF method utilizes an external startup computation with the Euler method at the beginning and then employs the BBDF method for each iteration. In summary, while CBBDF(4) demonstrates improved accuracy and performance, its computational demands are slightly higher due to the complexity of its iterative process and the requirement for additional calculations at each step.

6 Conclusion

This study introduces a self-starting three-stage composite block scheme based on the BDF method, which has been shown to be fourth-order accurate, convergent, and $A-$stable. In conclusion, the proposed method outperforms the existing BBDF method and achieves better numerical results with reduced maximum error, especially when using larger step sizes. Therefore, CBBDF(4) can serve as a valuable alternative solver for stiff IVPs of ODEs. For future research, CBBDF(4) could be applied to other models such as the electromagnetic transient in power systems to demonstrate its applicability in solving real-world problems.

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

Conflicts of Interest The authors declare no conflict of interest.

References

[1] N. Abd Rasid, Z. B. Ibrahim, Z. A. Majid & F. Ismail (2021). Formulation of a new implicit method for group implicit BBDF in solving related stiff ordinary differential equations. *Mathematics and Statistics*, *9*(2), 144–150. https://doi.org/10.13189/ms.2021.090208.

[2] K. J. Bathe & M. M. I. Baig (2005). On a composite implicit time integration procedure for nonlinear dynamics. *Computers & Structures*, *83*(31-32), 2513–2524. https://doi.org/10.1016/

j.compstruc.2005.08.001.

[3] X. Gao, C. S. Henriquez & W. Ying (2020). Composite backward differentiation formula for the bidomain equations. *Frontiers in Physiology*, *11*, Article ID: 591159. https://doi.org/10.3389/fphys.2020.591159.

[4] Z. B. Ibrahim, H. M. Ijam, S. J. Aksah & N. Abd Rasid (2024). Enhancing accuracy and efficiency in stiff ODE integration using variable step diagonal BBDF approaches. *Malaysian Journal of Fundamental and Applied Sciences*, *20*(5), 1083–1100. https://doi.org/10.11113/mjfas.v20n5.3530.

[5] Z. B. Ibrahim, N. Mohd Noor & K. I. Othman (2019). Fixed coefficient A (α) stable block backward differentiation formulas for stiff ordinary differential equations. *Symmetry*, *11*(7), Article ID: 846. https://doi.org/10.3390/sym11070846.

[6] Z. B. Ibrahim & A. A. Nasarudin (2020). A class of hybrid multistep block methods with A–stability for the numerical solution of stiff ordinary differential equations. *Mathematics*, *8*(6), Article ID: 914. https://doi.org/10.3390/math8060914.

[7] Z. B. Ibrahim, K. I. Othman & M. Suleiman (2007). Implicit r-point block backward differentiation formula for solving first-order stiff ODEs. *Applied Mathematics and Computation*, *186*(1), 558–565. https://doi.org/10.1016/j.amc.2006.07.116.

[8] Y. Ji & Y. Xing (2020). An optimized three-sub-step composite time integration method with controllable numerical dissipation. *Computers & Structures*, *231*, Article ID: 106210. https://doi.org/10.1016/j.compstruc.2020.106210.

[9] S. A. Junaidi, B. A. Jaafar & I. S. M. Zawawi (2024). Composite backward differentiation formulas for solving stiff ordinary differential equation. In *AIP Conference Proceedings*, volume 3189 pp. Article ID: 070005. AIP Publishing, Melville, New York. https://doi.org/10.1063/5.0224697.

[10] J. Li, H. Li, K. Yu & R. Zhao (2024). High-order accurate multi-sub-step implicit integration algorithms with dissipation control for hyperbolic problems. *Archive of Applied Mechanics*, *94*(8), 2285–2334. https://doi.org/10.1007/s00419-024-02637-y.

[11] H. Mohd Ijam, S. J. Aksah, A. F. N. Rasedee, N. Abd Rasid, A. Abdulsalam, N. H. Mohd Aris & F. Hazimi (2024). Numerical solutions of stiff chemical reaction problems using hybrid block backward differentiation formula. *Journal of Advanced Research in Numerical Heat Transfer*, *25*(1), 100–115. https://doi.org/10.37934/arnht.25.1.100115.

[12] H. Mohd Ijam & Z. B. Ibrahim (2019). Diagonally implicit block backward differentiation formula with optimal stability properties for stiff ordinary differential equations. *Symmetry*, *11*(11), Article ID: 1342. https://doi.org/10.3390/sym11111342.

[13] H. Mohd Ijam, Z. B. Ibrahim, Z. Abdul Majid & N. Senu (2020). Stability analysis of a diagonally implicit scheme of block backward differentiation formula for stiff pharmacokinetics models. *Advances in Difference Equations*, *2020*(1), 1–22. https://doi.org/10.1186/s13662-020-02846-z.

[14] I. S. Mohd Zawawi (2020). BBDF-Alpha for solving stiff ordinary differential equations with oscillating solutions. *Tamkang Journal of Mathematics*, *51*(2), 123–136. https://doi.org/10.5556/j.tkjm.51.2020.2964.

[15] H. Musa, M. Suleiman & N. Senu (2012). Fully implicit $3-$point block extended backward differentiation formula for stiff initial value problems. *Applied Mathematical Sciences*, *6*(85), 4211–4228.

[16] A. A. Nasarudin, Z. B. Ibrahim & H. Rosali (2020). On the integration of stiff ODEs using block backward differentiation formulas of order six. *Symmetry*, *12*(6), Article ID: 952. https://doi.org/10.3390/sym12060952.

[17] N. M. Noor, S. A. M. Yatim & Z. B. Ibrahim (2024). Fractional block method for the solution of fractional order differential equations. *Malaysian Journal of Mathematical Sciences*, *18*(1), 185–205. https://doi.org/10.47836/mjms.18.1.11.

[18] M. Rahmanzadeh & M. Barfeie (2019). An explicit time-stepping method based on error minimization for solving stiff system of ordinary differential equations. *Malaysian Journal of Mathematical Sciences*, *13*(3), 329–344.

[19] H. Ramos & M. A. Rufai (2023). A new one-step method with three intermediate points in a variable step-size mode for stiff differential systems. *Journal of Mathematical Chemistry*, *61*(4), 673–688. https://doi.org/10.1007/s10910-022-01427-7.

[20] M. A. Rufai, B. Carpentieri, A. A. Alimi & M. A. Babalola (2025). A new collocation method for solving some first-order differential models of biological systems. *Scientific African*, *28*, Article ID: e02735. https://doi.org/10.1016/j.sciaf.2025.e02735.

[21] M. A. Rufai, T. Tran & Z. A. Anastassi (2023). A variable step-size implementation of the hybrid Nyström method for integrating Hamiltonian and stiff differential systems. *Computational and Applied Mathematics*, *42*(4), Article ID:156. https://doi.org/10.1007/s40314-023-02273-2.

[22] C. Song & X. Zhang (2024). High-order composite implicit time integration schemes based on rational approximations for elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, *418*, Article ID: 116473. https://doi.org/10.1016/j.cma.2023.116473.

[23] H. Soomro, N. Zainuddin, H. Daud, J. Sunday, N. Jamaludin, A. Abdullah, M. Apriyanto & E. A. Kadir (2022). Variable step block hybrid method for stiff chemical kinetics problems. *Applied Sciences*, *12*(9), Article ID: 4484. https://doi.org/10.3390/app12094484.

[24] H. Soomro, N. Zainuddin, H. Daud, J. Sunday, N. Jamaludin, A. Abdullah, A. Mulono & E. A. Kadir (2023). $3-$point block backward differentiation formula with an off-step point for the solutions of stiff chemical reaction problems. *Journal of Mathematical Chemistry*, *61*(1), 75–97. https://doi.org/10.1007/s10910-022-01402-2.

[25] W. Ying, C. S. Henriquez & D. J. Rose (2011). Composite backward differentiation formula: An extension of the TR-BDF2 scheme. *Applied Numerical Mathematics*, *72*, 1–16.

[26] M. Z. M. Zabidi, Z. A. Majid & N. Senu (2014). Solving stiff differential equations using A-stable block method. *International Journal of Pure and Applied Mathematics*, *93*(3), 409–425. https://doi.org/10.12732/ijpam.v93i3.11.

[27] I. S. M. Zawawi, H. Aris & B. N. Jørgensen (2020). Transient analysis of electrical circuits using block backward differentiation formula. In *Proceedings of the 2020 The 9th International Conference on Informatics, Environment, Energy and Applications*, volume 2020 of *ICPS* pp. 99–103. Association for Computing Machinery, Amsterdam. https://doi.org/10.1145/3386762.3391922.

[28] H. Zhang, R. Zhang, Y. Xing & P. Masarati (2020). On the optimization of $n-$sub-step composite time integration methods. *Nonlinear Dynamics*, *102*(3), 1939–1962. https://doi.org/10.1007/s11071-020-06020-8.