ICCK Journal of Applied Mathematics
http://dx.doi.org/10.62762/JAM.2026.326643

RESEARCH ARTICLE

A Modified Third-Order Block Numerical Integration Technique for First Order Stiff and Oscillatory Differential Equations

Abdulrahman Adamu[1,*] and **Buhari Alhassan**[1]

[1] Department of Mathematics and Statistics, Al-Qalam University Katsina, Katsina State, Nigeria

Abstract

This paper proposes a modified two-point block backward differentiation formula (MBBDF) for the numerical solution of highly stiff and oscillatory ordinary differential equations. The method achieves third-order accuracy with a reduced error constant and computes two solution values simultaneously at each step. For the selected parameter value $\rho = -\frac{7}{8}$, stability analysis shows that the resulting scheme is A-stable, making it suitable for the numerical integration of stiff initial value problems. The method is also consistent and zero-stable, and hence convergent. Numerical experiments on benchmark stiff problems show that the relative accuracy of the considered methods depends on the problem and step size. While 2SBBDF generally produces the smallest maximum errors, the proposed MBBDF method maintains competitive accuracy and often requires substantially less computational time than 2SBBDF and 2ISBBDF. The results indicate that MBBDF provides a favorable compromise between numerical accuracy, stability, and computational efficiency for the solution of stiff and oscillatory ordinary differential equations.

Keywords: a-stability, order, convergence, block backward differentiation formula, stiff.

1 Introduction

Ordinary differential equations (ODEs) play a central role in the mathematical modeling of physical, biological, and engineering systems. Many real-world processes such as chemical reaction kinetics, atmospheric dynamics, population interactions, electrical networks, and mechanical vibrations are described by ODEs whose solutions evolve on vastly different time scales [1]. When these systems contain both rapidly decaying transient components and slowly varying dominant modes, the resulting differential equations are classified as stiff. Stiff ODEs present one of the most challenging classes of problems in numerical analysis due to the severe restrictions they impose on step sizes when explicit methods are used [2, 3].

Submitted: 11 May 2026
Revised: 16 July 2026
Accepted: 28 July 2026
Published: 04 September 2026

Vol. 2, **No.** 3, 2026.
10.62762/JAM.2026.326643

***Corresponding author:**
Abdulrahman Adamu
adamuaman@auk.edu.ng

Citation
Adamu, A., & Alhassan, B. (2026). A Modified Third-Order Block Numerical Integration Technique for First Order Stiff and Oscillatory Differential Equations. *ICCK Journal of Applied Mathematics*, 2(3), 242–250.

The defining feature of stiffness lies in the presence of large variations in the magnitudes of eigenvalues of the system's Jacobian matrix, which leads to strong sensitivity to perturbations in initial conditions and parameters. As a consequence, standard explicit schemes require extremely small step sizes to remain stable, making them highly inefficient or completely impractical for stiff systems [4]. To overcome these limitations, implicit integration methods have become the preferred tools for stiff initial value problems (IVPs). Backward differentiation formulas (BDFs), first systematically developed by Gear [5], remain among the most widely used implicit methods for stiff systems. These have been extended and generalized to yield extended backward differentiation formulas (EBDFs) [6], superclass of extended backward differentiation formulas (SEBDFs), implicit Runge–Kutta schemes, and block numerical methods that exploit stability advantages and improved error behavior [7–9].

Block methods, in particular, have gained attention for their ability to compute multiple solution values per integration step, increase parallel efficiency, and provide improved stability characteristics. Variants of block BDF techniques have been successfully employed for stiff and oscillatory differential equations [10–12]. Despite these advances, challenges remain in developing high-order, computationally efficient, and robust block methods that maintain strong stability properties, especially A-stability and L-stability, while accurately resolving stiff transients. In this study, we consider the following first-order stiff initial value problem:

$$v' = f(u,v), \qquad v(a) = v_0, \qquad a \le u \le b \tag{1}$$

The originality of this work is the development of a modified two-point block backward differentiation formula (MBBDF) that improves both accuracy and efficiency in solving stiff and oscillatory ordinary differential equations. The method attains third-order accuracy with a reduced error constant and maintains A-stability for $\rho = -\frac{7}{8}$. In addition, its block formulation allows for the simultaneous computation of two solution values, leading to improved performance compared to existing methods such as 2SBBDF and 2ISBBDF.

2 Formulation of the Method

Consider the 2-point superclass of the block backward differentiation formula (2SBBDF) developed by [3] which is given by

$$\sum_{j=0}^{3} \alpha_j y_{n+j-1} = h\beta_k \left(f_{n+k} - \rho f_{n+k-1}\right), \qquad k = 1,2. \tag{2}$$

We propose the following modification of (2).

$$\sum_{k=0}^{3} \sigma_k v_{n+k-1} = h\alpha_j \left(f_{n+j} + \rho f_{n+j-1}\right), \qquad j = 1,2. \tag{3}$$

The implicit scheme in equation (3) is formulated through a linear operator. The corresponding general linear operator L_j associated with (3) is defined as follows:

$$\begin{aligned} L_j\left[v(u_n), h\right]: \quad & \sigma_0 v_{n-1} + \sigma_1 v_n + \sigma_2 v_{n+1} + \sigma_3 v_{n+2} \\ & - h\alpha_j \left(f_{n+j} + \rho f_{n+j-1}\right) = 0. \end{aligned} \tag{4}$$

When $j = 1$, the formula for the first point is derived, and when $j = 2$, the formula for the second point is obtained. To illustrate this, consider the case $j = 1$ in equation (3), and define the corresponding operator L_1 as follows:

$$\begin{aligned} & \sigma_0 v\left(u_n - h\right) + \sigma_1 v\left(u_n\right) + \sigma_2 v\left(u_n + h\right) \\ & \quad + \sigma_3 v\left(u_n + 2h\right) - h\alpha_1 \left[f\left(u_n + h\right) + \rho f\left(u_n\right)\right] = 0. \end{aligned} \tag{5}$$

By expanding equation (5) in a Taylor series about u_n and grouping like terms, we obtain:

$$A_0 v\left(u_n\right) + A_1 h v'\left(u_n\right) + A_2 h^2 v''\left(u_n\right) + \ldots = 0, \tag{6}$$

where

$$\left.\begin{aligned} A_0 &= \sigma_0 + \sigma_1 + \sigma_2 + \sigma_3 = 0 \\ A_1 &= -\sigma_0 + \sigma_2 + 2\sigma_3 - \alpha_1(1+\rho) = 0 \\ A_2 &= \frac{1}{2}\sigma_0 + \frac{1}{2}\sigma_2 + 2\sigma_3 - \alpha_1 = 0 \\ A_3 &= -\frac{1}{6}\sigma_0 + \frac{1}{6}\sigma_2 + \frac{4}{3}\sigma_3 - \frac{1}{2}\alpha_1 = 0 \end{aligned}\right\}, \tag{7}$$

Next, we normalize the coefficient of v_{n+1} to 1. Solving the simultaneous system in equation (7) then yields values for σ_k and α_k (see Table 1).

Table 1. Coefficients of the first point formula.

σ_0	σ_1	σ_2	σ_3	α_1
$-\frac{1}{3}\frac{2\rho-1}{2\rho+1}$	$-\frac{\rho+2}{2\rho+1}$	1	$-\frac{1}{3}\frac{\rho-2}{2\rho+1}$	$\frac{2}{2\rho+1}$

Thus, substituting these values in (5), the first point is determined as follows:

$$\begin{aligned} v_{n+1} = &\frac{1}{3}\frac{2\rho-1}{2\rho+1}v_{n-1} + \frac{\rho+2}{2\rho+1}v_n \\ &+ \frac{1}{3}\frac{\rho-2}{2\rho+1}v_{n+2} + \frac{2}{2\rho+1}hf_{n+1} + \frac{2\rho}{2\rho+1}hf_n. \end{aligned} \tag{8}$$

The formula for the second point is derived by setting $j = 2$ in equation (4) and applying the same procedure used for the first point, yielding:

$$\begin{aligned} v_{n+2} = &-\frac{\rho-2}{2\rho+11}v_{n-1} + \frac{3(2\rho-3)}{2\rho+11}v_n - \frac{3(\rho-6)}{2\rho+11}v_{n+1} \\ &+ \frac{6}{2\rho+11}hf_{n+2} + \frac{6\rho}{2\rho+11}hf_{n+1}. \end{aligned} \tag{9}$$

Hence, the modified two-point superclass of block backward differentiation formula (MBBDF) is obtained as:

$$\left.\begin{aligned} v_{n+1} = &\frac{1}{3}\frac{2\rho-1}{2\rho+1}v_{n-1} + \frac{\rho+2}{2\rho+1}v_n + \frac{1}{3}\frac{\rho-2}{2\rho+1}v_{n+2} \\ &+ \frac{2}{2\rho+1}hf_{n+1} + \frac{2\rho}{2\rho+1}hf_n, \\ v_{n+2} = &-\frac{\rho-2}{2\rho+11}v_{n-1} + \frac{3(2\rho-3)}{2\rho+11}v_n - \frac{3(\rho-6)}{2\rho+11}v_{n+1} \\ &+ \frac{6}{2\rho+11}hf_{n+2} + \frac{6\rho}{2\rho+11}hf_{n+1} \end{aligned}\right\} \tag{10}$$

To ensure the method's absolute stability, we select ρ from the interval $(-1, 1)$ as discussed in [3, 13–15]. Setting $\rho = -\frac{7}{8}$ in equation (10) yields:

$$\left.\begin{aligned} v_{n+1} = &\frac{11}{9}v_{n-1} - \frac{3}{2}v_n + \frac{23}{18}v_{n+2} \\ &- \frac{8}{3}hf_{n+1} + \frac{7}{3}hf_n, \\ v_{n+2} = &\frac{23}{74}v_{n-1} - \frac{57}{37}v_n + \frac{165}{74}v_{n+1} \\ &+ \frac{24}{37}hf_{n+2} - \frac{21}{37}hf_{n+1} \end{aligned}\right\}. \tag{11}$$

3 Order and Error Constant of the Method

To determine the order of the method corresponding to equation (11), equation (11) can be rewritten as:

$$\left.\begin{aligned} &-\frac{11}{9}v_{n-1} + \frac{3}{2}v_n + v_{n+1} - \frac{23}{18}v_{n+2} \\ &\qquad = -\frac{8}{3}hf_{n+1} + \frac{7}{3}hf_n, \\ &-\frac{23}{74}v_{n-1} + \frac{57}{37}v_n - \frac{165}{74}v_{n+1} + v_{n+2} \\ &\qquad = \frac{24}{37}hf_{n+2} - \frac{21}{37}hf_{n+1} \end{aligned}\right\}, \tag{12}$$

Express (12) in matrix notation as follows:

$$\sum_{k=0}^{1} B_k^* V_{m+k-1} = h\sum_{k=0}^{1} C_k^* F_{m+k-1}, \tag{13}$$

where B_0^*, B_1^*, C_0^* and C_1^* are square matrices defined by:

$$B_0^* = \begin{pmatrix} -\frac{11}{9} & \frac{3}{2} \\ -\frac{23}{74} & \frac{57}{37} \end{pmatrix}, \qquad B_1^* = \begin{pmatrix} 1 & -\frac{23}{18} \\ -\frac{165}{74} & 1 \end{pmatrix},$$

$$C_0^* = \begin{pmatrix} 0 & \frac{7}{3} \\ 0 & 0 \end{pmatrix}, \qquad C_1^* = \begin{pmatrix} -\frac{8}{3} & 0 \\ -\frac{21}{37} & \frac{24}{37} \end{pmatrix}.$$

and V_{m-1}, V_m, F_{m-1} and F_m are column vectors defined by

$$V_m = \begin{pmatrix} v_{n+1} \\ v_{n+2} \end{pmatrix}, \qquad V_{m-1} = \begin{pmatrix} v_{n-1} \\ v_n \end{pmatrix},$$

$$F_{m-1} = \begin{pmatrix} f_{n-1} \\ f_n \end{pmatrix}, \qquad F_m = \begin{pmatrix} f_{n+1} \\ f_{n+2} \end{pmatrix}.$$

Equation (13) can be reformulated as

$$\begin{aligned} &\begin{pmatrix} -\frac{11}{9} & \frac{3}{2} \\ -\frac{23}{74} & \frac{57}{37} \end{pmatrix}\begin{pmatrix} v_{n-1} \\ v_n \end{pmatrix} + \begin{pmatrix} 1 & -\frac{23}{18} \\ -\frac{165}{74} & 1 \end{pmatrix}\begin{pmatrix} v_{n+1} \\ v_{n+2} \end{pmatrix} \\ &= h\begin{pmatrix} 0 & \frac{7}{3} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} f_{n-1} \\ f_n \end{pmatrix} + h\begin{pmatrix} -\frac{8}{3} & 0 \\ -\frac{21}{37} & \frac{24}{37} \end{pmatrix}\begin{pmatrix} f_{n+1} \\ f_{n+2} \end{pmatrix}. \end{aligned} \tag{14}$$

Let B_0^*, B_1^*, C_0^* and C_1^* be block matrices defined by

$$B_0^* = (B_0 \;\; B_1), \qquad B_1^* = (B_2 \;\; B_3),$$

$$C_0^* = (C_0 \;\; C_1), \qquad C_1^* = (C_2 \;\; C_3).$$

where

$$B_0 = \begin{pmatrix} -\frac{11}{9} \\ -\frac{23}{74} \end{pmatrix}, \qquad B_1 = \begin{pmatrix} \frac{3}{2} \\ \frac{57}{37} \end{pmatrix},$$

$$B_2 = \begin{pmatrix} 1 \\ -\frac{165}{74} \end{pmatrix}, \qquad B_3 = \begin{pmatrix} -\frac{23}{18} \\ 1 \end{pmatrix},$$

$$C_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad C_1 = \begin{pmatrix} \frac{7}{3} \\ 0 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} -\frac{8}{3} \\ -\frac{21}{37} \end{pmatrix}, \qquad C_3 = \begin{pmatrix} 0 \\ \frac{24}{37} \end{pmatrix}.$$

Definition 1: The order of the block method (11) and its associated linear operator given by:

$$L\left[v(u):h\right] = \sum_{k=0}^{3} \left[B_k v(u+kh) - hC_k v'(u+kh)\right], \tag{15}$$

is the unique integer p such that $D_0 = D_1 = D_2 = \ldots = D_p = 0$, and $D_{p+1} \neq 0$ where D_{p+1} is called the error constant. We have

$$D_0 = B_0 + B_1 + B_2 + B_3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

$$\begin{aligned} D_1 &= B_1 + 2B_2 + 3B_3 \\ &\quad - (C_0 + C_1 + C_2 + C_3) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

$$\begin{aligned} D_2 &= \frac{1}{2!}\left(B_1 + 2^2 B_2 + 3^2 B_3\right) \\ &\quad - \frac{1}{1!}\left(C_1 + 2C_2 + 3C_3\right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

$$\begin{aligned} D_3 &= \frac{1}{3!}\left(B_1 + 2^3 B_2 + 3^3 B_3\right) \\ &\quad - \frac{1}{2!}\left(C_1 + 2^2 C_2 + 3^2 C_3\right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

$$\begin{aligned} D_4 &= \frac{1}{4!}\left(B_1 + 2^4 B_2 + 3^4 B_3\right) \\ &\quad - \frac{1}{3!}\left(C_1 + 2^3 C_2 + 3^3 C_3\right) = \begin{pmatrix} -\frac{5}{12} \\ -\frac{31}{148} \end{pmatrix}. \end{aligned}$$

Therefore, the method (11) is of order 3 with error constant

$$D_4 = \begin{pmatrix} -\dfrac{5}{12} \\ -\dfrac{31}{148} \end{pmatrix}.$$

Definition 2 (Consistency). The MBBDF scheme is said to be consistent if it has order $p \geq 1$ [2].

Hence, the MBBDF scheme is consistent, as it achieves an order of accuracy $(p = 3)$, which exceeds the minimum requirement for consistency.

4 Stability Analysis of the Method

In this section, we examine the stability properties of method (11), focusing on zero-stability and A-stability. To do so, we consider the following test problem:

$$v' = \lambda v, \qquad \mathrm{Re}(\lambda) < 0, \qquad \lambda \text{ is a complex constant} \tag{16}$$

Substituting (16) into (11) with $f(u,v) = \lambda v$, we get

$$\left.\begin{aligned} v_{n+1} &= \frac{11}{9}v_{n-1} - \frac{3}{2}v_n + \frac{23}{18}v_{n+2} \\ &\quad - \frac{8}{3}\lambda h v_{n+1} + \frac{7}{3}\lambda h v_n, \\ v_{n+2} &= \frac{23}{74}v_{n-1} - \frac{57}{37}v_n + \frac{165}{74}v_{n+1} \\ &\quad + \frac{24}{37}\lambda h v_{n+2} - \frac{21}{37}\lambda h v_{n+1} \end{aligned}\right\}, \tag{17}$$

Set $\overline{h} = \lambda h$. By rearranging and combining like terms, we get

$$\begin{aligned} &\begin{pmatrix} 1 + \frac{8}{3}\overline{h} & -\frac{23}{18} \\ -\frac{165}{74} + \frac{21}{37}\overline{h} & 1 - \frac{24}{37}\overline{h} \end{pmatrix} \begin{pmatrix} v_{n+1} \\ v_{n+2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{11}{9} & -\frac{3}{2} + \frac{7}{3}\overline{h} \\ \frac{23}{74} & -\frac{57}{37} \end{pmatrix} \begin{pmatrix} v_{n-1} \\ v_n \end{pmatrix}. \end{aligned} \tag{18}$$

Let

$$E = \begin{pmatrix} 1 + \frac{8}{3}\overline{h} & -\frac{23}{18} \\ -\frac{165}{74} + \frac{21}{37}\overline{h} & 1 - \frac{24}{37}\overline{h} \end{pmatrix},$$

$$F = \begin{pmatrix} \frac{11}{9} & -\frac{3}{2} + \frac{7}{3}\overline{h} \\ \frac{23}{74} & -\frac{57}{37} \end{pmatrix}.$$

The stability polynomial of the method (11) is found by evaluating the determinant of $(Er - F)$ as follows:

$$\begin{aligned} \pi\left(r, \overline{h}\right) = &-\frac{821}{444}r^2 + \frac{203}{74}r^2\overline{h} + \frac{725}{222}r - \frac{64}{37}r^2\overline{h}^2 \\ &- \frac{128}{111}r\overline{h} - \frac{17}{12} + \frac{49}{37}r\overline{h}^2 - \frac{161}{222}\overline{h} = 0. \end{aligned} \tag{19}$$

By setting $\overline{h} = \lambda h = 0$ in (19), we obtain the first characteristic polynomial as:

$$-\frac{821}{444}r^2 - \frac{17}{12} + \frac{725}{222}r = 0. \tag{20}$$

To find the solution for r in equation (20), we get:

$$r = 1 \quad \text{and} \quad r = 0.7661388551.$$

Definition 3: The method (11) is said to be zero-stable if no root of the first characteristic polynomial has modulus greater than one and that any root with modulus one is simple [19]. Thus, according to the above definition, the method (11) is zero-stable.

Definition 4: A method (11) is said to be A-stable if its stability region covers the entire negative half-plane [20].

By plotting the stability polynomial (19) using Maple software, the absolute stability region of method (11) is obtained, as shown in Figure 1.

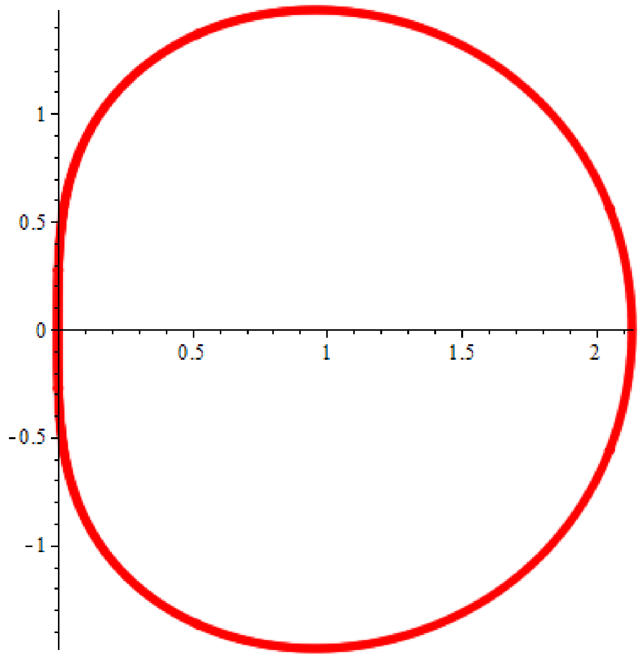

Figure 1. Stability region of MBBDF.

As shown in Figure 1, the absolute stability region corresponds to the region outside the boundary curve and contains the entire left half of the complex plane. Therefore, for the selected parameter value $\rho = -\frac{7}{8}$, method (11) is A-stable.

Theorem 1 (Dahlquist Equivalence Theorem): A block method is convergent if and only if it is both consistent and zero-stable [2].

Therefore, in accordance with the aforementioned theorem, the proposed MBBDF method is convergent, as it satisfies both the consistency and zero-stability conditions.

5 Implementation of the Method

Newton's iteration is employed to solve the nonlinear systems that emerge at each step of the method. The details of this iteration are provided below, starting with the definition of the error.

Definition 5: Let v_i and $v\,(u_i)$ be the approximate and exact solutions of (1) respectively. Then the absolute error is given by

$$(error_i)_t = \left|(v_i)_t - (v\,(u_i))_t\right|, \tag{21}$$

The maximum error is given by

$$MAXE = \max_{1\leq t\leq T}\left(\max_{1\leq i\leq N}(error_i)_t\right), \tag{22}$$

where T is the total number of steps and N is the number of equations. Define

$$\left.\begin{aligned} G_1 &= v_{n+1} - \frac{23}{18}v_{n+2} + \frac{8}{3}hf_{n+1} - \frac{7}{3}hf_n - \xi_1 \\ G_2 &= v_{n+2} - \frac{165}{74}v_{n+1} - \frac{24}{37}hf_{n+2} + \frac{21}{37}hf_{n+1} - \xi_2 \end{aligned}\right\}, \tag{23}$$

where

$$\left.\begin{aligned} \xi_1 &= \frac{11}{9}v_{n-1} - \frac{3}{2}v_n \\ \xi_2 &= \frac{23}{74}v_{n-1} - \frac{57}{37}v_n \end{aligned}\right\}. \tag{24}$$

Let $u_{n+k}^{(i+1)}$ denote the $(i+1)$th iteration and

$$e_{n+k}^{(i+1)} = v_{n+k}^{(i+1)} - v_{n+k}^{(i)}, \qquad k = 1,2 \tag{25}$$

Newton's iteration for the modified third order block numerical integration technique for stiff and oscillatory differential equations takes the form

$$v_{n+k}^{(i+1)} - v_{n+k}^{(i)} = -\left[G_k'\left(v_{n+k}^{(i)}\right)\right]^{-1}\left[G_k\left(v_{n+k}^{(i)}\right)\right], \quad k = 1,2, \tag{26}$$

which can be rewritten in the form

$$e_{n+k}^{(i+1)} = -\left[G_k'\left(v_{n+k}^{(i)}\right)\right]^{-1}\left[G_k\left(v_{n+k}^{(i)}\right)\right], \tag{27}$$

Therefore (27) can also be expressed as

$$\left[G_k'\left(v_{n+k}^{(i)}\right)\right]e_{n+k}^{(i+1)} = -\left[G_k\left(v_{n+k}^{(i)}\right)\right], \tag{28}$$

and in matrix form, equation (28) is equivalent to

$$\begin{pmatrix} 1+\frac{8}{3}h\frac{\partial f_{n+1}}{\partial v_{n+1}} & -\frac{23}{18} \\ -\frac{165}{74}+\frac{21}{37}h\frac{\partial f_{n+1}}{\partial v_{n+1}} & 1-\frac{24}{37}h\frac{\partial f_{n+2}}{\partial v_{n+2}} \end{pmatrix}\begin{pmatrix} e_{n+1}^{(i+1)} \\ e_{n+2}^{(i+1)} \end{pmatrix}$$
$$= \begin{pmatrix} -1 & \frac{23}{18} \\ \frac{165}{74} & -1 \end{pmatrix}\begin{pmatrix} v_{n+1}^{(i)} \\ v_{n+2}^{(i)} \end{pmatrix} + h\begin{pmatrix} 0 & \frac{7}{3} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} f_{n-1}^{(i)} \\ f_{n}^{(i)} \end{pmatrix}$$
$$+ h\begin{pmatrix} -\frac{8}{3} & 0 \\ -\frac{21}{37} & \frac{24}{37} \end{pmatrix}\begin{pmatrix} f_{n+1}^{(i)} \\ f_{n+2}^{(i)} \end{pmatrix} + \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}. \tag{29}$$

A program in C language has been written to implement (29).

6 Test Problems and Numerical Results

To verify the efficiency of the developed method, the following stiff ODEs are solved:

Problem 1:

$$\begin{aligned} v_1' &= -21v_1+19v_2-20v_3, && v_1(0)=1, \quad 0\le u\le 10,\\ v_2' &= 19v_1-21v_2+20v_3, && v_2(0)=0,\\ v_3' &= 40v_1-40v_2-40v_3, && v_3(0)=-1. \end{aligned}$$

Exact Solution:

$$\begin{aligned} v_1 &= 0.5\left(e^{-2u}+e^{-40u}(\cos 40u+\sin 40u)\right),\\ v_2 &= 0.5\left(e^{-2u}-e^{-40u}(\cos 40u+\sin 40u)\right),\\ v_3 &= 0.5\left[\left(2e^{-40u}\right)(\sin 40u-\cos 40u)\right]. \end{aligned}$$

Eigenvalues: $\lambda_1 = -2$, $\lambda_2 = -40+40i$ and $\lambda_3 = -40-40i$. Source: [15].

Problem 2:

$$\begin{aligned} v_1' &= 32v_1+66v_2+\tfrac{2}{3}u+\tfrac{2}{3}, && v_1(0)=\tfrac{1}{3}, \quad 0\le u\le 10,\\ v_2' &= -66v_1-133v_2-\tfrac{1}{3}u-\tfrac{1}{3}, && v_2(0)=\tfrac{1}{3}. \end{aligned}$$

Exact Solution:

$$\begin{aligned} v_1 &= \tfrac{2}{3}u+\tfrac{2}{3}e^{-u}-\tfrac{1}{3}e^{-100u},\\ v_2 &= -\tfrac{1}{3}u-\tfrac{1}{3}e^{-u}+\tfrac{2}{3}e^{-100u}. \end{aligned}$$

Eigenvalues: $\lambda_1=-1$ and $\lambda_2=-100$ [16].

Problem 3:

$$\begin{aligned} v_1' &= -0.1v_1-49.9v_2, && v_1(0)=2, \quad 0\le u\le 10,\\ v_2' &= -50v_2, && v_2(0)=1,\\ v_3' &= 70v_2-120v_3, && v_3(0)=2. \end{aligned}$$

Exact Solution:

$$\begin{aligned} v_1 &= e^{-0.1u}+e^{-50u},\\ v_2 &= e^{-50u},\\ v_3 &= e^{-50u}+e^{-120u}. \end{aligned}$$

Eigenvalues: $\lambda_1 = -0.1$, $\lambda_2 = -120$ and $\lambda_3 = -50$ [17].

The three test problems were solved using the 2-point Super Class of Block Backward Differentiation Formula (2SBBDF), the Improved Extended 2-point Super Class of Block Backward Differentiation Formula (2ISBBDF), and the proposed Modified 2-point Super Class of Block Backward Differentiation Formula (MBBDF). The numerical performance of the three methods was evaluated in terms of the maximum absolute error (MAXE) and CPU time for different step sizes h. The corresponding results for Problems 1–3 are presented in Tables 2, 3, and 4, respectively.

The following notation is used in Tables 2, 3, and 4: h denotes the step size; METHOD denotes the numerical method employed; MAXE denotes the maximum absolute error; and TIME denotes the CPU time. The abbreviations 2SBBDF, 2ISBBDF, and MBBDF refer to the 2-point Super Class of Block Backward Differentiation Formula [3], the Improved Extended 2-point Super Class of Block Backward Differentiation Formula [18], and the proposed Modified 2-point Super Class of Block Backward Differentiation Formula, respectively.

Table 2. Numerical results for Problem 1.

h	METHOD	MAXE	TIME
10^{-2}	2SBBDF	1.36641E-001	2.91800E-002
	2ISBBDF	3.31871E-001	3.21400E-002
	MBBDF	2.31185E-001	1.31400E-004
10^{-3}	2SBBDF	4.66162E-003	4.21400E-002
	2ISBBDF	3.62150E-002	3.47200E-002
	MBBDF	1.16618E-002	1.03100E-003
10^{-4}	2SBBDF	4.73311E-005	1.76100E-001
	2ISBBDF	3.42614E-004	4.69200E-001
	MBBDF	1.56774E-004	1.49600E-002
10^{-5}	2SBBDF	4.73388E-007	1.58600E+000
	2ISBBDF	3.72614E-005	4.91400E+000
	MBBDF	1.60216E-006	2.52400E-001
10^{-6}	2SBBDF	4.73388E-009	1.50000E+001
	2ISBBDF	3.94225E-006	6.27900E+001
	MBBDF	1.60331E-008	1.49600E+001

7 Discussion of Results

The numerical results for Problems 1–3 are presented in Tables 2, 3, and 4, respectively. The comparison shows that the three methods exhibit different trade-offs between accuracy and computational

Table 3. Numerical results for Problem 2.

h	METHOD	MAXE	TIME
10^{-2}	2SBBDF	1.21627E-002	3.30800E-002
	2ISBBDF	1.21794E-002	3.99100E-002
	MBBDF	1.21478E-002	2.04100E-004
10^{-3}	2SBBDF	7.01465E-003	3.36800E-002
	2ISBBDF	4.38292E-003	5.53900E-002
	MBBDF	1.20969E-002	2.35100E-003
10^{-4}	2SBBDF	9.53582E-005	3.71400E-002
	2ISBBDF	6.86698E-004	3.17300E-002
	MBBDF	2.66607E-004	2.71400E-003
10^{-5}	2SBBDF	9.82872E-007	8.43000E-002
	2ISBBDF	7.22478E-005	8.12700E-002
	MBBDF	3.21381E-006	2.97900E-002
10^{-6}	2SBBDF	9.85846E-009	6.05000E-001
	2ISBBDF	7.26167E-006	5.88600E-001
	MBBDF	3.32159E-008	4.62000E-002

Table 4. Numerical results for Problem 3.

h	METHOD	MAXE	TIME
10^{-2}	2SBBDF	6.68584E-002	2.75100E-002
	2ISBBDF	6.46866E-002	3.38600E-002
	MBBDF	1.16872E-001	2.30100E-004
10^{-3}	2SBBDF	1.72593E-002	3.63400E-002
	2ISBBDF	1.07438E-002	3.52300E-002
	MBBDF	2.93166E-002	2.52400E-003
10^{-4}	2SBBDF	2.40970E-004	1.27900E-001
	2ISBBDF	1.63804E-003	1.32400E-001
	MBBDF	6.66252E-004	2.71400E-002
10^{-5}	2SBBDF	2.49090E-006	1.07000E+000
	2ISBBDF	1.72475E-004	1.05900E+000
	MBBDF	8.12343E-006	2.04100E-001
10^{-6}	2SBBDF	2.49916E-008	1.03600E+001
	2ISBBDF	1.73371E-005	1.04200E+001
	MBBDF	8.41678E-008	2.62400E-001

efficiency. In general, the maximum errors decrease as the step size is reduced, indicating improved numerical accuracy under step-size refinement.

For Problem 1, the 2SBBDF method produces the smallest maximum errors for all the tested step sizes. The MBBDF method is less accurate than 2SBBDF but is generally more accurate than 2ISBBDF for the smaller step sizes. For example, at $h = 10^{-6}$, the maximum errors obtained by 2SBBDF, 2ISBBDF, and MBBDF are 4.73388×10^{-9}, 3.94225×10^{-6}, and 1.60331×10^{-8}, respectively. Thus, MBBDF provides an accuracy level close to that of 2SBBDF while substantially improving upon 2ISBBDF at fine step sizes.

For Problem 2, the relative accuracy of the methods depends on the selected step size. At $h = 10^{-2}$, MBBDF gives the smallest maximum error among the three methods. At finer step sizes, however, 2SBBDF generally produces the smallest errors. Nevertheless, MBBDF remains more accurate than 2ISBBDF for $h = 10^{-4}$, 10^{-5}, and 10^{-6}. This indicates that the accuracy advantage of MBBDF over the comparison methods is problem- and step-size-dependent rather than uniform.

A similar behavior is observed for Problem 3. At the relatively large step sizes $h = 10^{-2}$ and $h = 10^{-3}$, MBBDF produces larger maximum errors than both 2SBBDF and 2ISBBDF. As the step size decreases, its accuracy improves considerably. For $h = 10^{-4}$, 10^{-5}, and 10^{-6}, MBBDF gives smaller errors than 2ISBBDF, although 2SBBDF remains the most accurate method among the three.

The computational-time results show a clearer advantage for MBBDF over much of the tested range. In most of the experiments, particularly for moderate and small step sizes, MBBDF requires less CPU time than 2SBBDF and 2ISBBDF. For instance, in Problem 2 at $h = 10^{-6}$, the CPU times for 2SBBDF, 2ISBBDF, and MBBDF are 6.05000×10^{-1}, 5.88600×10^{-1}, and 4.62000×10^{-2}, respectively. Similarly, for Problem 3 at $h = 10^{-6}$, MBBDF requires 2.62400×10^{-1}, compared with 1.03600×10^{1} and 1.04200×10^{1} for 2SBBDF and 2ISBBDF, respectively. An exception occurs for Problem 1 at $h = 10^{-6}$, where the CPU time of MBBDF becomes comparable to that of 2SBBDF.

Overall, the numerical results do not indicate that MBBDF is uniformly more accurate than the existing methods. Instead, they demonstrate that MBBDF provides a useful compromise between numerical accuracy and computational efficiency. While 2SBBDF generally achieves the smallest maximum errors, MBBDF often requires substantially less computational time and, at sufficiently small step sizes, generally produces smaller errors than 2ISBBDF. These results suggest that MBBDF can be advantageous when computational efficiency is an important consideration while maintaining competitive numerical accuracy.

8 Conclusion

This work introduced a modified two-point block backward differentiation formula (MBBDF) for the numerical solution of highly stiff and oscillatory ordinary differential equations. The theoretical analysis shows that the proposed method is of order three and is consistent and zero-stable. For the selected parameter value $\rho = -\frac{7}{8}$, the resulting scheme is shown to be A-stable and is therefore suitable for the numerical treatment of stiff initial value problems.

The numerical experiments demonstrate that the relative accuracy of MBBDF, 2SBBDF, and 2ISBBDF depends on both the test problem and the step size. The 2SBBDF method generally produces the smallest maximum errors, whereas MBBDF exhibits competitive accuracy and often performs better than 2ISBBDF at smaller step sizes. Therefore, the numerical results do not suggest a uniform accuracy advantage of MBBDF over all the comparison methods.

The main practical advantage observed for MBBDF is its computational efficiency. For most of the tested step sizes, the proposed method requires less CPU time than 2SBBDF and 2ISBBDF, while maintaining acceptable numerical accuracy. This advantage is particularly evident in Problems 2 and 3 at the smaller step sizes considered.

Overall, the proposed MBBDF method provides a favorable compromise between accuracy, stability, and computational efficiency. The results suggest that it can serve as a practical alternative for the numerical solution of stiff and oscillatory ordinary differential equations, particularly in applications where computational cost is an important consideration. Further investigation using additional stiff test problems and variable-step implementations would be useful for assessing its performance over a broader range of applications.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Hairer, E., & Wanner, G. (1996). *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems* (Vol. 2). Springer. https://www.researchgate.net/publication/200175469

[2] Lambert, J. D. (1991). *Numerical Methods for Ordinary Differential Systems*. New York: John Wiley & Sons. https://dl.acm.org/doi/abs/10.5555/129839

[3] Suleiman, M. B., Musa, H., Ismail, F., Senu, N., & Ibrahim, Z. B. (2014). A new superclass of block backward differentiation formula for stiff ordinary differential equations. *Asian-european journal of mathematics, 7*(01), 1350034. [CrossRef]

[4] Zawawi, I. S. M., Ibrahim, Z. B., Ismail, F., & Majid, Z. A. (2012). Diagonally implicit block backward differentiation formulas for solving ordinary differential equations. *International journal of mathematics and mathematical sciences, 2012*(1), 767328. [CrossRef]

[5] Gear, C. W. (1971). The automatic integration of ordinary differential equations. *Communications of the ACM, 14*(3), 176–179. [CrossRef]

[6] Cash, J. R. (1980). On the integration of stiff systems of ODEs using extended backward differentiation formulae. *Numerische Mathematik, 34*(3), 235-246. [CrossRef]

[7] Mohd Zawawi, I. S., Ibrahim, Z. B., & Othman, K. I. (2015). Derivation of diagonally implicit block backward differentiation formulas for solving stiff initial value problems. *Mathematical problems in engineering, 2015*(1), 179231. [CrossRef]

[8] Yatim, S. A. M., Ibrahim, Z. B., Othman, K. I., & Suleiman, M. B. (2013). A numerical algorithm for solving stiff ordinary differential equations. *Mathematical Problems in Engineering, 2013*(1), 989381. [CrossRef]

[9] Ibrahim, Z. B., Othman, K. I., & Suleiman, M. (2007). Implicit r-point block backward differentiation formula for solving first-order stiff ODEs. *Applied Mathematics and Computation, 186*(1), 558-565. [CrossRef]

[10] Abdulganiy, R. I., Akinfenwa, O. A., & Okunuga, S. A. (2018). Construction of L stable second derivative trigonometrically fitted block backward differentiation formula for the solution of oscillatory initial value problems. *African Journal of Science, Technology, Innovation and Development, 10*(4), 411-419. https://hdl.handle.net/10520/EJC-fa42b17e4

[11] Ijam, H. M., Ibrahim, Z. B., & Zawawi, I. S. M. (2024). Stiffly stable diagonally implicit block backward differentiation formula with adaptive step size strategy

for stiff ordinary differential equations. *Matematika*, 27-47. [CrossRef]

[12] Nasarudin, A. A., Ibrahim, Z. B., & Rosali, H. (2020). On the integration of stiff ODEs using block backward differentiation formulas of order six. *Symmetry, 12*(6), 952. [CrossRef]

[13] Mohd Ijam, H., & Ibrahim, Z. B. (2019). Diagonally implicit block backward differentiation formula with optimal stability properties for stiff ordinary differential equations. *Symmetry, 11*(11), 1342. [CrossRef]

[14] Jana Aksah, S., Ibrahim, Z. B., & Mohd Zawawi, I. S. (2019). Stability analysis of singly diagonally implicit block backward differentiation formulas for stiff ordinary differential equations. *Mathematics, 7*(2), 211. [CrossRef]

[15] Suleiman, M. B., Musa, H., Ismail, F., & Senu, N. (2013). A new variable step size block backward differentiation formula for solving stiff initial value problems. *International Journal of Computer Mathematics, 90*(11), 2391-2408. [CrossRef]

[16] Bala, N., Alhassan, B., & Musa, H. (2022). A new 3-point implicit block numerical method for solving first order stiff initial value problems. *Timbou-Africa Academic Publications, International Journal of Science Research and Technology*, 9(9). ISSN: 2623-7861. https://www.researchgate.net/publication/366720880

[17] Musa, H., Suleiman, M. B., Ismail, F., Senu, N., Majid, Z. A., & Ibrahim, Z. B. (2014). A new fifth order implicit block method for solving first order stiff ordinary differential equations. *Malaysian Journal of Mathematical Sciences, 8*(S), 45-59. https://www.researchgate.net/publication/316516689

[18] Abdulrahman, A., & Hamisu, M. (2025). An Improved Extended 2-Point Super Class of Block Backward Differentiation Formula for Solving Stiff Initial Value Problems. *Journal of Science and Technology, 17*(1), 11-22. https://journal.uthm.edu.my/index.php/JST/article/view/20763

[19] Lambert, J. D. (1973). *Computational Methods in Ordinary Differential Equations*. New York: John Wiley & Sons.

[20] Dahlquist, G. G. (1963). A special stability problem for linear multistep methods. *BIT Numerical Mathematics, 3*(1), 27-43. [CrossRef]

Abdulrahman Adamu is a PhD candidate in the Department of Mathematics, Umaru Musa Yar'adua University, Katsina, Nigeria. He is currently a staff member in the Department of Mathematics and Statistics, Al-Qalam University, Katsina. His research interests include Numerical Analysis and Computational Mathematics, with a particular focus on the development of numerical methods for solving stiff initial value problems (IVPs). (Email: adamuaman@auk.edu.ng)

Buhari Alhassan is a Lecturer in the Department of Mathematics and Statistics, Al-Qalam University, Katsina, Nigeria. He obtained his NCE in Mathematics/Chemistry from Isah Kaita College of Education, Dutsinma, and BSc, MSc, and PhD degrees in Mathematics (Numerical Analysis) from Umaru Musa Yar'adua University, Katsina, Nigeria. His research interests include Numerical Analysis and Scientific Computing, with particular emphasis on the development and implementation of numerical methods for ordinary and fractional differential equations. He has published several papers in peer-reviewed journals and is actively involved in teaching and research. (Email: buharialhassan@auk.edu.ng)