DATA ANALYTICS AND APPLIED MATHEMATICS
E-ISSN: 2773-4854
VOLUME 6, ISSUE 2, 2025, 16 – 30
DOI: https://doi.org/10.15282/daam.v6i2.12656

RESEARCH ARTICLE

Designing an implicit Block Backward Differentiation Formula (BBDF) for stiff ordinary differential equations

Nur Anis Izwany Azhari[1], Norshakila Abd Rasid[2,*], Hazizah Mohd Ijam[3], Roslah Arsad[2], Mohamad Hidayad Ahmad Kamal[2], Noorazlina Mohamid Salih[4], and Saufianim Jana Aksah[5]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Negeri Sembilan Branch, Seremban 3 Campus, 70300, Seremban 3, Negeri Sembilan, Malaysia
[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch, Tapah Campus, 35400 Tapah Road, Perak, Malaysia
[3]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[4]Malaysia Electrical Engineering Technology Section, Universiti Kuala Lumpur Malaysian Institute of Marine Engineering Technology, Jalan Pantai Remis, 32200 Lumut Perak, Malaysia
[5]Centre of Foundation Studies, Universiti Teknologi MARA, Cawangan Selangor, Kampus Dengkil, 43800 Dengkil, Selangor, Malaysia

ABSTRACT - First-order Ordinary Differential Equations (ODEs) are often characterized by stiffness, especially in models that describe complex real-world processes. This work presents a four-point, fixed-coefficient, diagonally implicit block backward differentiation formula (4BBDF) of second order, developed to address the numerical challenges associated with stiffness. The formulation incorporates a diagonal matrix into the Lagrange interpolation polynomial and is constructed using Maple to ensure accuracy and stability. Newton's method is used to handle the nonlinear systems that arise. The proposed 4BBDF method is mathematically verified to be consistent, zero-stable, A-stable, and of second-order accuracy. Its implementation in C++ shows improved computational efficiency, reducing the number of steps required by approximately 50% when compared to existing methods. These results indicate that the proposed scheme is a reliable and effective tool for solving stiffness in ODE.

ARTICLE HISTORY

Received	:	20th June 2025
Revised	:	16th Aug 2025
Accepted	:	18th Aug 2025
Published	:	30th Sept 2025

KEYWORDS
BBDF
Stiffness
Diagonally Implicit
First order ODEs

1. INTRODUCTION

Stiff problems frequently arise in scientific and engineering applications and are known to be particularly challenging to analyse and solve due to their complex structure and dynamic behaviour [1-5]. We consider stiff initial value problems for first-order systems of ordinary differential equations (ODEs), posed in the general form in (1):

$$Y' = AY + F(x, Y), \quad Y(a) = \varphi, \tag{1}$$

where $Y(x) = (y_1, \dots, y_s)^T, \varphi = (\varphi_1, \dots, \varphi_s)^T$, and A is an $s \times s$ matrix with eigenvalues having large negative real parts. The interval of integration is $x \in [a, b]$ [6].

Although the concept of stiffness is widely acknowledged, a single, universally accepted definition remains elusive. As a result, multiple characterizations have been proposed in the literature. One common feature is that the system is stable, with no eigenvalues whose real parts are significantly positive, and at least one eigenvalue has a strongly negative real part, leading to rapid decay in some components [6]. Another indicator is that the time step is dictated more by stability constraints than by accuracy requirements [7]. This variation in time scales creates significant challenges for numerical methods, especially explicit ones, which typically require small step sizes to ensure stability [8]. Additionally, stiff systems tend to have eigenvalues with negative real parts and a large ratio between the highest and smallest magnitudes of those values [7]. In this study, we adopt the definition of stiffness presented in reference [9].

Definition 1.1 The linear system (1) is said to be stiff if (i) $\text{Re}\lambda_t < 0, t = 1,2, \dots, m$, and (ii) $max_{t=1,2,\dots,m}|\text{Re}\lambda_t| \gg min_{t=1,2,\dots,m}|\text{Re}\lambda_t|$ where $\lambda_t, t = 1,2, \dots, m$, are the eigenvalues of A. The ratio

$$\left[max_{t=1,2,\dots,m}|\text{Re}\lambda_t|\right]:\left[min_{t=1,2,\dots,m}|\text{Re}\lambda_t|\right] \tag{2}$$

is called stiffness ratio.

One of the most effective methods for solving stiff systems is the Backward Differentiation Formula (BDF). The method was introduced in [10] and belongs to the class of implicit multistep methods. It is particularly well-suited for stiff problems due to its strong stability properties. Unlike explicit approaches, BDF uses several previous time steps to estimate the current value, requiring the solution of an algebraic system at each step. Although this increases

***CORRESPONDING AUTHOR | Norshakila Abd Rasid**| ✉ norshakila@uitm.edu.my

computational effort, the method's stability makes it a preferred choice for many stiff applications. Comprehensive discussions on BDF, including its derivation, stability, and applications to stiff systems, are provided in [6–8].

Rosser [11] first introduced the concept of block computation through the Runge–Kutta framework, which was extended and adapted to various other numerical methods. A version of the implicit BDF scheme that incorporates block approaches was introduced in [12]. These methods compute solution values at multiple time points simultaneously, reducing overhead from step-by-step evaluations and improving overall efficiency. In an r-point block method, the values $y_{n+1},\ldots, y_{n+r}$, are obtained concurrently at the time steps $x_{n+1},\ldots, x_{n+r}$, based on prior solutions and their derivatives [12-15]. Collectively, these methods show that block techniques can effectively solve stiff problems by calculating multiple solution values at once, while still maintaining both stability and accuracy [16-19].

Recent developments in time integration strategies include diagonally implicit formulations and stability-oriented methods for solving stiff ODEs. Fully implicit schemes are known for their robustness and precision but are limited by the computational burden of solving large, coupled nonlinear systems. Therefore, diagonally implicit structures such as the Diagonally Implicit Runge–Kutta (DIRK) methods [20] and the Diagonally Implicit BBDF introduced in [21] have gained significant attention. The Diagonally scheme employs a lower triangular coefficient matrix with identical diagonal entries, often referred to as singly diagonally implicit. This structure enables each stage to require only one nonlinear solve, significantly reducing computational effort. As a result, such methods strike an effective balance between stability, accuracy, and efficiency, making them well-suited for stiff problems. For the BBDF framework, methods are developed to incorporate diagonally implicit schemes differing in the number of solution points and the use of previous values [21-24].

Most block BDF methods described in the literature are limited to two or three points due to concerns about computational cost and implementation difficulty. Previous studies have demonstrated that increasing the block size can significantly reduce the total number of integration steps. For example, Ismail et al. [25] observed that 3-point block methods consistently required fewer steps than comparable 2-point approaches, as they compute more solution values within each block. Building on this concept, the present work proposes a four-point implicit Diagonal Block Backward Differentiation Formula (4BBDF) as an advancement in this method class. By progressing four solution values per block, the proposed formulation aims to maintain both the accuracy and stability necessary for solving stiff problems, while also reducing the required integration steps. To keep the scheme feasible, the diagonally implicit structure introduced in [21-22] is retained. The resulting method delivers an efficient and practical alternative to current multistep techniques, seeking to address a gap in numerical approaches for stiff ordinary differential equations.

The subsequent sections outline the detailed formulation of the proposed method, followed by its theoretical analysis and application to selected test problems, along with a discussion of the corresponding results.

2. METHODOLOGY

Before implementing any new numerical method, a rigorous theoretical examination is necessary to ensure its validity and practical effectiveness. The 4BBDF method, like other multistep schemes, must be analysed in terms of its formulation, order of accuracy, convergence behaviour, and stability characteristics. Without such analysis, the method risks producing unreliable results, particularly when applied to stiff systems or long-term simulations. This study follows established numerical analysis principles by systematically evaluating these key properties in Sections 2.1 to 2.5 to confirm the robustness of the proposed method.

Section 2.1 introduces the derivation of the 4BBDF formula. Section 2.2 discusses the method's order, which reflects its level of approximation accuracy. Section 2.3 addresses the convergence analysis to ensure that the numerical solution approaches the exact solution as the step size decreases. Section 2.4 investigates the absolute stability region by plotting the characteristic polynomial in the complex plane, assessing the method's performance for stiff problems. Finally, Section 2.5 outlines the implementation process, including the computational procedures required to apply the method effectively.

2.1 Derivation of the Formula

The formulation of the 4-point Block Backward Differentiation Formula (4BBDF) is derived using Lagrange interpolation, starting from the general polynomial form:

$$P_k(x) = \sum_{j=0}^{k} L_{k,j}(x) f(x_j). \tag{3}$$

where the Lagrange basis polynomial is defined as:

$$L_{k,j}(x) = \prod_{\substack{i=0\\ i\neq j}}^{k} \frac{(x - x_i)}{(x_j - x_i)}, \mathrm{j} = 0, 1, \ldots, \mathrm{k} \tag{4}$$

The scheme simultaneously computes the four solution values $y_{n+1}, y_{n+2}, y_{n+3}$ and y_{n+4} , using two previous values and interpolating through a growing set of nodes, as illustrated in Figure 1.

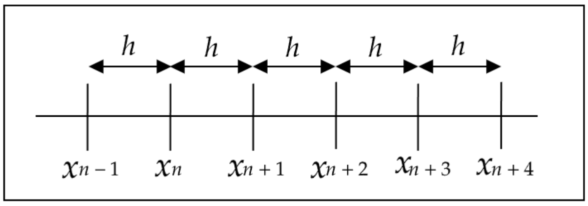

Figure 1. Interpolation points of 4BBDF

Following the method described in [21] and [22], we define scalar quantity $s = \frac{x - x_{n+1}}{h}$, which represents the localised displacement from x_{n+1} in units of the step size h. This scalar variable simplifies the interpolation by rescaling the nodes to a dimensionless form centred at x_{n+1}. Using this transformation, we construct a second-degree Lagrange interpolating polynomial to approximate y_{n+1} based on the values at $(y_{n-1}, x_{n-1}), (y_n, x_n)$ and (y_{n+1}, x_{n+1}):

$$P_2(x) = \frac{(x - x_n)(x - x_{n+1})}{(x_{n-1} - x_n)(x_{n-1} - x_{n+1})} y_{n-1} + \cdots + \frac{(x - x_{n-1})(x - x_n)}{(x_{n+1} - x_{n-1})(x_{n+1} - x_n)} y_{n+1} \tag{5}$$

The process is extended for y_{n+2}, y_{n+3} and y_{n+4} by increasing the number of interpolation points accordingly.

For y_{n+2} (4-point interpolation):

$$\begin{aligned} P_3(x) = {} & \frac{(x - x_n)(x - x_{n+1})(x - x_{n+2})}{(x_{n-1} - x_n)(x_{n-1} - x_{n+1})(x_{n-1} - x_{n+2})} y_{n-1} + \cdots \\ & + \frac{(x - x_{n-1})(x - x_n)(x - x_{n+1})}{(x_{n+2} - x_{n-1})(x_{n+2} - x_n)(x_{n+2} - x_{n+1})} y_{n+2} \end{aligned} \tag{6}$$

For y_{n+3} (5-point interpolation):

$$\begin{aligned} P_4(x) = {} & \frac{(x - x_n)(x - x_{n+1})(x - x_{n+2})(x - x_{n+3})}{(x_{n-1} - x_n)(x_{n-1} - x_{n+1})(x_{n-1} - x_{n+2})(x_{n-1} - x_{n+3})} y_{n-1} + \cdots \\ & + \frac{(x - x_{n-1})(x - x_n)(x - x_{n+1})(x - x_{n+2})}{(x_{n+3} - x_{n-1})(x_{n+3} - x_n)(x_{n+3} - x_{n+1})(x_{n+3} - x_{n+2})} y_{n+3} \end{aligned} \tag{7}$$

For y_{n+4} (6-point interpolation):

$$\begin{aligned} P_5(x) = {} & \frac{(x - x_n)(x - x_{n+1})(x - x_{n+2})(x - x_{n+3})(x - x_{n+4})}{(x_{n-1} - x_n)(x_{n-1} - x_{n+1})(x_{n-1} - x_{n+2})(x_{n-1} - x_{n+3})(x_{n-1} - x_{n+4})} y_{n-1} + \cdots \\ & + \frac{(x - x_{n-1})(x - x_n)(x - x_{n+1})(x - x_{n+2})(x - x_{n+3})}{(x_{n+4} - x_{n-1})(x_{n+4} - x_n)(x_{n4} - x_{n+1})(x_{n+4} - x_{n+2})(x_{n+4} - x_{n+3})} y_{n+4} \end{aligned} \tag{8}$$

Let the grid points be $x_{n-1} = 0$, $x_n = h$, $x_{n+1} = 2h$, $x_{n+2} = 3h, x_{n+3} = 4h$ and $x_{n+4} = 5h$. Then set $x = x_{n+1} + sh$, and differentiate each interpolation polynomial with respect to s, then evaluate $s = 0,1,2,3$ to obtain the 4BBDF scheme:

$$\begin{aligned} y_{n+1} &= -\frac{1}{3} y_{n-1} + \frac{4}{3} y_n + \frac{2}{3} h f_{n+1} \\ y_{n+2} &= \frac{2}{11} y_{n-1} - \frac{9}{11} y_n + \frac{18}{11} y_{n+1} + \frac{6}{11} h f_{n+2} \\ y_{n+3} &= -\frac{3}{25} y_{n-1} + \frac{16}{25} y_n - \frac{36}{25} y_{n+1} + \frac{48}{25} y_{n+2} + \frac{12}{25} h f_{n+3} \\ y_{n+4} &= \frac{12}{137} y_{n-1} - \frac{75}{137} y_n + \frac{200}{137} y_{n+1} - \frac{300}{137} y_{n+2} + \frac{300}{137} y_{n+3} + \frac{60}{137} h f_{n+4} \end{aligned} \tag{9}$$

This scheme employs a Predict-Evaluate-Correct-Evaluate (PECE) approach. The predictor step uses interpolation at earlier points x_{n-2}, x_{n-1} and x_n to estimate initial guesses for y_{n+1} through y_{n+4}. The general predictor formula is

$$\mathrm{P}(x)=\frac{(x-x_{n-1})(x-x_n)}{(x_{n-2}-x_{n-1})(x_{n-2}-x_n)}y_{n-2}+\frac{(x-x_{n-2})(x-x_n)}{(x_{n-1}-x_{n-2})(x_{n-1}-x_n)}y_{n-1}+\frac{(x-x_{n-2})(x-x_{n-1})}{(x_n-x_{n-2})(x_n-x_{n-1})}y_n \tag{10}$$

Set $x_{n-2}=0,\ x_{n-1}=h,\ x_n=2h,\ x_{n+1}=3h,\ x_{n+2}=4h, x_{n+3}=5h$ and $x_{n+4}=6h$. Substitute $x=x_{n+1}+sh$ and evaluate for $s=0,1,2,3$ to yield the predictor values:

$$\begin{aligned} y_{n+1}^{(p)} &= y_{n-2}-3y_{n-1}+3y_n \\ y_{n+2}^{(p)} &= 3y_{n-2}-8y_{n-1}+6y_n \\ y_{n+3}^{(p)} &= 6y_{n-2}-15y_{n-1}+10y_n \\ y_{n+4}^{(p)} &= 10y_{n-2}-24y_{n-1}+15y_n \end{aligned} \tag{11}$$

2.2 Order of Method

The order of the 4BBDF method can be determined by expressing the linear multistep method (LMM) in matrix form, as outlined in [26]. The general form k-step LMM is:

$$\sum_{j=0}^{k} A_j y_{n+j} = h\sum_{j=0}^{k} B_j f_{n+\mathrm{j}} \tag{12}$$

where A_j and B_j are real-valued coefficient matrices of size 4 × 4, and k denotes the number of steps. By organizing Equation (9) into its coefficient matrices, we obtain:

$$A_0=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, A_1=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, A_2=\begin{bmatrix}\frac{1}{3}\\ \frac{-2}{11}\\ \frac{3}{25}\\ \frac{-12}{137}\end{bmatrix}, A_3=\begin{bmatrix}\frac{-4}{3}\\ \frac{9}{11}\\ \frac{-16}{25}\\ \frac{75}{137}\end{bmatrix}, A_4=\begin{bmatrix}1\\ \frac{-18}{11}\\ \frac{36}{25}\\ \frac{-200}{137}\end{bmatrix},\quad A_5=\begin{bmatrix}0\\1\\ \frac{-48}{25}\\ \frac{300}{137}\end{bmatrix}, A_6=\begin{bmatrix}0\\0\\1\\ \frac{-300}{137}\end{bmatrix},$$

$$A_7=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}, B_1=B_2=B_3=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, B_4=\begin{bmatrix}\frac{2}{3}\\0\\0\\0\end{bmatrix}, B_5=\begin{bmatrix}0\\ \frac{6}{11}\\0\\0\end{bmatrix}, B_6=\begin{bmatrix}0\\0\\ \frac{12}{25}\\0\end{bmatrix}, B_7=\begin{bmatrix}0\\0\\0\\ \frac{60}{137}\end{bmatrix} \tag{13}$$

Definition 2.1 A linear multistep method as given in Equation (12), along with its associated linear operator

$$L[y(x);h]=\textstyle\sum_{j=0}^{k}\left[A_j y(x+jh)-hB_j y'(x+jh)\right] \tag{14}$$

is said to be of order q if the coefficients satisfy the following order conditions:

$$C_0=C_1=C_2=\cdots=C_q=0 \ \text{ and } C_{q+1}\neq 0 \tag{15}$$

The constant C_{q+1} is referred to as the error constant [9].

To verify the order of the method, the constants C_i are computed by applying the following formula [9-10]:

$$C_\mathrm{i}=\textstyle\sum_{j=0}^{k}\left[\frac{j^i}{i!}A_j-\frac{j^{i-1}}{(i-1)!}B_j\right],\quad i=0,1,2,\ldots \tag{16}$$

To determine the order, substitute $j=0,1,2,\ldots$ and let $k=7$ into the coefficient expression as follows:

$$C_0=\sum_{j=0}^{7} A_j = \mathrm{A}_0+\mathrm{A}_1+\mathrm{A}_2+\mathrm{A}_3+\mathrm{A}_4+\mathrm{A}_5+\mathrm{A}_6+\mathrm{A}_7=\begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{17}$$

$$C_1 = \sum_{j=0}^{7}\left(jA_j - B_j\right) = 0 \cdot \mathrm{A}_0 + 1 \cdot \mathrm{A}_1 + 2 \cdot \mathrm{A}_2 + 3 \cdot \mathrm{A}_3 + 4 \cdot \mathrm{A}_4 + 5 \cdot \mathrm{A}_5 + 6 \cdot \mathrm{A}_6 + 7 \cdot \mathrm{A}_7$$

$$-(B_1 + B_2 + B_3 + B_4 + B_5 + B_6 + B_7) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

$$C_2 = \sum_{j=0}^{7}\left(\frac{\mathrm{j(j-1)}}{2!}A_j - jB_j\right)$$

$$= \frac{1}{2}(0^2 \cdot A_0 + 1^2 \cdot A_1 + 2^2 \cdot A_2 + 3^2 \cdot A_3 + 4^2 \cdot A_4 + 5^2 \cdot A_5 + 6^2 \cdot A_6 + 7^2 \cdot A_7)$$

$$-(1 \cdot B_1 + 2 \cdot B_2 + 3 \cdot B_3 + 4 \cdot B_4 + 5 \cdot B_5 + 6.B_6 + 7 \cdot B_7) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{19}$$

$$C_3 = \sum_{j=0}^{7}\left(\frac{j^3 A_j}{3!} - \frac{j^{3-1}B_j}{(3-1)!}\right)$$

$$= \frac{1}{6}(0^3 \cdot A_0 + 1^3 \cdot A_1 + 2^3 \cdot A_2 + 3^3 \cdot A_3 + 4^3 \cdot A_4 + 5^3 \cdot A_5 + 6^3 \cdot A_6 + 7^3 \cdot A_7)$$

$$-\frac{1}{2}(1^2 \cdot B_1 + 2^2 \cdot B_2 + 3^2 \cdot B_3 + 4^2 \cdot B_4 + 5^2 \cdot B_5 + 6^2 \cdot B_6 + 7^2 \cdot B_7) = \begin{bmatrix} \frac{-2}{9} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

Since $C_0 = C_1 = C_2 = 0$ and $C_3 \neq 0$, we can conclude that the 4BBDF is of order 2.

2.3 Convergence

According to Hall and Watt [27], a linear multistep method (LMM) is convergent if only if it satisfies the following definition:

Definition 2.2 A linear multistep method is convergent if it is both consistent and zero-stable.

We begin by verifying the consistency of the 4BBDF method.

Definition 2.3 A linear multistep method (12) is said to be consistent if it has order $q \geq 1$. Equivalently, the method is consistent if the following two conditions are satisfied: [9]

$$\sum_{j=0}^{k} A_j = 0; \qquad \sum_{j=0}^{k} jA_j = \sum_{j=0}^{k} B_j = \tag{21}$$

$$(i) \sum_{j=0}^{7} A_j = A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{22}$$

$$(ii)\ \sum_{j=0}^{7} jA_j = \sum_{j=0}^{7} B_7 = 0\cdot \mathrm{A}_0 + 1\cdot \mathrm{A}_1 + 2\cdot \mathrm{A}_2 + 3\cdot \mathrm{A}_3 + 4\cdot \mathrm{A}_4 + 5\cdot \mathrm{A}_5 + 6\cdot \mathrm{A}_6 + 7\cdot \mathrm{A}_7$$

$$= B_1 + B_2 + B_3 + B_4 + B_5 + B_6 + B_7 = \begin{bmatrix} \frac{2}{3} \\ \frac{6}{11} \\ \frac{12}{25} \\ \frac{60}{137} \end{bmatrix} \tag{23}$$

Conditions (i) and (ii) ensure that the local truncation error vanishes to at least order one, thereby confirming that the method is consistent.

Next, we verify the zero-stability of the method. The following definition is used to determine this property:

Definition 2.4 The linear multistep method defined by Equation (9) is said to be zero-stable if no root of its first characteristic polynomial has modulus greater than one, and if every root with modulus one is simple. [28]

To assess this, we express Equation (9) in matrix form:

$$AY_m = \mathrm{B}Y_{m-1} \tag{24}$$

Letting $\hat{h} = \lambda h$, the matrices are defined as:

$$A = \begin{bmatrix} 1-\frac{2}{3}\hat{\mathrm{h}} & 0 & 0 & 0 \\ \frac{-18}{11} & 1-\frac{6}{11}\hat{\mathrm{h}} & 0 & 0 \\ \frac{36}{25} & \frac{-48}{25} & 1-\frac{12}{25}\hat{\mathrm{h}} & 0 \\ \frac{-200}{137} & \frac{300}{137} & \frac{-300}{137} & 1-\frac{60}{137}\hat{\mathrm{h}} \end{bmatrix}, Y_m = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \\ y_{n+4} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & \frac{1}{3} & \frac{-4}{3} \\ 0 & 0 & \frac{-2}{11} & \frac{9}{11} \\ 0 & 0 & \frac{3}{25} & \frac{-16}{25} \\ 0 & 0 & \frac{-12}{137} & \frac{75}{137} \end{bmatrix}, \quad Y_{m-1} = \begin{bmatrix} y_{n-3} \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix} \tag{25}$$

The first characteristic polynomial is given by $\rho(t,\hat{h}) = \det\,(tA - B)$. Substituting $\hat{h} = 0$, we obtain:

$$\rho(t,0) = \mathrm{t}^4 - \frac{113602}{113025}\mathrm{t}^3 + \frac{577}{113025}\mathrm{t}^2 \tag{26}$$

Solving $\rho(t) = 0$ yields the characteristic roots:

$$\mathrm{t} = 0, t = 1, t = 0.005105\ldots \tag{27}$$

Since all roots lie within or on the unit circle $|t| \leq 1$ and the root $\mathrm{t} = 1$ is simple, the method satisfies the zero-stability condition. As both consistency and zero-stability are achieved, we conclude that the 4BBDF method is convergent.

2.4 Stability region of 4BBDF

The stability region of the 4BBDF method is investigated graphically using *Maple* software, in accordance with the following definitions as described in [8]:

Definition 2.5 A linear multistep method (LMM) is said to be absolutely stable in a region of the complex plane $\Re$ (real part) such that for all $h \in \Re$, all roots of the stability polynomial $\rho(t, h)$ related to the method, $|t_s| < 1,\ s = 1,2, \dots, k$.

Definition 2.6 A method is said to be A-stable if the region of absolute stability $\Re$ includes the entire left half of the complex plane, that is, for all $Re(h\lambda) < 0$ where h is the step size and λ is a characteristic value of the differential equation.

To analyse this, the substitution, $t = e^{i\theta}$ is applied to the stability polynomial and related functions. The resulting plot reveals the boundary of the stability region in the complex plane.

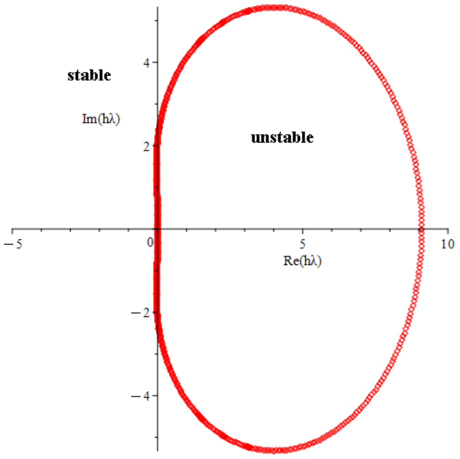

Figure 2. Stability region of 4BBDF

Figure 2 illustrates that the modulus condition $|t| < 1$ is satisfied for certain values of $h\lambda$ lying in the left half-plane. This demonstrates that the method is unstable in the region where $Re(h\lambda) > 0$ and stable where $Im(h\lambda)$ dominates with $Re(h\lambda) < 0$. Therefore, the 4BBDF method satisfies the condition for A-stability, as its absolute stability region include the entire left half of the complex plane.

2.5 Implementation of Method

This section outlines the implementation of the 4BBDF method using Newton's iteration, adapted from the approach described by [12]. The Newton iteration formula for solving the nonlinear system associated with the 4BBDF method is given by:

$$y_{n+j}^{(i+1)} = y_{n+j}^{(i)} - \left[\mathrm{F}_j^{(i)}\left(y_{n+j}^{(i)}\right)\right] . \left[\mathrm{F'}_j^{(i)}\left(y_{n+j}^{(i)}\right)\right]^{-1}, \qquad j = 1,2,3,4 \tag{28}$$

Here, $y_{n+j}^{(i)}$ denoted the i-th iterative approximation of the unknowns $y_{n+1}, y_{n+2}, y_{n+3}$ and y_{n+4}, for $j = 1,2,3,4,$ respectively. The absolute error between successive iterations is defined as:

$$e_{n+j}^{(i+1)} = y_{n+j}^{(i+1)} - y_{n+j}^{(i)}, \qquad j = 1,2,3,4 \tag{29}$$

Using this notation, Equation (28) can be rewritten as a linearized system:

$$\left[F'_j\left(y_{n+j}^{(i)}\right)\right] . e_{n+j}^{(i+1)} = -\left[F_j\left(y_{n+j}^{(i)}\right)\right], \qquad j = 1,2,3,4. \tag{30}$$

The nonlinear function F_j, derived from the 4BBDF formula in Equation (9) are written as follows:

$$\begin{aligned} F_1 &= y_{n+1} - \frac{2}{3} h f_{n+1} - \vartheta_1 \\ F_2 &= y_{n+2} - \frac{18}{11} - \frac{6}{11} h f_{n+2} - \vartheta_2 \\ F_3 &= y_{n+3} + \frac{36}{25} y_{n+1} - \frac{48}{25} y_{n+2} - \frac{12}{25} h f_{n+3} - \vartheta_3 \end{aligned} \tag{31}$$

$$F_4 = y_{n+4} - \frac{200}{137}y_{n+1} + \frac{300}{137}y_{n+2} - \frac{300}{137}y_{n+3} - \frac{60}{137}hf_{n+4} - \vartheta_4$$

where ϑ_1, ϑ_2, ϑ_3, ϑ_4 represent known combinations of previous values from the method.y. These functions are substituted into the Newton update in Equation (31), resulting the following linear system:

$$\begin{bmatrix} 1 - \frac{2}{3}h\frac{\partial f_{n+1}}{\partial y_{n+1}} & 0 & 0 & 0 \\ -\frac{18}{11} & 1 - \frac{6}{11}h\frac{\partial f_{n+2}}{\partial y_{n+2}} & 0 & 0 \\ \frac{36}{25} & -\frac{48}{25} & 1 - \frac{12}{25}h\frac{\partial f_{n+3}}{\partial y_{n+3}} & 0 \\ -\frac{200}{137} & \frac{300}{137} & -\frac{300}{137} & 1 - \frac{60}{137}h\frac{\partial f_{n+4}}{\partial y_{n+4}} \end{bmatrix} \begin{bmatrix} e_{n+1}^{(i+1)} \\ e_{n+2}^{(i+1)} \\ e_{n+3}^{(i+1)} \\ e_{n+4}^{(i+1)} \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 & 0 & 0 \\ \frac{18}{11} & -1 & 0 & 0 \\ -\frac{36}{25} & \frac{48}{25} & -1 & 0 \\ \frac{200}{137} & -\frac{300}{137} & \frac{300}{137} & -1 \end{bmatrix} \begin{bmatrix} y_{n+1}^{(i)} \\ y_{n+2}^{(i)} \\ y_{n+3}^{(i)} \\ y_{n+4}^{(i)} \end{bmatrix} + h \begin{bmatrix} \frac{2}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{6}{11} & 0 & 0 & 0 \\ 0 & 0 & \frac{12}{25} & 0 & 0 \\ 0 & 0 & 0 & \frac{60}{137} & 0 \end{bmatrix} \begin{bmatrix} f_{n+j}^{(i)} \\ f_{n+j}^{(i)} \\ f_{n+j}^{(i)} \\ f_{n+j}^{(i)} \end{bmatrix} + \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ \vartheta_3 \\ \vartheta_4 \end{bmatrix} \qquad (32)$$

This linear system is solved at each Newton iteration to obtain the correction vector $e^{(i+1)}$. The updated values of the solution are then computed as:

$$y_{n+j}^{(i+1)} = y_{n+j}^{(i)} + e_{n+j}^{(i+1)}, \qquad j = 1,2,3,4 \qquad (33)$$

The iterative process is repeated until a predefined convergence criterion is satisfied. This ensures that the 4BBDF method yields sufficiently accurate results and is effectively applied at each step of the numerical approximation.

3. NUMERICAL EXPERIMENT

To evaluate the performance of the proposed 4BBDF method, a set of established benchmark problems consisting of linear stiff ordinary differential equations is employed. These test cases are widely used in numerical analysis to assess the accuracy and robustness of numerical solvers for stiff initial value problems. The problems selected include both single equations and systems of first-order ODEs, each with known exact solutions for comparison.

Problem 1

$$y' = 100(\sin x - y), \qquad y(0) = 0, \qquad 0 \leq x \leq 3$$

Exact solution:

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

Eigenvalue:

$$\lambda = -100$$

Source: [28]

Problem 2

$$y' = -20y + 20\sin x + \cos x, \qquad y(0) = 1, \qquad 0 \leq x \leq 2$$

Exact solution:

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

Eigenvalue:

$$\lambda = -20$$

Source: [29]

Problem 3

$$y_1{}' = -20y_1 - 19y_2, \qquad y_1(0) = 2, \qquad 0 \le x \le 5$$
$$y_2{}' = -19y_1 - 20y_2, \qquad y_2(0) = 0,$$

Exact solution:

$$y_1(x) = e^{-39x} + e^{-x}, \qquad y_2(x) = e^{-39x} - e^{-x}$$

Eigenvalue:

$$\lambda_1 = -1, \lambda_2 = -39$$

Source: [30]

Problem 4

$$y_1' = -32y_1 + 66y_2 + \frac{2}{3}x + \frac{2}{3} \qquad y_1(0) = \frac{1}{3}, 0 \le x \le 5$$
$$y_2' = -66y_1 - 133y_2 - \frac{1}{3}x - \frac{1}{3} \qquad y_2(0) = \frac{1}{3}$$

Exact solution:

$$y_1(x) = \frac{2}{3}x + \frac{2}{3}e^{-x} + \frac{1}{3}e^{-100x}, \qquad y_2(x) = -\frac{1}{3}x - \frac{1}{3}e^{-x} + \frac{2}{3}e^{-100x}$$

Eigenvalue:

$$\lambda_1 = -1, \lambda_2 = -100$$

Source [29]

Problem 5

$$y_1' = -9y_1 + 24y_2 + 5\cos x - \frac{1}{3}\sin x, \qquad y_1(0) = \frac{4}{3}, \qquad 0 \le x \le 10$$
$$y_2' = -24y_1 - 51y_2 + 9\cos x + \frac{1}{3}\sin x, \qquad y_2(0) = \frac{2}{3}$$

Exact solution:

$$y_1(x) = 2e^{-3x} - e^{-39x} + \frac{1}{3}\cos x, \qquad y_2(x) = 2e^{-39x} - e^{-3x} - \frac{1}{3}\cos x$$

Eigenvalue:

$$\lambda_1 = -3, \lambda_2 = -39$$

Source: [30]

4. RESULTS AND DISCUSSION

The performance of the proposed 4BBDF method is evaluated through comparison with established methods of similar class. The numerical results are assessed based on the maximum global error (MAXE) and the total number of steps (STEPs) required for each test case. The computations are conducted using fixed step sizes $h = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ and 10^{-6}.

To provide a visual comparison of the accuracy, log–log plots of MAXE versus h are presented in Figures 3 to 7. These graphs illustrate the error behaviour of the tested methods across different problems and highlight the convergence trend as the step size decreases.

The notations used in the tables are defined as follows:

h : Step size
MAXE : Maximum error
$STEPs$: Overall steps taken
BBDF : Implicit 2-point BBDF of proposed in [12]
DOBBDF(2) : Diagonal Implicit BBDF with Off-step Points in [22]
4BBDF : Derived method

The error at each step is calculated as

$$\boldsymbol{err_i} = y_{exact,i} - y_{approx,i} \tag{34}$$

where $y_{exact,i}$ is the exact value and $y_{approx,i}$ is the numerical approximation at the i-th grid point.
The maximum error (MAXE) is the defined by

$$\text{MAXE} = \max_{1 \le \text{i} \le \text{NS}} \left| y_{exact,i} - y_{approx,i} \right| \tag{35}$$

where NS denotes the total numbers of computed steps [12].

Table 1. Numerical Result for Problem 1

h	Methods	MAXE	$STEPs$
10^{-2}	BBDF	1.82733e−004	150
	DOBBDF(2)	7.85138e−004	151
	4BBDF	1.82771e−004	76
10^{-3}	BBDF	1.15700e−004	1500
	DOBBDF(2)	3.50485e−005	1501
	4BBDF	1.21950e−004	751
10^{-4}	BBDF	1.55714e−006	15000
	DOBBDF(2)	4.03031e−007	15001
	4BBDF	1.61643e−006	7501
10^{-5}	BBDF	1.60347e−008	150000
	DOBBDF(2)	4.09940e−009	150001
	4BBDF	1.67517e−008	75001
10^{-6}	BBDF	1.60817e−010	1500000
	DOBBDF(2)	4.10637e−011	1500001
	4BBDF	1.68115e−010	750001

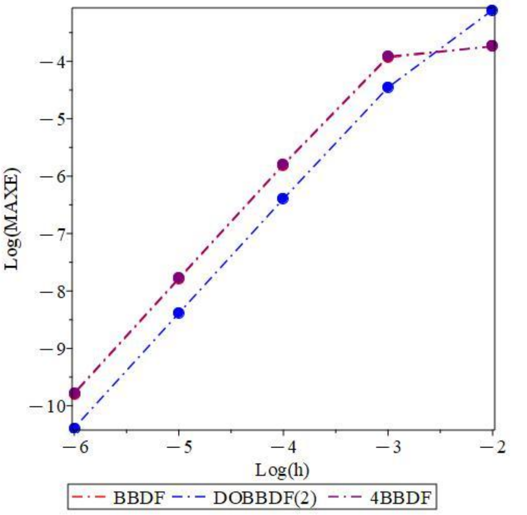

Figure 3. Graph of log(MAXE) vs log(h) for Problem 1

Table 2. Numerical Result for Problem 2

h	Methods	MAXE	$STEPs$
10^{-2}	BBDF	3.29311e−002	100
	DOBBDF(2)	1.20640e−002	101
	4BBDF	3.52096e−002	51
10^{-3}	BBDF	6.02846e−004	1000
	DOBBDF(2)	1.58193e−004	1001
	4BBDF	6.26871e−004	501
10^{-4}	BBDF	6.39304e−006	10000
	DOBBDF(2)	1.63667e−006	10001
	4BBDF	6.67419e−006	5001
10^{-5}	BBDF	6.43060e−008	100000
	DOBBDF(2)	1.64224e−008	100001
	4BBDF	6.72195e−008	50001
10^{-6}	BBDF	6.43436e−010	1000000
	DOBBDF(2)	1.64279e−010	1000001
	4BBDF	6.72674e−010	500001

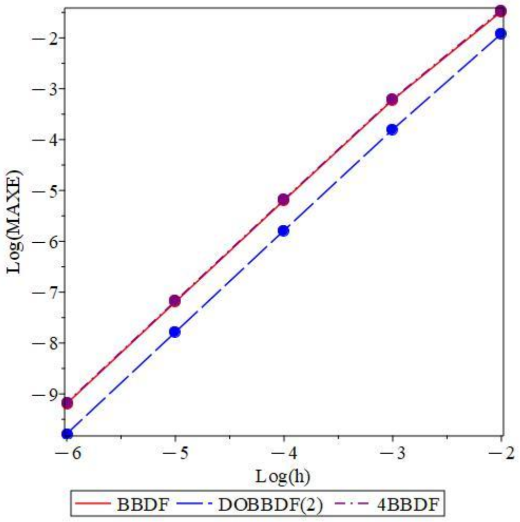

Figure 4. Graph of log(MAXE) vs log(h) for Problem 2

Table 3. Numerical Result for Problem 3

h	Methods	MAXE	$STEPs$
10^{-2}	BBDF	6.29433e−002	250
	DOBBDF(2)	3.42431e−002	251
	4BBDF	6.85453e−002	126
10^{-3}	BBDF	2.15556e−003	2500
	DOBBDF(2)	5.83901e−004	2501
	4BBDF	2.24905e−003	1251
10^{-4}	BBDF	2.41757e−005	25000
	DOBBDF(2)	6.20525e−006	25001
	4BBDF	2.52050e−005	12501
10^{-5}	BBDF	2.44533e−007	250000
	DOBBDF(2)	6.24648e−008	250001
	4BBDF	2.55578e−007	125001
10^{-6}	BBDF	2.44812e−009	2500000
	DOBBDF(2)	6.25061e−010	2500001
	4BBDF	2.55933e−009	1250001

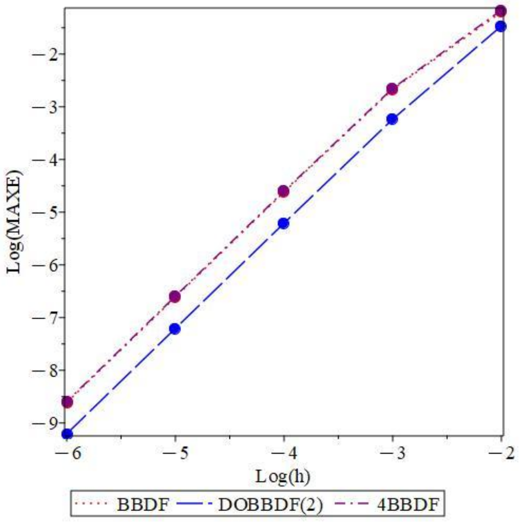

Figure 5. Graph of log(MAXE) vs log(h) for Problem 3

Table 4. Numerical Result for Problem 4

h	Methods	MAXE	$STEPs$
10^{-2}	BBDF	1.21585e−002	50
	DOBBDF(2)	5.23314e−002	51
	4BBDF	1.21566e−002	26
10^{-3}	BBDF	7.71283e−003	500
	DOBBDF(2)	2.33643e−003	501
	4BBDF	8.12948e−003	251
10^{-4}	BBDF	1.03804e−004	5000
	DOBBDF(2)	2.68674e−005	5001
	4BBDF	1.07756e−004	2501
10^{-5}	BBDF	1.06893e−006	50000
	DOBBDF(2)	2.73280e−007	50001
	4BBDF	1.11672e−006	25001
10^{-6}	BBDF	1.07206e−008	500000
	DOBBDF(2)	2.73744e−009	500001
	4BBDF	1.12071e−008	250001

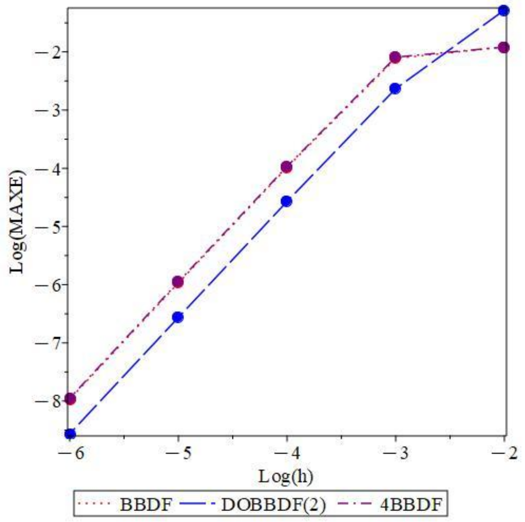

Figure 6. Graph of log(MAXE) vs log(h) for Problem 4

Table 5. Numerical Result for Problem 5

h	Methods	MAXE	$STEPs$
10^{-2}	BBDF	1.24297e−001	500
	DOBBDF(2)	6.80670e−002	501
	4BBDF	1.35436e−001	251
10^{-3}	BBDF	4.29409e−003	5000
	DOBBDF(2)	1.16348e−003	5001
	4BBDF	4.48045e−003	2501
10^{-4}	BBDF	4.81799e−005	50000
	DOBBDF(2)	1.23667e−005	50001
	4BBDF	5.02308e−005	25001
10^{-5}	BBDF	4.87351e−007	500000
	DOBBDF(2)	1.24491e−007	500001
	4BBDF	5.09362e−007	250001
10^{-6}	BBDF	4.87909e−009	5000000
	DOBBDF(2)	1.24574e−009	5000001
	4BBDF	5.10073e−009	2500001

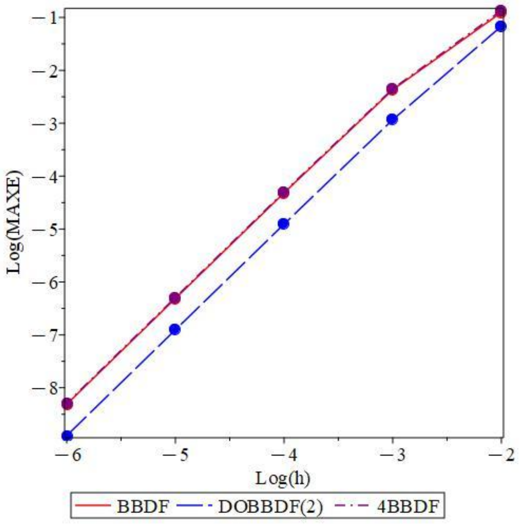

Figure 7. Graph of log(MAXE) vs log(h) for Problem 2

Tables 1–5 and Figures 3–7 present the numerical results demonstrating the performance of the tested methods across a range of step sizes h. As h decreases, the maximum error (MAXE) consistently reduces, reflecting the convergence behavior noted by [12]. Figures 3–7 further show that the 4BBDF method yields error trends that closely align with those of the BBDF method at smaller step sizes, though its accuracy remains slightly lower than that of DOBBDF(2). Notably, the 4BBDF method achieves competitive MAXE values while requiring significantly fewer computational steps than the fully implicit BBDF method.

The enhanced accuracy observed in DOBBDF(2) can be attributed to the use of off-step points, which improve approximation quality. However, this improvement comes at the cost of a substantially higher number of time steps. In contrast, the 4BBDF method offers an efficient compromise between accuracy and computational effort, reducing the number of steps by approximately half while maintaining acceptable error levels benefiting from its diagonally implicit structure and the use of additional points, as highlighted in [21], [22] and [25]. This feature makes 4BBDF a practical and computationally efficient approach for solving stiff ordinary differential equations (ODEs).

Overall, all methods show improved accuracy with decreasing step size, confirming the expected convergence trend. Among them, 4BBDF is distinguished by its ability to significantly reduce computational overhead without compromising reliability, underscoring its effectiveness for stiff ODE problems.

5. CONCLUSIONS

In this study, a 4-point Diagonally Implicit Block Backward Differentiation Formula (4BBDF) of order two was formulated for the efficient numerical solution of stiff first-order ordinary differential equations (ODEs). The method incorporates a diagonal coefficient matrix and is implemented using Maple software, which facilitates numerical stability, simplifies the algorithmic structure, and enhances computational efficiency through improved Newton iteration schemes.

Theoretical analysis confirms that the 4BBDF method satisfies the essential criteria of consistency, zero-stability, and A-stability, making it well-suited for stiff problems. Numerical experiments further support its reliability, demonstrating competitive accuracy while requiring approximately half the computational steps compared to conventional fully implicit methods.

Overall, the 4BBDF scheme, developed within the Block BDF framework, offers a robust and efficient numerical tool with significant potential for application in scientific and engineering problems involving stiff systems.

ACKNOWLEDGEMENTS

Institution(s)

All the authors would like to express their gratitude to Universiti Teknologi MARA, Perak Branch, Tapah Campus, Universiti Teknologi MARA, Negeri Sembilan Branch, Seremban 3 Campus, Universiti Teknologi MARA, Shah Alam, Universiti Teknologi MARA, Cawangan Selangor, Kampus Dengkil, and Universiti Kuala Lumpur Malaysian Institute of Marine Engineering Technology who provided support, insights, and resources that contributed to the successful completion of this work.

Fund

This study was not supported by any grants from funding bodies in the public, private, or not-for-profit sector

Individual Assistant

NA

AUTHOR CONTRIBUTIONS

All authors contributed equally to this work.

DECLARATION OF ORIGINALITY

The authors declare no conflict of interest to report regarding this study conducted.

GENERATIVE AI DECLARATIONS

The authors claim that artificially intelligent-assisted technologies in the form of generative AI were not used to generate content, ideas, or theories. We have just utilised AI to enhance readability and refine the language. This was used with extreme human control and oversight. The authors take full responsibility for reviewing and approving the content.

REFERENCES

[1] Zainuddin N, Ibrahim ZB, Zawawi ISM. Diagonal block method for stiff van der Pol equation. IAENG International Journal of Applied Mathematics. 2023;53(1):1-8.

[2] Soomro H, Zainuddin N, Daud H, Sunday J, Jamaludin N, Abdullah A, Kadir EA. Variable step block hybrid method for stiff chemical kinetics problems. Applied Sciences. 2022;12(9):4484.

[3] Auzinger W, Jawecki T, Koch O, Pukach P, Stolyarchuk R, Weinmüller E. Some aspects on numerical stability of evolution equations of stiff type; use of computer algebra. In: 2021 IEEE XVIIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). IEEE; 2021. p. 180-3.

[4] Faskhutdinova RI, Faskhutdinov AG, Enikeeva LV, Gubaydullin IM. Study of stiff differential equations of mathematical description of isomerization of pentane-hexane cut process. In: Journal of Physics: Conference Series. Vol. 2131, no. 2. IOP Publishing; 2021. p. 022003

[5] Mehta R, Malhotra S, Pandit D, Sahni M. Solution of 1st order stiff ordinary differential equations using feed forward neural network and Bayesian regularization algorithm. Statistics. 2022;10(2):366-77

[6] Butcher JC. Numerical methods for ordinary differential equations. 3rd ed. Hoboken, NJ: John Wiley and Sons; 2016.

[7] Lambert JD. Numerical methods for ordinary differential systems: the initial value problem. Hoboken, NJ: John Wiley and Sons; 1991.

[8] Wanner G, Hairer E. Solving ordinary differential equations II. Berlin: Springer; 1996. p. 2.

[9] Lambert JD. Computational methods in ordinary differential equations. Hoboken, NJ: John Wiley and Sons; 1973. pp. 22–23, 231–233.

[10] Curtiss CF, Hirschfelder JO. Integration of stiff equations. Proceedings of the National Academy of Sciences of the United States of America. 1952;38:235-43

[11] Rosser JB. A Runge–Kutta for all seasons. SIAM Review. 1967 Jul 1;9(3):417–52.

[12] Ibrahim ZB, Othman KI, Suleiman M. Implicit r-point block backward differentiation formula for solving first-order stiff ODEs. Applied Mathematics and Computation. 2007 Jan 1;186(1):558–65.

[13] Adoghe LO, Omole EO, Fadugba SE. Third derivative method for solving stiff system of ordinary differential equations. International Journal of Mathematics in Operational Research. 2022;23(3):412-25.

[14] Husin NM, Zawawi ISM, Zainuddin N, Ibrahim ZB. Accuracy improvement of block backward differentiation formulas for solving stiff ordinary differential equations using modified versions of Euler's method. Mathematics and Statistics. 2022;10(5):942-55

[15] Musa H, Alhassan B. Super class of implicit extended backward differentiation formulae for the numerical integration of stiff initial value problems. Computational Algorithms and Numerical Dimensions. 2025;4(1):18-33.

[16] Rasid NA, Ibrahim ZB, Majid ZA, Ismail F, Ismail A. An efficient direct diagonal hybrid block method for stiff second order differential equations. Proceedings of Advanced Maritime Technologies and Applications. 2022 Jan;166:147–56

[17] Jaafar BA, Zawawi ISM. An advanced numerical approach for solving stiff initial value problems using a self-starting two-stage composite block scheme. Computational Methods for Differential Equations. 2025;13(1):

[18] Soomro H, Zainuddin N, Daud H, Sunday J, Jamaludin N, Abdullah A, Mulono A, Kadir EA. 3-point block backward differentiation formula with an off-step point for the solutions of stiff chemical reaction problems. Journal of Mathematical Chemistry. 2023 Jan;61(1):75–97.

[19] Ijam MH, Aksah SJ, Rasedee AFN, Rasid AN, Abdulsalam A, Aris MNH, Hazimi F. Numerical solutions of stiff chemical reaction problems using hybrid block backward differentiation formula. Journal of Advanced Research in Numerical Heat Transfer. 2024 Jan;25(1):100–1

[20] Alexander R. Diagonally implicit Runge–Kutta methods for stiff ODEs. SIAM Journal on Numerical Analysis. 1977 Dec 1;14(6):1006–21

[21] Zawawi ISM, Ibrahim ZB, Ismail F, Majid ZA. Diagonally implicit block backward differentiation formulas for solving ordinary differential equations. International Journal of Mathematics and Mathematical Sciences. 2012 Jan 1;2012(1):767328.

[22] Ibrahim ZB, Ijam HM, Aksah SJ, Rasid NA. Enhancing accuracy and efficiency in stiff ODE integration using variable step diagonal BBDF approaches. Malaysian Journal of Fundamental and Applied Sciences. 2024 May 1;20(5):1083–100.

[23] Ijam HM, Ibrahim ZB, Zawawi ISM. Stiffly stable diagonally implicit block backward differentiation formula with adaptive step size strategy for stiff ordinary differential equations. MATEMATIKA. 2024 Jan 1;40(1):27–47

[24] Rasid NA, Ibrahim ZB, Majid ZA, Ismail F. Formulation of a new implicit method for group implicit BBDF in solving related stiff ordinary differential equations. Mathematics and Statistics. 2021 Mar 1;9(2):144–50.

[25] Ismail F, Ken YL, Othman M. Explicit and implicit 3-point block methods for solving special second order ordinary differential equations directly. International Journal of Mathematical Analysis. 2009;3(5):239-54.

[26] Henrici P. Discrete variable methods in ODEs. New York: John Wiley & Sons; 1962.

[27] Hall G, Watt JM. Modern numerical methods for ordinary differential equations. Oxford: Clarendon Press; 1976.

[28] Dahlquist GG. A special stability problem for linear multistep methods. BIT Numerical Mathematics. 1963;3(1):27–43

[29] Burden RL, Faires JD. *Numerical analysis*. 7th ed. Pacific Grove (CA): Brooks/Cole; 2001.

[30] Othman KI, Ibrahim ZB, Suleiman M, Majid Z. Automatic intervalwise block partitioning using Adams-type method and backward differentiation formula for solving ODEs. *Applied Mathematics and Computation*. 2007 Feb 1;188(2):1642–6.