



Article

Diagonally Implicit Block Backward Differentiation Formula with Optimal Stability Properties for Stiff Ordinary Differential Equations

Hazizah Mohd Ijam ¹  and Zarina Bibi Ibrahim ^{1,2,*} 

¹ Institute for Mathematical Research, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia; hazizahmohdijam@gmail.com

² Department of Mathematics, Faculty of Science, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia

* Correspondence: zarinabb@upm.edu.my; Tel.: +603-8946-6861

Received: 20 August 2019; Accepted: 26 September 2019; Published: 1 November 2019



Abstract: This paper aims to select the best value of the parameter ρ from a general set of linear multistep formulae which have the potential for efficient implementation. The ρ -Diagonally Implicit Block Backward Differentiation Formula (ρ -DIBBDF) was proposed to approximate the solution for stiff Ordinary Differential Equations (ODEs) to achieve the research objective. The selection of ρ for optimal stability properties in terms of zero stability, absolute stability, error constant and convergence are discussed. In the diagonally implicit formula that uses a lower triangular matrix with identical diagonal entries, allowing a maximum of one lower-upper (LU) decomposition per integration stage to be performed will result in substantial computing benefits to the user. The numerical results and plots of accuracy indicate that the ρ -DIBBDF method performs better than the existing fully and diagonally Block Backward Differentiation Formula (BBDF) methods.

Keywords: optimal stability properties; diagonally implicit; block backward differentiation formula; stiff ODEs

1. Introduction

In many mathematical models of real-world problems, Initial Value Problems (IVPs) involving systems of Ordinary Differential Equations (ODEs) exhibit a phenomenon called stiffness. This phenomenon frequently arises in the study of vibrations, chemical engineering, electrical circuits and control theory. Throughout the history of the development of numerical methods for stiff ODEs, considerable attention has been given to creating a robust algorithm with optimal stability properties. This has led us to design an efficient method that possesses a small error constant, zero stability and a large stability region. Such approaches have been taken by numerous researchers, such as [1–7].

In this article, we consider the IVPs for first order ODEs of the form

$$y' = f(x, y), \quad (1)$$

with $y(a) = y_0$ in the interval of $a \leq x \leq b$. The system (1) is said to be linear with constant coefficients if $f(x, y) = Ay + \Phi(x)$, where A is an $m \times m$ constant matrix, while y , f and $\Phi(x)$ are m -dimensional vectors. The matrix A has distinct eigenvalues, λ_i and corresponding eigenvectors, c_i , where $i = 1, 2, \dots, m$. The system of ODEs has a general solution in the form of

$$y(x) = \sum_{i=0}^m k_i e^{\lambda_i x} c_i + \Psi(x).$$

According to the definition provided in [8], (1) is considered stiff if the eigenvalues λ_i of $\frac{\partial f}{\partial y}$ satisfy the following conditions:

- (i) $Re(\lambda_i) < 0$ and
- (ii) $\max_i |Re(\lambda_i)| \gg \min_i |Re(\lambda_i)|$ where the ratio $\frac{\max_i |Re(\lambda_i)|}{\min_i |Re(\lambda_i)|}$ is called the stiffness ratio or stiffness index.

For a non-linear system, (1) is stiff in an interval I of x if, for $x \in I$, the eigenvalues $\lambda_i(x)$ satisfy (i) and (ii) above.

According to [8], stiffness requires solving the implicit equations by using Newton iteration, which in turn demands an evaluation of the Jacobian at each step. These requirements will increase the computational time and therefore, is not cost-efficient for the users. Because of the high cost of evaluating the stages in a fully implicit method, [9] reported that many researchers have opted to reduce it to a diagonally implicit method. Prior works along this line were discussed by [10–14].

In practical applications involving stiff ODEs, the Diagonally Implicit Runge–Kutta (DIRK) class is the most often used among the Implicit Runge–Kutta (IRK) methods. Figure 1 illustrates the fully implicit and diagonally implicit classes of Runge–Kutta (RK) methods, which can be defined as a 3×3 matrix. As stated in [15], RK methods are characterized by excellent stability properties that make them useful for solving stiff ODEs systems. However, for a Fully Implicit Runge–Kutta (FIRK) method, a system of $n \times r$ non-linear equations must be solved in each of its integration stage, where n is the dimension of the problem and r is the number of stages, as described by [16]. Thus, the authors of [17,18] proposed a DIRK method that uses a lower triangular matrix, A with $a_{ij} = 0$ for $i < j$. It implies that the DIRK technique needs to solve a series of r -implicit structures for each n instead of an $n \times r$ system as in the FIRK method. This structure permits solving each stage of the system separately instead of solving all the stages simultaneously. Another common requirement, as stated in [19], is for the non-zero diagonal entries of A to be identical, allowing for a maximum of one lower-upper (LU) decomposition per integration stage.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (a) \text{ FIRK}$$

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (b) \text{ DIRK}$$

Figure 1. Structure of the A -matrix for fully and diagonally implicit subclasses of the RK methods, given in [16].

The well-known Backward Differentiation Formula (BDF) has been the technique of choice for the numerical solution of stiff differential equations for many years. The classical BDF method approximates the solution for y_{n+1} at x_{n+1} point in every step. Ibrahim et al. in [20] introduced the Block Backward Differentiation Formula (BBDF) method to reduce the number of integration steps and the computational time of existing numerical integrator while maintaining the accuracy. Many attempts have been made to implement the BBDF method in solving stiff problems because it has been proven as more accurate and efficient than the non-block method (see [20–22]) and existing solvers (see [23,24]).

Motivated by the fact that the existing works carried out based on RK and BBDF are suitable for solving stiff ODEs, we aim to formulate an efficient BBDF method in a diagonally implicit form that is expected to be faster than the fully implicit methods in the existing literature.

This paper is organized as follows. The derivation of the method is presented in Section 2. In Section 3, we discuss the stability properties of the method encompassing zero stability, absolute stability, order of the method and convergence. Next, the implementation of the derived method using

Newton iteration is discussed in Section 4, followed by the numerical results for the proposed and existing methods in Section 5. Finally, the conclusions are provided in Section 6.

2. The ρ -Diagonally Implicit Block Backward Differentiation Formula

Various methods can be used to compute the approximate solution of $y(x)$. One of these methods is the general linear k -step method in the form of

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j}. \quad (2)$$

If $\beta_0 = \beta_1 = \dots = \beta_{k-1} = 0$ and $\beta_k \neq 0$, (2) becomes a BDF as outlined by Gear in [25] in the form of

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \beta_k f_{n+k}. \quad (3)$$

Vijitha-Kumara in [26] modified the formula in (3) by taking an arbitrary $\beta_{k-1} \neq 0$, introducing the free parameter ρ and formulating the non-block Fixed Step Formula (FSF) of

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \beta_k (f_{n+k} - \rho f_{n+k-1}), \quad (4)$$

where $\beta_{k-1} = \rho \beta_k$.

The definition of the k -step 2-point block linear multistep method (LMM) of BBDF as given by Ibrahim et al. in [20] is in the form of

$$\sum_{j=0}^k A_{j,i} y_{n+j} = h \sum_{j=0}^k B_{j,i} f_{n+j}, \quad (5)$$

where $k = i = 1, 2$ for y_{n+1} and y_{n+2} respectively; $A_{k,i} = 1$; $A_{j,i}, B_{j,i}$ are $r \times r$ coefficient matrices; $f_{n+j} = f(x_{n+j}, y_{n+j})$ and $h =$ step size used. This constant step size method is implemented by approximating y_{n+1}, \dots, y_{n+j} concurrently in a block at the time discretization points of x_{n+1}, \dots, x_{n+j} .

In this section, we will derive the ρ -Diagonally Implicit Block Backward Differentiation Formula (ρ -DIBBDF) based on the derivation of BBDF by Ibrahim et al. in [20] and FSF by Vijitha-Kumara in [26]. FSF is a non-block method, where the computation proceeds to an approximation of y_{n+1} at the x_{n+1} one step at a time. To increase the efficiency of the classical approach, we proposed a method that constructed in a block, where the solutions of y_{n+1} and y_{n+2} were approximated concurrently in a block by using three back values at the points x_n, x_{n-1} and x_{n-2} of the previous block. The development of our method involves the hybrid-like process of implementing the FSF and BBDF, which produce as many A -stable formulae as possible. Suleiman et al. [22] extended the formula in (5), implemented the strategy in [26] by adding extra future points and proposed the 2-point Superclass of BBDF (2SBBDf). The BBDF and 2SBBDf are formulated in a fully implicit manner. We are motivated to develop the ρ -DIBBDF to enhance the efficiency of the fully implicit BBDF method by proposing a diagonally implicit method that requires fewer computations of the differentiation coefficients, thus minimizing the cumulative error (refer to [10]). Our proposed method differs from the Diagonally Implicit 2-point BBDF (DI2BBDF) in [10], which provides a fixed formula for each point and order because our method generates a different set of formulae depending on the free parameter chosen in an attempt to achieve the optimal stability properties and accurate numerical results.

The ρ -DIBBDF method takes the general form of

$$\sum_{j=0}^{k+2} \alpha_{j-2,k} y_{n+j-2} = h\beta_{k,k} (f_{n+k} - \rho f_{n+k-1}), \quad k = 1, 2 \tag{6}$$

where $\beta_{k-1,k} = \rho\beta_{k,k}$. The linear difference operator L_i associated with the formula in (6), is defined as

$$L[y(x_n); h] = \sum_{j=0}^{k+2} \alpha_{j-2,k} y_{n+j-2} - h\beta_{j,k} (y'_{n+k} - \rho y'_{n+k-1}). \tag{7}$$

Expanding y_{n+j-2} and its derivative by using the Taylor series method and collecting the common terms of the derivative y in (7) gives

$$L[y(x_n); h] = C_0 y(x) + C_1 h y'(x) + \dots + C_q h^q y^{(q)}(x). \tag{8}$$

The general form of the constant C_q is given by

$$\begin{aligned} C_0 &= \sum_{j=0}^{k+2} \alpha_{j-2,k}, \\ C_1 &= \sum_{j=0}^{k+2} \frac{(j-2)^1}{1!} \alpha_{j-2,k} - \frac{k^{(0)}}{0!} \beta_{j,k} + \rho \beta_{j,k}, \\ &\vdots \\ C_q &= \sum_{j=0}^{k+2} \frac{(j-2)^q}{q!} \alpha_{j-2,k} - \frac{k^{(q-1)}}{(q-1)!} \beta_{j,k} + \frac{(k-1)^{(q-1)}}{(q-1)!} \rho \beta_{j,k}, \quad q = 2, 3, \dots \end{aligned} \tag{9}$$

The values of $k = 1, 2$ in (6) indicate the first and second points, respectively. By setting $\alpha_{1,1} = 1, \alpha_{2,2} = 1, \alpha_{0,2} = 0$ and solving (9) simultaneously, we obtain the following coefficients of $\alpha_{j-2,k}$ and $\beta_{j,k}$ in terms of ρ as listed in Table 1:

Table 1. List of coefficients for ρ -DIBBDF.

k	$\alpha_{-2,k}$	$\alpha_{-1,k}$	$\alpha_{0,k}$	$\alpha_{1,k}$	$\alpha_{2,k}$	$\beta_{0,k}$	$\beta_{1,k}$	$\beta_{2,k}$
1	$\frac{\rho + 2}{2\rho - 11}$	$\frac{-3(2\rho + 3)}{2\rho - 11}$	$\frac{3(\rho + 6)}{2\rho - 11}$	1	0	$\frac{-6\rho}{2\rho - 11}$	$\frac{-6}{2\rho - 11}$	0
2	$\frac{2\rho + 3}{6\rho - 19}$	$\frac{-2(3\rho + 4)}{6\rho - 19}$	0	$\frac{-2(\rho - 12)}{6\rho - 19}$	1	0	$\frac{-12\rho}{6\rho - 19}$	$\frac{-12}{6\rho - 19}$

Substituting the coefficients obtained in Table 1 into (6) gives the following general corrector formula for the 2-point ρ -DIBBDF:

$$\begin{aligned} y_{n+1} &= -\frac{\rho + 2}{2\rho - 11} y_{n-2} + \frac{3(2\rho + 3)}{2\rho - 11} y_{n-1} - \frac{3(\rho + 6)}{2\rho - 11} y_n + \frac{6\rho}{2\rho - 11} h f_n - \frac{6}{2\rho - 11} h f_{n+1}, \\ y_{n+2} &= -\frac{2\rho + 3}{6\rho - 19} y_{n-2} + \frac{2(3\rho + 4)}{6\rho - 19} y_{n-1} + \frac{2(\rho - 12)}{6\rho - 19} y_{n+1} + \frac{12\rho}{6\rho - 19} h f_{n+1} - \frac{12}{6\rho - 19} h f_{n+2}. \end{aligned} \tag{10}$$

The matrix form of the corrector formula in (10) is therefore represented by

$$\begin{aligned} \begin{bmatrix} 1 & 0 \\ -\frac{2(\rho-12)}{6\rho-19} & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} &= \begin{bmatrix} 0 & -\frac{\rho+2}{2\rho-11} \\ 0 & -\frac{2\rho+3}{6\rho-19} \end{bmatrix} \begin{bmatrix} y_{n-3} \\ y_{n-2} \end{bmatrix} + \begin{bmatrix} \frac{3(2\rho+3)}{2\rho-11} & -\frac{3(\rho+6)}{2\rho-11} \\ \frac{2(3\rho+4)}{6\rho-19} & 0 \end{bmatrix} \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} \\ &+ h \begin{bmatrix} 0 & \frac{6\rho}{2\rho-11} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix} + h \begin{bmatrix} -\frac{6}{2\rho-11} & 0 \\ \frac{12\rho}{6\rho-19} & -\frac{12}{6\rho-19} \end{bmatrix} \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}. \end{aligned} \quad (11)$$

3. Selection of Parameter ρ

To determine whether the numerical method can provide acceptable results, we need to investigate the stability of the method. In this section, the practical efficiency of the numerical method concerning three essential concepts, namely zero stability, absolute stability and convergence will be studied. According to Dahlquist in [27], a potentially useful numerical method for solving stiff ODE systems must have good accuracy and a reasonably large region of absolute stability. In designing an efficient method, authors in [2,3,7] considered developing methods with smaller error constants. The development of our method closely follows the works of [22,26] but in the DI scheme. To overcome the setback in choosing the best ρ as reported in [28], we will choose a better value for ρ with optimal stability properties that will give better accuracy to the approximate solution. The parameter ρ is restricted to $(-1, 1)$ so that the underlying ρ -DIBBDF in (6) satisfies the necessary condition for stiff stability. The relevant proof for y_{n+1} is provided in this section concerning the theorem due to [29] (refer to [26]). The proof of the second point is straightforward by adopting an approach similar to that of the first point.

To begin with, the associate ϱ and σ polynomials of (6) are given as follows

$$\begin{aligned} \varrho(\tilde{\zeta}) &= \tilde{\zeta}^3 + \alpha_{0,k}\tilde{\zeta}^2 + \alpha_{-1,k}\tilde{\zeta} + \alpha_{-2,k}, \\ \sigma(\tilde{\zeta}) &= \beta_{k,k}(\tilde{\zeta}^3 - \rho\tilde{\zeta}^2). \end{aligned}$$

Then, by using the following polynomials of $r(z)$ and $s(z)$,

$$\begin{aligned} r(z) &= \left(\frac{1-z}{2}\right)^3 \varrho\left(\frac{1+z}{1-z}\right), \\ s(z) &= \left(\frac{1-z}{2}\right)^3 \sigma\left(\frac{1+z}{1-z}\right), \end{aligned}$$

we define $P(z, \mu) = r(z) - \mu s(z)$ where $\mu = h\lambda$. $P(z, \mu)$ can be simplified as

$$-(2\rho - 11)P(z, \mu) = a_3z^3 + a_2z^2 + a_1z + a_0, \quad (12)$$

where

$$\begin{aligned} a_0 &= 6(\rho - 1)\mu, \\ a_1 &= -12(\rho - 1) + 6(\rho - 3)\mu, \\ a_2 &= -12(\rho - 3) - 6(\rho + 3)\mu, \\ a_3 &= 8(\rho + 5) - 6(\rho + 1)\mu. \end{aligned} \quad (13)$$

The following lemma will be used in the next theorem.

Lemma 1. Let $p(x) = a_3x^3 + a_2x^2 + a_1x + a_0$, where a_i are real numbers and $a_3 \neq 0$. Then $p(x)$ is a Hurwitz polynomial if and only if the conditions (i) and (ii) both hold.

(i) a_i are either all positive or negative,

(ii) $a_1a_2 - a_0a_3 > 0$.

Proof. See [30]. \square

According to [26], (6) is A_0 -stable if and only if its corresponding $P(z, \mu)$ is a Hurwitz polynomial for all $\mu < 0$.

Theorem 1. *The method is A_0 -stable for all ρ in $(-1, 1)$.*

Proof. Clearly from (13), $a_i > 0$, where $i = 0, 1, 2, 3$ for all $\mu < 0$ and $\rho \in (-1, 1)$. For (ii), $a_1a_2 - a_0a_3$ can be written as $144(\rho - 1)(\rho - 3) - 48(\rho^2 - 8\rho + 13)\mu + 288\mu^2$. Thus, $a_1a_2 - a_0a_3 > 0$ for all $\mu < 0$ and $\rho \in (-1, 1)$. By Lemma 1, $-(2\rho - 11)P(z, \mu)$ in (12) is a Hurwitz polynomial and so is $P(z, \mu)$ for all $\mu < 0$ and $\rho \in (-1, 1)$. \square

The following theorem due to [31] gives necessary and sufficient conditions for stiff stability.

Theorem 2. *The conditions (i)–(iv) are necessary and sufficient for a convergent method to be stiffly stable.*

- (i) *The method is A_0 -stable,*
- (ii) *The modulus of any root of the polynomial $\frac{\varrho(\xi)}{\xi-1}$ is less than 1,*
- (iii) *The roots of $\sigma(\xi)$ of modulus 1 are simple,*
- (iv) *If ξ_0 is a root of $\sigma(\xi)$ with $|\xi_0| = 1$, then $\frac{\varrho(\xi)}{\xi\sigma'(\xi)}$ at $\xi = \xi_0$ is real and positive.*

The following lemma will be used to prove the next theorem.

Lemma 2. *Let $p(x) = a_2x^2 + a_1x + a_0$, where $a_0, a_1, a_2 \neq 0$ and a_0, a_1, a_2 are real. Then $p(x)$ is a Schur polynomial if and only if the conditions of (i) and (ii) are met.*

- (i) $|a_0| < |a_2|$,
- (ii) $|a_1| < |a_2 + a_0|$.

Proof. See [26]. \square

Theorem 3. *The method is strongly stable for $\rho \in (-1, 1)$.*

Proof. It suffices to show that $\frac{\varrho(\xi)}{\xi-1}$ is a Schur polynomial for $\rho \in (-1, 1)$. From Table 1, $\alpha_{-2,1} + \alpha_{-1,1} + \alpha_{0,1} + 1 = 0$, $\varrho(\xi)$ can be simplified as $\varrho(\xi) = (\xi - 1)(a_2x^2 + a_1x + a_0)$, where $a_0 = -\alpha_{-2,1} = -\frac{\rho+2}{2\rho-11}$, $a_1 = -(\alpha_{-2,1} + \alpha_{-1,1}) = -\left(\frac{-5\rho-7}{2\rho-11}\right)$ and $a_2 = 1$. Since $-\frac{\rho+2}{2\rho-11} \leq \frac{1}{3}$ for $-1 < \rho < 1$, we have $a_0 < a_2$. Now, $a_2 + a_0 = \frac{\rho-13}{2\rho-11}$ and $|a_2 + a_0| - |a_1| = \frac{6(\rho-1)}{2\rho-11} > 0$ for $-1 < \rho < 1$. Thus, by Lemma 2, $\frac{\varrho(\xi)}{\xi-1}$ is a Schur polynomial. \square

Theorem 4. *The method is stiffly stable for $\rho \in (-1, 1)$.*

Proof. $\sigma(\xi) = \beta_{1,1}(\xi^3 - \rho\xi^2)$ where $\beta_{1,1} = \frac{-6}{2\rho-11}$. The roots of $\sigma(\xi)$ are $0, \rho$ and it has simple root of modulus 1 when $\rho = -1$. Now Theorem 2 together with Theorem 1 and Theorem 3 imply that the method is stiffly stable for all $\rho \in (-1, 1)$. \square

Corollary 1. *The method is $A(\alpha)$ -stable for all $\rho \in (-1, 1)$.*

Proof. Stiff stability implies $A(\alpha)$ -stability (refer to [31]). \square

For the order 3 method, the intensive work done by [26] shows that $\rho = -0.75$ will produce accurate numerical results with optimal stability properties. The stability analysis of the proposed method for $\rho = -0.75$ will be presented in the following subsections.

3.1. Zero Stability

Zero stability is one of the important forms of stability for the numerical solution of IVPs. It is defined by [32] as:

Definition 1. The method in (10) is said to have zero stability if no root of its characteristic polynomial has a modulus higher than one and if any root with a modulus of one is simple.

Consider the scalar test equation $y' = \lambda y$. Substitute $h\lambda = \bar{h}$ into (11) and rewrite it in the matrix form to get

$$\begin{bmatrix} 1 + \frac{6}{2\rho-11}\bar{h} & 0 \\ -\frac{2(\rho-12)}{6\rho-19} - \frac{12\rho}{6\rho-19}\bar{h} & 1 + \frac{12}{6\rho-19}\bar{h} \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} = \begin{bmatrix} \frac{3(2\rho+3)}{2\rho-11} & -\frac{3(\rho+6)}{2\rho-11} + \frac{6\rho}{2\rho-11}\bar{h} \\ \frac{2(3\rho+4)}{6\rho-19} & 0 \end{bmatrix} \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} + \begin{bmatrix} 0 & -\frac{\rho+2}{2\rho-11} \\ 0 & -\frac{2\rho+3}{6\rho-19} \end{bmatrix} \begin{bmatrix} y_{n-3} \\ y_{n-2} \end{bmatrix} \quad (14)$$

which is equivalent to $A_0 Y_m = B_0 Y_{m-1} + C_0 Y_{m-2}$. By inserting the coefficients of A_0 , B_0 and C_0 in (14) into the determinant formula viz. $|A_0 t^2 - B_0 t - C_0|$, we obtain the following stability polynomial in terms of ρ :

$$\begin{aligned} R(t, \bar{h}) = & \left[\frac{-1}{(2\rho-11)(6\rho-19)} \right] 72\bar{h}^2 \rho^2 t^3 - 72\bar{h}^2 t^4 - 24\bar{h} \rho^2 t^3 - 60\bar{h} \rho t^4 - 12\rho^2 t^4 + 24\bar{h} \rho^2 t^2 - 288\bar{h} \rho t^3 \\ & + 246\bar{h} t^4 + 30\rho^2 t^3 + 104\rho t^4 + 12\bar{h} \rho t^2 + 108\bar{h} t^3 - 24\rho^2 t^2 - 24\rho t^3 - 209t^4 - 18\bar{h} t^2 + 6\rho^2 t \\ & - 96\rho t^2 + 261t^3 + 16\rho t - 63t^2 + 11t = 0. \end{aligned} \quad (15)$$

By plugging $\bar{h} = 0$ and solving (15) with respect to t , we obtain the roots, t as listed in (16).

$$\begin{aligned} t = & 0, 1, \frac{1}{12\rho^2 - 104\rho + 209} \left(9\rho^2 + \sqrt{9\rho^4 + 1152\rho^3 + 2346\rho^2 - 120\rho - 1623 + 40\rho + 26} \right), \\ & - \frac{1}{12\rho^2 - 104\rho + 209} \left(-9\rho^2 + \sqrt{9\rho^4 + 1152\rho^3 + 2346\rho^2 - 120\rho - 1623 - 40\rho - 26} \right). \end{aligned} \quad (16)$$

By setting $\rho = -0.75$, we obtain the roots, $t = 0, 1, 0.003617 \pm 0.08982i$. Since the roots of the stability polynomial for $\rho = -0.75$ satisfy Definition 1, we conclude that the method is said to be zero stable.

3.2. Absolute Stability

According to [32], if the employed method has a region of absolute stability that includes the entire negative-plane, then there will be no constraint on the step length imposed by the stability.

Definition 2. A numerical method is said to be A -stable if $\mathfrak{R}_A \supseteq \{h\lambda \mid \text{Re}(h\lambda) < 0\}$.

However, the A -stability requirement places a severe restriction on selecting appropriate LMMs. This restriction is known as Dahlquist's second barrier, which states that the order of an A -stable LMM must be ≤ 2 (see [27]). This demanding requirement motivates the following definition by [33]:

Definition 3. A numerical method is said to be $A(\alpha)$ -stable, $\alpha \in (0, \pi/2)$ if $\mathfrak{R}_A \supseteq \{h\lambda \mid -\alpha < \pi - \arg(h\lambda) < \alpha\}$ as shown in Figure 2.

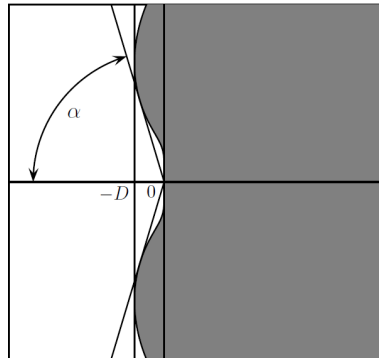


Figure 2. $A(\alpha)$ and stiff stability, as featured in [34].

The boundary of the stability region is determined by putting $t = e^{i\theta}$ in (15) where $0 \leq \theta \leq 2\pi$ for which $|t| < 1$. The stable region is located outside the boundary of the solid line and the unstable region is within the enclosed area. The absolute stability region of our method for $\rho = -0.75$ is plotted in a complex $h\lambda$ plane and shown below.

Referring to Figure 3, we can observe that almost the whole left-plane of the circle lies within a stable region and hence, based on Definition 3, the method is considered $A(\alpha)$ -stable.

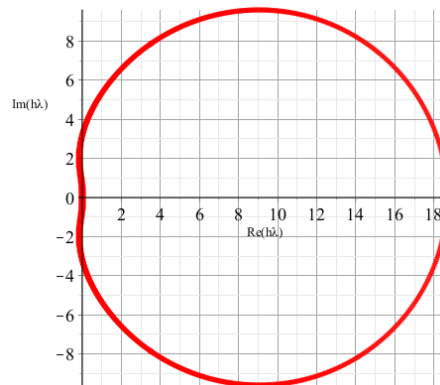


Figure 3. Stability region of ρ -DIBBDF for $\rho = -0.75$.

3.3. Order of the Method and Error Constant

The definition of the order of the method given by [8] is provided in Definition 4, as follows:

Definition 4. The linear multistep method (5) and the associated difference operator, L_i as defined by (6), are said to be of order p if in (8), $C_0 = C_1 = \dots = C_p = 0$ and $C_{p+1} \neq 0$, where C_{p+1} is the error constant.

The error constant of the method can be obtained by substituting the corresponding values of $\alpha_{j,k}$ and $\beta_{j,k}$ in Table 1 into (9), which gives

$$\begin{aligned}
C_0 &= \sum_{j=0}^4 \alpha_{j,k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_1 &= \sum_{j=0}^4 \frac{j}{1!} \alpha_{j,k} - \beta_{j,k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_2 &= \sum_{j=0}^4 \frac{j^2}{2!} \alpha_{j,k} - \frac{j}{1!} \beta_{j,k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_3 &= \sum_{j=0}^4 \frac{j^3}{3!} \alpha_{j,k} - \frac{j^2}{2!} \beta_{j,k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_4 &= \sum_{j=0}^4 \frac{j^4}{4!} \alpha_{j,k} - \frac{j^3}{3!} \beta_{j,k} = \begin{bmatrix} \frac{1}{2} \frac{\rho+3}{2\rho-11} \\ \frac{3(\rho+2)}{6\rho-19} \end{bmatrix} = \begin{bmatrix} \frac{-9}{100} \\ \frac{-15}{94} \end{bmatrix}.
\end{aligned} \tag{17}$$

Since $C_4 \neq 0$ in (17), following Definition 4, we conclude that the derived method has the order 3 with $\frac{-9}{100}$ and $\frac{-15}{94}$ as the error constants of y_{n+1} and y_{n+2} respectively.

3.4. Convergence

An essential property of an acceptable LMM is that the solution generated by the method converges to an exact solution as the step size approaches zero. A theorem provided by [35] states that a method of an LMM class in (6) is convergent if and only if, it is both consistent and zero stable. The proof for this theorem can be found in [35]. For this theorem to be satisfied, the following consistency conditions given by [8] must be fulfilled:

- (a) $\sum_{j=0}^{k+2} \alpha_{j-2,k} = 0$,
- (b) $\sum_{j=0}^{k+2} j\alpha_{j-2,k} = \sum_{j=0}^{k+2} \beta_{j,k}$.

By applying the above consistency conditions for the respective coefficients of y_{n+1} and y_{n+2} in Table 1, we obtained:

$$\begin{aligned}
\text{a) } \sum_{j=0}^{k+2} \alpha_{j-2,k} &= \begin{bmatrix} \frac{\rho+2}{2\rho-11} \\ \frac{2\rho+3}{6\rho-19} \end{bmatrix} + \begin{bmatrix} \frac{-3(2\rho+3)}{2\rho-11} \\ \frac{-2(3\rho+4)}{6\rho-19} \end{bmatrix} + \begin{bmatrix} \frac{3(\rho+6)}{2\rho-11} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ \frac{-2(\rho-12)}{6\rho-19} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
\text{b) } \sum_{j=0}^{k+2} j\alpha_{j-2,k} &= 0 \begin{bmatrix} \frac{\rho+2}{2\rho-11} \\ \frac{2\rho+3}{6\rho-19} \end{bmatrix} + 1 \begin{bmatrix} \frac{-3(2\rho+3)}{2\rho-11} \\ \frac{-2(3\rho+4)}{6\rho-19} \end{bmatrix} + 2 \begin{bmatrix} \frac{3(\rho+6)}{2\rho-11} \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 1 \\ \frac{-2(\rho-12)}{6\rho-19} \end{bmatrix} + 4 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{6\rho-6}{2\rho-11} \\ \frac{12\rho-12}{6\rho-19} \end{bmatrix} = \begin{bmatrix} \frac{21}{25} \\ \frac{42}{47} \end{bmatrix} \\
&= \sum_{j=0}^{k+2} \beta_{j,k}.
\end{aligned}$$

Since the method for $\rho = -0.75$ is zero stable as stated in Section 3.1 and both consistency conditions are successfully met, we conclude that the derived method converged.

The α , D in $A(\alpha)$ -stability and error constants (EC) of the underlying ρ -DIBBDF in (7) for $\rho = -0.75, -0.60, 0.50$ and 0.95 are given in Table 2. We choose these ρ values to compare with $\rho = -0.75$ to ensure a higher accuracy for the numerical results by selecting the best parameter ($\rho = -0.75$), as proven for the non-block FSF in [26].

Table 2. Comparison of stability analysis for ρ -DIBBDF, BBDF and DI2BBDF.

ρ or Method	−0.75	−0.60	0.50	0.95	BBDF	DI2BBDF
α	85.657°	86.084°	88.352°	90°	90.0°	83.647°
D	−0.156	−0.115	−0.016	0	0	−0.226
$ EC y_{n+1}$	0.0900	0.0984	0.1750	0.2170	0.1667	0.1364
$ EC y_{n+2}$	0.1596	0.1858	0.4688	0.6654	0.1364	0

4. Implementation of the Method

In this section, the approximation of y_{n+1} and y_{n+2} values in (14) will be implemented using the Newton's iteration. Formula in (10) can be represented in the following form:

$$\begin{aligned} y_{n+1} &= \alpha_1 y_{n+2} + \beta_1 h f_n + \beta_2 h f_{n+1} + \eta_1, \\ y_{n+2} &= \alpha_2 y_{n+1} + \beta_3 h f_{n+1} + \beta_4 h f_{n+2} + \eta_2, \end{aligned} \quad (18)$$

where η_1 and η_2 are the back values. Equation (18) in the form of the matrix-vector is equivalent to

$$(I - A)Y = h(B_1 F_1 + B_2 F_2) + \zeta,$$

with

$$\begin{aligned} I &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 0 & \alpha_1 \\ \alpha_2 & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 & \beta_1 \\ 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} \beta_2 & 0 \\ \beta_3 & \beta_4 \end{bmatrix}, \\ Y &= \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix}, F_1 = \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}, F_2 = \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}, \zeta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}. \end{aligned} \quad (19)$$

Let

$$\hat{F} = (I - A)Y - h(B_1 F_1 + B_2 F_2) - \zeta = 0. \quad (20)$$

By applying Newton's iteration to the system in (20), the $(i + 1)$ th iterative value of y_{n+j} is generated and we obtain

$$y_{n+j}^{(i+1)} = y_{n+j}^{(i)} - \frac{\hat{F}(y_{n+j}^{(i)})}{\hat{F}'(y_{n+j}^{(i)})}, \quad j = 1, 2. \quad (21)$$

Equation (21) is equivalent to

$$y_{n+j}^{(i+1)} - y_{n+j}^{(i)} = - \frac{(I - A)Y_{n+j}^{(i)} - h(B_1 F_1 + B_2 F_2) - \zeta}{(I - A) - h \left[B_1 \frac{\partial F_1}{\partial Y} (Y_{n+j}^{(i)}) + B_2 \frac{\partial F_2}{\partial Y} (Y_{n+j}^{(i)}) \right]}, \quad (22)$$

where $\frac{\partial F}{\partial Y} (Y_{n+j}^{(i)})$ denotes the Jacobian matrix of F with respect to Y . By letting $E_{n+j}^{(i+1)} = y_{n+j}^{(i+1)} - y_{n+j}^{(i)}$, (22) can be expressed in the simplest form

$$E_{n+j}^{(i+1)} = \bar{P}^{-1} \bar{Q}. \quad (23)$$

It follows that

$$\bar{P} E_{n+j}^{(i+1)} = \bar{Q}, \quad (24)$$

where

$$\bar{P} = (I - A) - h \left[B_1 \frac{\partial F_1}{\partial Y} \left(Y_{n+j}^{(i)} \right) + B_2 \frac{\partial F_2}{\partial Y} \left(Y_{n+j}^{(i)} \right) \right], \bar{Q} = -(I - A) Y_{n+j}^{(i)} - h (B_1 F_1 + B_2 F_2) - \zeta. \quad (25)$$

By plugging the corresponding entries of I, A, B_1, B_2 in (19) into (25) we obtain

$$\bar{P} = \begin{bmatrix} 1 - \beta_2 h \frac{\partial f_{n+1}}{\partial y_{n+1}} & -\alpha_1 \\ -\alpha_2 - \beta_3 h \frac{\partial f_{n+1}}{\partial y_{n+1}} & 1 - \beta_4 h \frac{\partial f_{n+2}}{\partial y_{n+2}} \end{bmatrix}, \bar{Q} = \begin{bmatrix} -y_{n+1}^i + \alpha_1 y_{n+2}^i + \beta_1 h f_n^i + \beta_2 h f_{n+1}^i + \eta_1 \\ -y_{n+2}^i + \alpha_2 y_{n+1}^i + \beta_3 h f_{n+1}^i + \beta_4 h f_{n+2}^i + \eta_2 \end{bmatrix}. \quad (26)$$

Equation (21) in matrix form for formula in (10) will produce

$$\begin{bmatrix} 1 + \left(\frac{2}{\rho-3} \right) h \frac{\partial f_{n+1}}{\partial y_{n+1}} & 0 \\ -\frac{3}{4} \left(\frac{\rho-3}{\rho-2} \right) - \frac{3\rho}{2\rho-4} h \frac{\partial f_{n+1}}{\partial y_{n+1}} & 1 + \left(\frac{3}{2\rho-4} \right) h \frac{\partial f_{n+1}}{\partial y_{n+1}} \end{bmatrix} \begin{bmatrix} E_{n+1}^{(i+1)} \\ E_{n+2}^{(i+1)} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ \frac{3}{4} \left(\frac{\rho-3}{\rho-2} \right) & -1 \end{bmatrix} \begin{bmatrix} y_{n+1}^{(i)} \\ y_{n+2}^{(i)} \end{bmatrix} + h \begin{bmatrix} 0 & \frac{2\rho}{\rho-3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_{n-1}^{(i)} \\ f_n^{(i)} \end{bmatrix} + h \begin{bmatrix} -\frac{2}{\rho-3} & 0 \\ \frac{3\rho}{2\rho-4} & -\frac{3}{2\rho-4} \end{bmatrix} \begin{bmatrix} f_{n+1}^{(i)} \\ f_{n+2}^{(i)} \end{bmatrix}. \quad (27)$$

The approximate values of y_{n+1} and y_{n+2} are therefore derived from

$$y_{n+j}^{(i+1)} = y_{n+j}^{(i)} + E_{n+j}^{(i+1)}, \quad j = 1, 2. \quad (28)$$

The computation was performed in *PECE* mode in accordance with the terminology used in the LMM context. P and C indicate one predictor or corrector implementation, respectively and E indicates one evaluation of the function $y' = f(x, y)$. The predictor formula used in this *PECE* sequence are given by:

$$\begin{aligned} y_{n+1}^{(p)} &= -y_{n-1} + 2y_n, \\ y_{n+2}^{(p)} &= -2y_{n-1} + 3y_n. \end{aligned} \quad (29)$$

The approximation of y_{n+1} and y_{n+2} values will be executed simultaneously in every step. The *PECE* block method mode is described as:

$$\begin{aligned} P : y_{n+1}^{(p)} &\rightarrow E : f_{n+1}^{(p)} \rightarrow C : y_{n+1}^{(c)} \rightarrow E : f_{n+1}^{(c)}, \\ P : y_{n+2}^{(p)} &\rightarrow E : f_{n+2}^{(p)} \rightarrow C : y_{n+2}^{(c)} \rightarrow E : f_{n+2}^{(c)}. \end{aligned} \quad (30)$$

5. Numerical Results

In the interest of validating the numerical results of our method, the ρ -DIBBDF algorithm was written in C programming language on the Microsoft Visual C++ platform to obtain the approximate values. BBDF in [20] and DI2BBDF in [10] are chosen as the methods of comparison because they are of the same order as our derived method. For our method, we choose $\rho = -0.60, 0.50, 0.95$ to compare with $\rho = -0.75$. Due to space limitation, only some ρ values will be examined. A detailed description of the selection of ρ , was discussed in [26]. For the four selected test problems, the numerical results of the maximum error and execution time are given in Tables 3–6, with

$$MAXE = \underbrace{\max}_{1 \leq i \leq T} \left(\underbrace{\max}_{1 \leq i \leq N} |(y_i)_t - (y(x_i))_t| \right),$$

where T is the total number of steps, N is the number of equations, y_i and $y(x_i)$ are the approximated and exact solutions, respectively.

Test Problem 1:

$$y' = -2\pi \sin(2\pi x) - \frac{1}{10^{-3}} (y - \cos(2\pi x)), \quad y(0) = 1, \quad 0 \leq x \leq 1$$

Exact solution:

$$y(x) = \cos(2\pi x)$$

Source: [36]

Test Problem 2:

$$y' = 5e^{5x} (y - x)^2 + 1, \quad y(0) = -1, \quad 0 \leq x \leq 1$$

Exact solution:

$$y(x) = x - e^{-5x}$$

Source: [37]

Test Problem 3:

$$\begin{aligned} y_1' &= -y_2' - 10^{-5} y_1 (1 - y_1^2 - y_2^2), & y_1(0) &= 1, & 0 \leq x \leq 3 \\ y_2' &= y_1' - 3 \times 10^{-5} y_2 (1 - y_1^2 - y_2^2), & y_2(0) &= 0 \end{aligned}$$

Exact solution:

$$\begin{aligned} y_1(x) &= \cos(x) \\ y_2(x) &= \sin(x) \end{aligned}$$

Source: [36]

Test Problem 4:

$$\begin{aligned} y_1' &= -21y_1 + 19y_2 - 20y_3, & y_1(0) &= 1, & 0 \leq x \leq 10 \\ y_2' &= 19y_1 - 21y_2 + 20y_3, & y_2(0) &= 0 \\ y_3' &= 40y_1 - 40y_2 - 40y_3, & y_3(0) &= -1 \end{aligned}$$

Exact solution:

$$\begin{aligned} y_1(x) &= 0.5 \left[e^{-2x} + e^{-40x} (\cos(40x) + \sin(40x)) \right] \\ y_2(x) &= 0.5 \left[e^{-2x} - e^{-40x} (\cos(40x) + \sin(40x)) \right] \\ y_3(x) &= 2e^{-40x} \left[-\frac{1}{2} \cos(40x) + \frac{1}{2} \sin(40x) \right] \end{aligned}$$

Source: [8]

The notations used in Tables 3–6 are described as follows:

H : Step size

MAXE : Maximum error

TIME : Execution time (seconds)

ρ -DIBBDF(ρ_i) : ρ -Diagonally Implicit Block Backward Differentiation Formula (ρ value)

BBDF : Block Backward Differentiation Formula of order 3 in [20]

DI2BBDF : Diagonally Implicit 2-point BBDF of order 3 in [10]

Table 3. Numerical results for test problem 1.

H	Method	MAXE	TIME
10^{-2}	ρ -DIBBDF(−0.75)	3.61318×10^{-2}	7.90031×10^{-6}
	ρ -DIBBDF(−0.60)	3.83043×10^{-2}	8.90948×10^{-6}
	ρ -DIBBDF(0.50)	1.04695×10^{-1}	1.90458×10^{-5}
	ρ -DIBBDF(0.95)	1.70999×10^{-1}	3.15808×10^{-5}
	BBDF	7.75777×10^8	9.31533×10^{-6}
	DI2BBDF	3.49466×10^{-1}	2.37411×10^{-5}
10^{-4}	ρ -DIBBDF(−0.75)	5.14905×10^{-7}	1.55234×10^{-5}
	ρ -DIBBDF(−0.60)	5.25483×10^{-7}	2.45657×10^{-5}
	ρ -DIBBDF(0.50)	6.58550×10^{-7}	2.81742×10^{-5}
	ρ -DIBBDF(0.95)	1.18569×10^{-6}	3.92283×10^{-5}
	BBDF	7.89764×10^{-6}	8.58038×10^{-5}
	DI2BBDF	7.77654×10^{-7}	6.88387×10^{-5}
10^{-6}	ρ -DIBBDF(−0.75)	6.28992×10^{-11}	2.49431×10^{-4}
	ρ -DIBBDF(−0.60)	6.44415×10^{-11}	2.96971×10^{-4}
	ρ -DIBBDF(0.50)	9.41198×10^{-11}	3.05194×10^{-3}
	ρ -DIBBDF(0.95)	4.17385×10^{-10}	3.50504×10^{-3}
	BBDF	7.89758×10^{-8}	4.38280×10^{-3}
	DI2BBDF	6.86129×10^{-11}	3.38284×10^{-3}

Table 4. Numerical results for test problem 2.

H	Method	MAXE	TIME
10^{-2}	ρ -DIBBDF(−0.75)	3.02746×10^{-3}	6.23108×10^{-6}
	ρ -DIBBDF(−0.60)	3.08609×10^{-3}	9.37159×10^{-6}
	ρ -DIBBDF(0.50)	3.79190×10^{-3}	1.59550×10^{-5}
	ρ -DIBBDF(0.95)	6.39361×10^{-3}	3.01500×10^{-5}
	BBDF	2.27791×10^{-2}	1.89023×10^{-4}
	DI2BBDF	4.51915×10^{-3}	2.88128×10^{-5}
10^{-4}	ρ -DIBBDF(−0.75)	3.97922×10^{-7}	2.66365×10^{-5}
	ρ -DIBBDF(−0.60)	4.07670×10^{-7}	3.28285×10^{-5}
	ρ -DIBBDF(0.50)	5.95266×10^{-7}	6.75064×10^{-5}
	ρ -DIBBDF(0.95)	2.63877×10^{-6}	7.72533×10^{-5}
	BBDF	2.49799×10^{-4}	4.18826×10^{-4}
	DI2BBDF	4.33979×10^{-7}	1.15268×10^{-4}
10^{-6}	ρ -DIBBDF(−0.75)	3.99347×10^{-11}	2.33363×10^{-4}
	ρ -DIBBDF(−0.60)	4.09109×10^{-11}	2.52284×10^{-4}
	ρ -DIBBDF(0.50)	6.00101×10^{-11}	1.85521×10^{-3}
	ρ -DIBBDF(0.95)	2.85265×10^{-10}	2.54802×10^{-3}
	BBDF	2.49998×10^{-6}	9.91201×10^{-3}
	DI2BBDF	4.34093×10^{-11}	4.43944×10^{-3}

Table 5. Numerical results for test problem 3.

H	Method	MAXE	TIME
10^{-2}	ρ -DIBBDF(−0.75)	8.78849×10^{-5}	1.45541×10^{-5}
	ρ -DIBBDF(−0.60)	9.04698×10^{-5}	1.50898×10^{-5}
	ρ -DIBBDF(0.50)	1.13442×10^{-4}	1.95526×10^{-5}
	ρ -DIBBDF(0.95)	5.29869×10^{-4}	2.24700×10^{-5}
	BBDF	1.06418×10^{-2}	8.95481×10^{-4}
	DI2BBDF	2.25443×10^{-4}	4.13131×10^{-4}
10^{-4}	ρ -DIBBDF(−0.75)	1.58367×10^{-8}	1.57766×10^{-4}
	ρ -DIBBDF(−0.60)	1.62268×10^{-8}	1.94590×10^{-4}
	ρ -DIBBDF(0.50)	2.35125×10^{-8}	2.17491×10^{-4}
	ρ -DIBBDF(0.95)	9.59352×10^{-8}	3.57073×10^{-4}
	BBDF	1.11445×10^{-4}	4.94205×10^{-3}
	DI2BBDF	1.73985×10^{-8}	3.35674×10^{-3}
10^{-6}	ρ -DIBBDF(−0.75)	6.09042×10^{-11}	1.53734×10^{-3}
	ρ -DIBBDF(−0.60)	6.20290×10^{-11}	1.80857×10^{-3}
	ρ -DIBBDF(0.50)	6.62064×10^{-11}	3.11916×10^{-3}
	ρ -DIBBDF(0.95)	4.47822×10^{-10}	8.19571×10^{-3}
	BBDF	1.11489×10^{-6}	1.80581×10^{-2}
	DI2BBDF	6.30545×10^{-11}	1.49118×10^{-2}

Table 6. Numerical results for test problem 4.

H	Method	MAXE	TIME
10^{-2}	ρ -DIBBDF(−0.75)	1.45990×10^{-1}	3.69631×10^{-5}
	ρ -DIBBDF(−0.60)	1.50371×10^{-1}	4.54829×10^{-5}
	ρ -DIBBDF(0.50)	1.87600×10^{-1}	1.29152×10^{-4}
	ρ -DIBBDF(0.95)	2.43046×10^{-1}	2.59329×10^{-4}
	BBDF	1.14580×10^{25}	2.03291×10^{-4}
	DI2BBDF	6.08664×10^{-1}	1.34293×10^{-4}
10^{-4}	ρ -DIBBDF(−0.75)	5.11045×10^{-5}	1.29651×10^{-4}
	ρ -DIBBDF(−0.60)	5.23545×10^{-5}	1.70325×10^{-4}
	ρ -DIBBDF(0.50)	7.67139×10^{-5}	2.30800×10^{-4}
	ρ -DIBBDF(0.95)	3.40368×10^{-4}	3.10669×10^{-4}
	BBDF	8.16801×10^{-3}	2.37062×10^{-3}
	DI2BBDF	5.55654×10^{-5}	1.06007×10^{-3}
10^{-6}	ρ -DIBBDF(−0.75)	5.11183×10^{-9}	1.61469×10^{-2}
	ρ -DIBBDF(−0.60)	5.23685×10^{-9}	1.63234×10^{-2}
	ρ -DIBBDF(0.50)	7.68199×10^{-9}	2.65020×10^{-2}
	ρ -DIBBDF(0.95)	3.65574×10^{-8}	3.65687×10^{-2}
	BBDF	8.22481×10^{-5}	6.33621×10^{-1}
	DI2BBDF	5.55636×10^{-9}	5.27749×10^{-1}

Tables 3–6 display the numerical results of ρ -DIBBDF, BBDF and DI2BBDF using different step sizes of 10^{-2} , 10^{-4} and 10^{-6} . Based on the results, we observe that our method with $\rho = -0.75$ obtains a smaller MAXE than $\rho = -0.60, 0.50, 0.95$ for all the test problems. This is due to the stability properties discussed in Section 3, whereby a good choice of the parameter ρ , we can produce a method that has better accuracy. For MAXE, we observe that for all the ρ values, the ρ -DIBBDF outperforms the BBDF method. This is due to the nature of the BBDF method, which has more interpolating points in the fully implicit form, thus increasing the cumulative errors during the computation process (refer to [10]). Compared to DI2BBDF, our method obtains comparable accuracy for all the tested problems. For TIME, our method with $\rho = -0.75$ manages to solve the test problems in less execution time compared to BBDF and DI2BBDF.

To illustrate the performances of the suggested method and other compared methods, graphical presentations of the numerical results are shown in Figures 4–7. For a particular abscissa, the lowest value of the coordinate is considered to be more efficient at the abscissa considered. It follows that the graphs of $\log_{10} \text{MAXE}$ against $\log_{10} \text{TIME}$ demonstrate the advantage of the ρ -DIBBDF(−0.75) method over the other ρ , BBDF and DI2BBDF methods in terms of efficiency. Overall, we observe that the ρ -DIBBDF(−0.75) method performs better than the other ρ , BBDF and DI2BBDF methods.

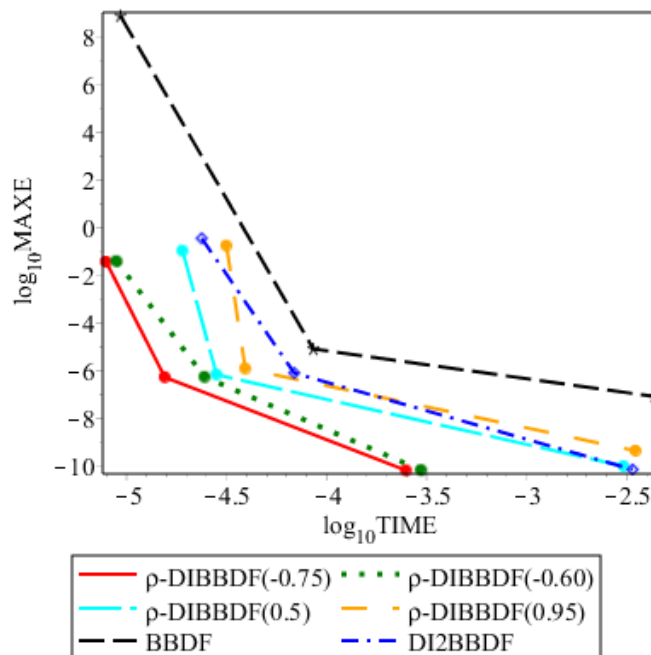


Figure 4. Efficiency curves for test problem 1.

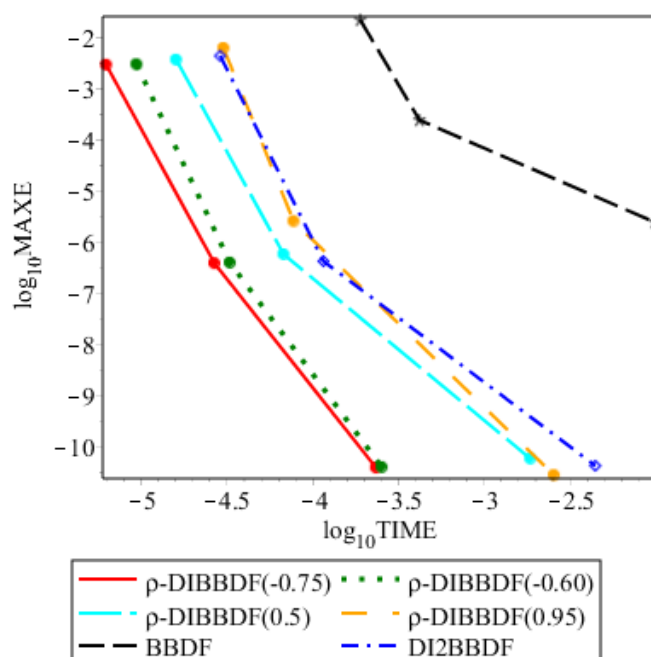


Figure 5. Efficiency curves for test problem 2.

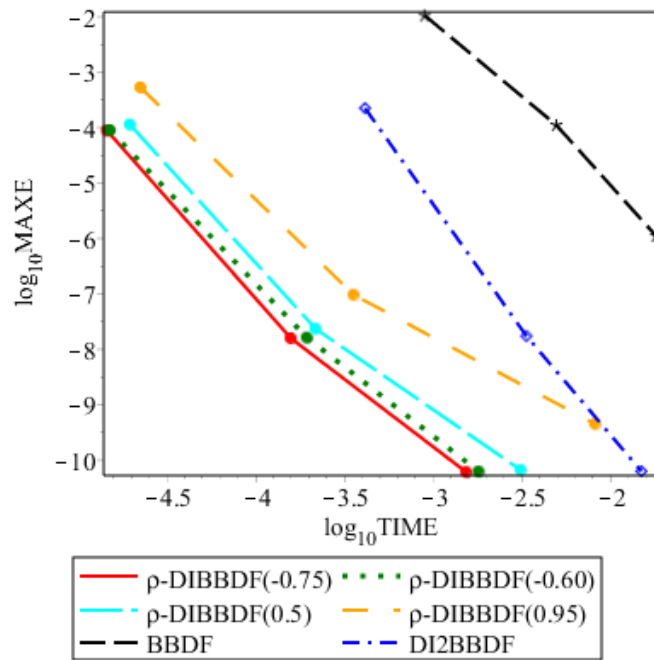


Figure 6. Efficiency curves for test problem 3.

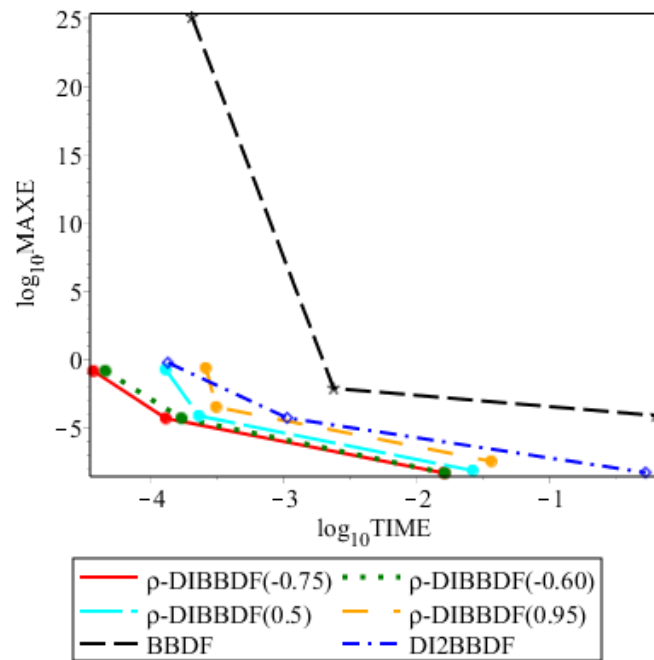


Figure 7. Efficiency curves for test problem 4.

6. Conclusions

In this study, we developed the ρ -DIBBDF method with the best choice of the parameter ρ that holds optimal stability properties. The stability analysis shows that this order 3 method is zero stable, $A(\alpha)$ -stable and convergent. Based on Section 3, we noted that the stability properties of the methods compared in Table 2 are related to the MAXE obtained in Section 5. By choosing the best value for the parameter ρ , we developed a method that possesses a smaller error constant, a reasonably big α and a larger stability region than the compared methods, thus leading to more accurate results. The proposed method performed better with higher accuracy and less execution time compared to

the existing methods, BBDF and DI2BBDF because the DI structure applied to the formula led to an efficient implementation.

Therefore, we conclude that the ρ -DIBBDF method has significance as an efficient numerical method for solving stiff first order ODEs.

Author Contributions: Conceptualization, H.M.I.; Methodology, H.M.I. and Z.B.I.; Software, H.M.I.; Validation, Z.B.I.; Formal analysis, H.M.I.; Investigation, H.M.I.; Resources, H.M.I. and Z.B.I.; Data curation, H.M.I. and Z.B.I.; Writing—original draft preparation, H.M.I.; Writing—review and editing, H.M.I. and Z.B.I.; Visualization, H.M.I.; Supervision, Z.B.I.; Project administration, Z.B.I.

Funding: This research was funded by Universiti Putra Malaysia under Grant Putra, project number GP-IPS/2017/9526600 and GP/2018/9650600.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rockswold, G.K. Implementation of α -type multistep methods for stiff differential equations. *J. Comput. Appl. Math.* **1988**, *22*, 63–69. [[CrossRef](#)]
2. Ismail, G.A.F.; Ibrahim, I.H. New efficient second derivative multistep methods for stiff systems. *Appl. Math. Model.* **1999**, *23*, 279–288. [[CrossRef](#)]
3. Butcher, J.C.; Singh, A.D. The choice of parameters in general linear methods for stiff problems. *Appl. Numer. Math.* **2000**, *34*, 59–84. [[CrossRef](#)]
4. Hojjati, G.; Rahimi Ardabili, M.Y.; Hosseini, S.M. New second derivative multistep methods for stiff systems. *Appl. Math. Model.* **2006**, *30*, 466–476. [[CrossRef](#)]
5. Okuonghae, R.I.; Ikhile, M.N.O. $A(\alpha)$ -stable linear multistep methods for stiff IVPs in ODEs. *Mathematica* **2011**, *50*, 73–90.
6. Okuonghae, R.I.; Ikhile, M.N.O. On the construction of high order $A(\alpha)$ -stable hybrid linear multistep methods for stiff IVPs in ODEs. *Numer. Anal. Appl.* **2012**, *5*, 231–241. [[CrossRef](#)]
7. Lang, J.; Hundsdorfer, W. Extrapolation-based implicit-explicit peer methods with optimised stability regions. *J. Comput. Phys.* **2017**, *337*, 203–215. [[CrossRef](#)]
8. Lambert, J.D. *Computational Methods in Ordinary Differential Equations*; John Wiley & Sons, Inc.: New York, NY, USA, 1973.
9. Ababneh, O.Y.; Ahmad, R.; Ismail, E.S. Design of new diagonally implicit Runge-Kutta methods for stiff problems. *Appl. Math. Sci.* **2009**, *3*, 2241–2253.
10. Mohd Zawawi, I.S.; Ibrahim, Z.B.; Othman, K.I. Derivation of diagonally implicit block backward differentiation formulas for solving stiff initial value problems. *Math. Probl. Eng.* **2015**, *2015*, 13. [[CrossRef](#)]
11. Isa, S.; Majid, Z.A.; Ismail, F.; Rabiei, F. Diagonally implicit multistep block method of order four for solving fuzzy differential equations using Seikkala derivatives. *Symmetry* **2018**, *10*, 42. [[CrossRef](#)]
12. Ghawadri, N.; Senu, N.; Fawzi, A.F.; Ismail, F.; Ibrahim, Z.B. Diagonally implicit Runge-Kutta type method for directly solving special fourth-order ordinary differential equations with III-posed problem of a beam on elastic foundation. *Algorithms* **2019**, *12*, 10. [[CrossRef](#)]
13. Crockatt, M.M.; Christlieb, A.J.; Garrett, C.K.; Hauck, C.D. Hybrid methods for radiation transport using diagonally implicit Runge-Kutta and space-time discontinuous Galerkin time integration. *J. Comput. Phys.* **2019**, *376*, 455–477. [[CrossRef](#)]
14. Jana Aksah, S.; Ibrahim, Z.B.; Mohd Zawawi, I.S. Stability analysis of singly diagonally implicit block backward differentiation formulas for stiff ordinary differential equations. *Mathematics* **2019**, *7*, 211. [[CrossRef](#)]
15. Pasic, H. An efficient method for solving implicit and explicit stiff differential equations. *Int. J. Numer. Meth. Eng.* **2000**, *48*, 55–78. [[CrossRef](#)]
16. Ritschel, T. Numerical Methods for Solution of Differential Equations. Bachelor's Thesis, Technical University of Denmark, Lyngby, Denmark, 2013.
17. Norsett, S.P. *Semi-Explicit Runge-Kutta Methods*; Report Mathematics and Computation No. 6/74; Dept. of Mathematics, Universiti of Trondheim: Trondheim, Norway, 1974.
18. Alexander, R. Diagonally implicit Runge-Kutta methods for solving stiff O.D.E.'s. *SIAM J. Numer. Anal.* **1977**, *14*, 1006–1021. [[CrossRef](#)]

19. Skvortsov, L.M. Diagonally implicit Runge-Kutta methods for stiff problems. *Comput. Math. Math. Phys.* **2006**, *46*, 2110–2123. [[CrossRef](#)]
20. Ibrahim, Z.B.; Othman, K.I.; Suleiman, M.B. Implicit r -point block backward differentiation formula for solving first-order stiff ODEs. *Appl. Math. Comput.* **2007**, *186*, 558–565. [[CrossRef](#)]
21. Ibrahim, Z.B.; Suleiman, M.B.; Othman, K.I. Fixed coefficients block backward differentiation formulas for the numerical solution of stiff ordinary differential equations. *Eur. J. Sci. Res.* **2008**, *21*, 508–520.
22. Suleiman M.B.; Musa, H.; Ismail F.; Senu, N.; Ibrahim, Z.B. A new superclass of block backward differentiation formula for stiff ordinary differential equations. *Asian-Eur. J. Math.* **2014**, *7*, 1–17. [[CrossRef](#)]
23. Mohd Nasir, N.A.A.; Ibrahim, Z.B.; Suleiman, M.B.; Othman, K.I.; Rahim, Y.F. Numerical solution of tumor-immune interaction using 2-point block backward differentiation method. *Int. J. Mod. Phys. Conf. Ser.* **2012**, *9*, 278–284. [[CrossRef](#)]
24. Mohd Yatim, S.A.; Ibrahim, Z.B.; Othman, K.I.; Suleiman, M.B. A numerical algorithm for solving stiff ordinary differential equations. *Math. Probl. Eng.* **2013**, *2013*, 11.
25. Gear, C.W. Algorithm 407: DIFSUB for solution of ordinary differential equations. *Commun. ACM* **1971**, *14*, 185–190. [[CrossRef](#)]
26. Vijitha-Kumara, K.H.Y. Variable Stepsize Variable Order Multistep Methods for Stiff Ordinary Differential Equations. Ph.D. Thesis, Iowa State University, Ames, IA, USA, 1985.
27. Dahlquist, G. A special stability problem for linear multistep methods. *BIT* **1963**, *3*, 27–43. [[CrossRef](#)]
28. Ibrahim, Z.B.; Mohd Noor, N.; Othman, K.I. Fixed coefficient $A(\alpha)$ stable block backward differentiation formulas for stiff ordinary differential equations. *Symmetry* **2019**, *11*, 846 [[CrossRef](#)]
29. Liniger, W. A criterion for A -stability of linear multistep integration formulae. *Computing* **1968**, *3*, 280–285. [[CrossRef](#)]
30. Rockswold, G.K. Stable Variable Step Stiff Methods for Ordinary Differential Equations. Ph.D. Thesis, Iowa State University, Ames, IA, USA, 1983.
31. Jeltsch, R. Stiff stability and its relation to A_0 - and $A(0)$ -stability. *SIAM J. Numer. Anal.* **1976**, *13*, 8–17. [[CrossRef](#)]
32. Lambert, J.D. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*; John Wiley & Sons, Inc.: New York, NY, USA, 1991.
33. Widlund, O.B. A note on unconditionally stable linear multistep methods. *BIT* **1967**, *7*, 65–70. [[CrossRef](#)]
34. Butcher, J.C. Forty-five years of A -stability. *J. Numer. Anal. Ind. Appl. Math.* **2009**, *4*, 1–9.
35. Hall, G.; Watt, J.M. *Modern Numerical Methods for Ordinary Differential Equations*; Clarendon Press: Oxford, UK, 1976.
36. Kushnir, D.; Rokhlin, V. A highly accurate solver for stiff ordinary differential equations. *SIAM J. Sci. Comput.* **2011**, *34*, 1296–1315. [[CrossRef](#)]
37. Lee, H.C.; Chen, C.K.; Hung, C.I. A modified group-preserving scheme for solving the initial value problems of stiff ordinary differential equations. *Appl. Math. Comput.* **2002**, *133*, 445–459. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).