ABSTRACT
Stiff initial value problems in ordinary differential equations occur when solution components evolve at varying rates, posing challenges for traditional computational methods. Specialized techniques are crucial for maintaining accuracy and stability during rapid transitions, emphasizing their significance in developing reliable numerical algorithms across scientific and engineering applications. This study aims to develop a new fixed coefficient 3-point diagonally implicit block backward differentiation formula for the numerical solution of first order stiff initial value problems. The method is constructed by integrating a triangular matrix into the coefficient matrix of an existing extended 3-point super class of block BDF for solving stiff initial value problems. The selection of a fixed coefficient within the interval accompanies this integration \((-1, 1)\) to ensure optimal stability. The method is found to order five. Stability analysis indicates that the method is consistent, zero-stable, and almost A-stable, validating its applicability to stiff initial value problems. Implementation of the method involves Newton’s iteration, and a code in the C programming language is devised to demonstrate its effectiveness. Comparative examination of numerical outcomes with the existing 3BBDF and 3ESBBDF methods highlights the proposed method’s enhanced accuracy and reduced computation time.

INTRODUCTION
In this research, we explore the general form of the first-order stiff initial value problems (IVPs) presented as follows:

\[ y' = f(x, y), \quad y(a) = y_0, \quad a \leq x \leq b \]  

(1)

It is postulated that the function \( f(x, y) \) adheres to the Lipschitz conditions (Alhassan et al., 2023). Despite various attempts by researchers to apply diverse analytical approaches to solve equation (1), solutions to specific initial value problems (IVPs) have proven to be intricate or beyond analytical resolution. Hence, the necessity of advocating for numerical methods becomes apparent. If the solution of IVPs (1) using a specified numerical method becomes unstable when a large number of step lengths are chosen due to its physical property of causing rapid variation in the solution, then it is called Stiff IVPs. Stiff IVPs are commonly encountered in Chemical kinetics, electric circuits, string variations, control systems, and more (Musa et al., 2022).

When employing a designated numerical method for solving IVPs (1), instability emerges when opting for an extensive number of step lengths, attributed to its inherent characteristic of inducing swift variation in the solution. This phenomenon is termed Stiff IVPs and is frequently encountered in chemical kinetics, electric circuits, string variation, control systems, and other domains (Musa et al., 2022).

Numerical methods for tackling stiff IVPs can be classified as either block or non-block, and both can be explicit or implicit. Implicit Linear Multistep Methods (LMM) have demonstrated superior effectiveness in addressing stiff IVPs compared to their explicit counterparts. Instances of non-block implicit methods are documented in [Cash, 1980; Curtiss & Hirschfelder, 1952; Dalquist, 1974; Alexander, 1977], while examples of block implicit methods are outlined in [Musa et al., 2012; Ibrahim et al., 2007a; Musa et al., 2022; Suleiman et al., 2014; Musa & Muhammad, 2019; Alhassan et al., 2023; Bala et al., 2022]. Ibrahim et al. (2007b & 2019) identified...
an implicit fixed coefficient block method for solving stiff IVPs.

This research proposes a new fixed coefficient diagonally implicit method grounded in the block backward differentiation formula (NFDIBBDF) for addressing stiff initial value problems. The subsequent sections will explore the derivation of the method, stability analysis, and implementation, highlighting the potential of our innovative numerical approach in overcoming this pivotal challenge.

**METHODOLOGY**

This section contains the derivation of the proposed method using Taylor’s series and the derivation of the predictor method, which will predict the initial approximation for stiff initial value problems.

**Derivation of (NFDIBBDF) Method**

Consider the extended 3-point super class of block backward differentiation formula for solving stiff initial value problems developed by Musa et al. (2019) of the form:

\[
\sum_{i=1}^{n} a_{i,j}y_{n+j-2} = h\beta_{k,i}(f_{n+k} - pf_{n+k-2}), \quad k = 1,2,3 \tag{2}
\]

In this research, we will modify (2) by introducing a triangular matrix in the coefficient matrix of the method (2) and choosing the value of \( p = \frac{3}{5} \) from the interval \((-1,1)\) which gives better stability region required for solving stiff IVPs. The new formula would compute the approximated solution values \( y_{n+1}, y_{n+2} \) and \( y_{n+3} \) simultaneously in a block using three previous values \( y_{n-2}, y_{n-1} \) and \( y_n \) with constant step size \( h \).

**Definition 1: A New Fixed Coefficient Diagonally Implicit Block Backward Differentiation Formula (NFDIBBDF)** is defined as:

\[
\sum_{j=0}^{2+k} a_{i,j}y_{n+j-2} = h\beta_{k,i}(f_{n+k} - \frac{3}{5}f_{n+k-2}), \quad k = 1,2,3 \tag{3}
\]

To derive the first point \( y_{n+1} \), we define the linear operator of (3) associated with the first point as:

\[
L_1[y(x_n), h]: a_{0,0}y_{n-2} + a_{1,1}y_{n-1} + a_{2,1}y_n + a_{3,1}y_{n+1} - h\beta_{1,1}(f_{n+1} - \frac{3}{5}f_{n-1}) = 0, \tag{4}
\]

By expanding (4) as Taylor’s series about any point \( x_n \) and after collecting like terms, we get:

\[
c_0y(x_n) + c_1h^2y''(x_n) + c_2h^3y'''(x_n) + c_3h^4y''''(x_n) + \ldots = 0 \tag{5}
\]

where,

\[
\begin{align*}
c_0 &= a_{0,1} + a_{1,1} + a_{2,1} + a_{3,1} = 0 \\
c_{1,1} &= -2a_{0,1} - a_{1,1} + a_{3,1} - \frac{2}{5}\beta_{1,1} = 0 \\
c_{2,1} &= 2a_{0,1} + \frac{1}{2}a_{1,1} + a_{3,1} - \frac{2}{5}\beta_{1,1} = 0 \\
c_{3,1} &= -\frac{4}{3}a_{0,1} - \frac{1}{6}a_{1,1} + \frac{1}{6}a_{3,1} - \frac{1}{5}\beta_{1,1} = 0
\end{align*} \tag{6}
\]

Solving the system of simultaneous equations in (6) for the values of \( a_{j,l} \) and \( \beta_{j,l} \) by normalizing the coefficient of \( y_{n+1} \) to one and substituting the values obtained in (4) yields the formula for the first point as:

\[
y_{n+1} = \frac{2}{29}y_{n-2} - \frac{27}{29}y_{n-1} + \frac{54}{29}y_n + \frac{15}{29}hf_{n+1} - \frac{5}{29}hf_{n-1} \tag{7}
\]

Similarly, to derive the second point \( y_{n+2} \), we defined the linear operator associated with the second point as:

\[
L_2[y(x_n), h]: a_{0,2}y_{n-2} + a_{1,2}y_{n-1} + a_{2,2}y_n + a_{3,2}y_{n+1} + a_{4,2}y_{n+2} - h\beta_{2,2}(f_{n+2} - \frac{3}{5}f_n) = 0, \tag{8}
\]

The corresponding approximate relationship for the equation (8) is given by:

\[
a_{0,2}y(x_n - 2h) + a_{1,2}y(x_n - h) + a_{2,2}y(x_n) + a_{3,2}y(x_n + h) + a_{4,2}y(x_n + 2h) \tag{9}
\]

\[-h\beta_{2,2}(y'(x_n + 2h) - \frac{2}{5}y(x_n)) = 0
\]

Again, by expanding (9) as Taylor’s series about any point \( x_n \) and collecting like terms, we get:

\[
c_{0,2}y(x_n) + c_{1,2}h^2y''(x_n) + c_{2,2}h^3y'''(x_n) + c_{3,2}h^4y''''(x_n) + \ldots = 0 \tag{10}
\]

where,

\[
\begin{align*}
c_{0,2} &= a_{0,2} + a_{1,2} + a_{2,2} + a_{3,2} + a_{4,2} = 0 \\
c_{1,2} &= -2a_{0,2} - a_{1,2} + a_{3,2} + 2a_{4,2} - \frac{2}{5}\beta_{2,2} = 0 \\
c_{2,2} &= 2a_{0,2} + \frac{1}{2}a_{1,2} + \frac{1}{2}a_{3,2} + 2a_{4,2} - 2\beta_{2,2} = 0 \\
c_{3,2} &= -\frac{4}{3}a_{0,2} - \frac{1}{6}a_{1,2} + \frac{1}{6}a_{3,2} + \frac{1}{3}a_{4,2} - \frac{2}{5}\beta_{2,2} = 0 \\
c_{4,2} &= \frac{2}{5}a_{0,2} + \frac{1}{24}a_{1,2} + \frac{1}{24}a_{3,2} + \frac{1}{3}a_{4,2} + \frac{2}{5}\beta_{2,2} = 0
\end{align*} \tag{11}
\]

The coefficient of \( y_{n+2} \) is similarly normalized to 1, by adopting the same procedure as in the derivation of the first point, we obtain the following formula for the second point:

\[
y_{n+2} = -\frac{3}{35}y_{n-2} + \frac{7}{16}y_{n-1} - \frac{45}{32}y_n + \frac{33}{16}y_{n+1} + \frac{15}{32}hf_{n+2} - \frac{32}{32}hf_n \tag{12}
\]

In obtaining the third point formula, a similar procedure is applied as in the derivation of the first and second point formulae yields:
Thus, by combining the formulae in (7), (12), and (13), we have obtained a New 3-point fixed coefficient diagonally implicit block backward differentiation formula (NFDIBBDF) as:

\[
\begin{cases}
C_{0,1} = γ_{0,1} + γ_{1,1} + γ_{2,1} + γ_{3,1} + γ_{4,1} = 0 \\
C_{1,1} = -3γ_{0,1} - 2γ_{1,1} - γ_{2,1} + γ_{4,1} = 0 \\
C_{2,1} = \frac{9}{2}γ_{0,1} + 2γ_{1,1} + \frac{1}{2}γ_{2,1} + \frac{1}{2}γ_{4,1} = 0 \\
C_{3,1} = -\frac{9}{2}γ_{0,1} - \frac{4}{3}γ_{1,1} - \frac{1}{6}γ_{2,1} + \frac{1}{6}γ_{4,1} = 0
\end{cases}
\]

Solving these set of equations in the Maple18 environment after setting \(γ_{4,1} = 1\), we obtain the coefficient for the first point given in Table 1 below:

**Table 1: Coefficient of the first point**

<table>
<thead>
<tr>
<th></th>
<th>(γ_{0,1})</th>
<th>(γ_{1,1})</th>
<th>(γ_{2,1})</th>
<th>(γ_{3,1})</th>
<th>(γ_{4,1})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>−4</td>
<td>6</td>
<td>−4</td>
<td>1</td>
</tr>
</tbody>
</table>

By substituting these obtained coefficients in equation (17), the first point formula is therefore obtained as:

\[
y_{n-3} − 4y_{n-2} + 6y_{n-1} − 4y_n + y_{n+1} = 0
\]

Which is equivalent to

\[
y_{n+1} = −y_{n-3} + 4y_{n-2} − 6y_{n-1} + 4y_n
\]

**Second Point: k = 2.**

Likewise, when the coefficients \(γ_{4,2}\) and \(γ_{6,2}\) are set to zero in equation (16), the linear operator (16) transforms into:

\[
y_{n-3} + γ_{1,2}y_{n-2} + γ_{2,2}y_{n-1} + γ_{3,2}y_n + γ_{5,2}y_{n+2} = 0
\]

The following system of simultaneous linear equations is derived by expanding the Taylor series around the point \(x_n\), resulting in:

\[
\begin{cases}
C_{0,2} = γ_{0,2} + γ_{1,2} + γ_{2,2} + γ_{3,2} + γ_{5,2} = 0 \\
C_{1,2} = -3γ_{0,2} - 2γ_{1,2} - γ_{2,2} + 2γ_{5,2} = 0 \\
C_{2,2} = \frac{9}{2}γ_{0,2} + 2γ_{1,2} + \frac{1}{2}γ_{2,2} + 2γ_{5,2} = 0 \\
C_{3,2} = -\frac{9}{2}γ_{0,2} - \frac{4}{3}γ_{1,2} - \frac{1}{6}γ_{2,2} + \frac{1}{6}γ_{5,2} = 0
\end{cases}
\]

After setting \(γ_{5,2} = 1\) and solving this set of equations in the Maple 18 environment, we acquire the coefficients for the second point as presented in Table 2 below:
By substituting these acquired coefficients into equation (17), we derive the formula for the second point as follows

$$4y_{n-3} - 15y_{n-2} + 20y_{n-1} - 10y_n + y_{n+2} = 0$$ (23)

This is, therefore, equivalent to

$$y_{n+2} = -4y_{n-3} + 15y_{n-2} - 20y_{n-1} + 10y_n$$ (24)

**Third Point: k = 3**

To obtain the formula for the third point, we apply the same procedure as in the derivation of the first and second point formulas, resulting in:

$$y_{n+3} = -10y_{n-3} + 36y_{n-2} - 45y_{n-1} + 20y_n$$ (25)

Hence, the 3-point explicit block predictor method is therefore given by

$$y_{n+1} = -y_{n-3} + 4y_{n-2} - 6y_{n-1} + 4y_n$$

$$y_{n+2} = -4y_{n-3} + 15y_{n-2} - 20y_{n-1} + 10y_n$$

$$y_{n+3} = -10y_{n-3} + 36y_{n-2} - 45y_{n-1} + 20y_n$$

**STABILITY OF THE (NFDIBDF) METHOD**

The stability of implicit numerical methods for stiff initial value problems is essential for preventing numerical instabilities and obtaining accurate and reliable solutions. Zero and A-stability are key criteria in assessing the robustness of this method (14), allowing for efficient simulations of problems with disparate timescales (Cash, 2015).

**Definition 2** (Zero stability): A block method (14) is said to be zero stable if all the roots of first characteristics

$$y_{n+1} = \frac{2}{29}y_{n-2} - \frac{27}{29}y_{n-1} + \frac{54}{29}y_n + \frac{15}{29}h\lambda y_{n+1} - \frac{9}{29}h\lambda y_{n-1}$$

$$y_{n+2} = -\frac{3}{32}y_{n-2} + \frac{7}{16}y_{n-1} - \frac{45}{32}y_n + \frac{33}{16}y_{n+1} + \frac{45}{32}h\lambda y_{n+2} - \frac{9}{32}h\lambda y_n$$

$$y_{n+3} = \frac{27}{347}y_{n-2} - \frac{165}{347}y_{n-1} + \frac{410}{347}y_n - \frac{720}{347}y_{n+1} + \frac{795}{347}y_{n+2} + \frac{150}{347}h\lambda y_{n+3} - \frac{90}{347}h\lambda y_{n+1}$$

Rearranging and collecting the like terms of equation (27) leads to

$$\begin{align*}
(1 - \frac{15}{29}h\lambda) y_{n+1} &= \frac{2}{29}y_{n-2} - \frac{27}{29}y_{n-1} + \frac{54}{29}y_n + \frac{15}{29}h\lambda y_{n+1} - \frac{9}{29}h\lambda y_{n-1} \\
(1 - \frac{33}{16}h\lambda) y_{n+2} &= -\frac{3}{32}y_{n-2} + \frac{7}{16}y_{n-1} - \frac{45}{32}y_n + \frac{33}{16}y_{n+1} + \frac{45}{32}h\lambda y_{n+2} - \frac{9}{32}h\lambda y_n \\
(1 - \frac{150}{347}h\lambda) y_{n+3} &= -\frac{795}{347}y_{n+2} + \frac{720}{347}y_{n+1} + \frac{90}{347}h\lambda y_{n+3} + \frac{410}{347}y_n
\end{align*}$$

(28)

The matrix formulation of these equations is written as:

<table>
<thead>
<tr>
<th>$y_{0,2}$</th>
<th>$y_{1,2}$</th>
<th>$y_{2,2}$</th>
<th>$y_{3,2}$</th>
<th>$y_{5,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>-15</td>
<td>20</td>
<td>-10</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Coefficient of the second point

For an implicit numerical method to be A-stable, its stability region in the complex plane should include the entire left-half plane (Lambert, 1991). A-stability, short for absolute stability, is a desirable property of numerical methods for solving ordinary differential equations (ODEs). Specifically, it pertains to implicit methods used in the numerical integration of ODEs. A-stability ensures that the numerical method remains stable over a wide range of problem characteristics and time step sizes (Lambert, 1973).

**Definition 3** (A-stability): A block method (14) is said to be A-stable if the stability region covers the entire negative half plane (Lambert, 1991).

A-stability is crucial when dealing with stiff ODEs, where the variables have significant differences in timescales. Stiff problems can be challenging for numerical methods, and A-stable methods provide robustness by allowing for larger time steps without sacrificing stability (Cash, 1980).

Implicit methods involve solving algebraic equations at each time step and often possess A-stability. This property makes them particularly well-suited for stiff problems, as they can efficiently handle the numerical integration without being overly sensitive to the choice of time step (Suleiman et al., 2014).

Hence, in this section, we will examine the zero-stability and A-stability of our method. This analysis will be conducted using a first-order scalar differential equation represented as $y' = \lambda y$. By applying this equation to the expressions in formula (14), we obtain:

https://scientifica.umyu.edu.ng/  Yusuf et al., /USci, 3(1): 001 – 014, March 2024

4
Putting $\bar{h} = h\lambda$ in matrix equation (29), we have

$$\begin{bmatrix} (1 - \frac{15}{29} h\lambda) & 0 & 0 \\ \frac{33}{16} & (1 - \frac{15}{32} h\lambda) & 0 \\ \frac{720}{347} + \frac{90}{347} h\lambda & -\frac{795}{347} & (1 - \frac{150}{347} h\lambda) \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \end{bmatrix} = \begin{bmatrix} \frac{2}{29} & \frac{-27}{29} & \frac{9}{29} h\lambda \\ \frac{-3}{32} & \frac{7}{16} & \frac{45}{32} h\lambda \\ \frac{27}{347} & \frac{-165}{347} & \frac{410}{347} \end{bmatrix} \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}$$

(29)

If $m$ is the number of block and $r$ is the number of points in the block, then $n = mr$, where $r = 3$ and $n = 3m$. By (Bala et al., 2022), we let

$$Y_m = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \end{bmatrix} = \begin{bmatrix} y_{3m+1} \\ y_{3m+2} \\ y_{3m+3} \end{bmatrix}$$

Equation (30) can also be expressed in the following form:

$$C_0 Y_m = C_1 Y_{m-1}$$

(31)

where,

$$C_0 = \begin{bmatrix} (1 - \frac{15}{29} h\lambda) & 0 & 0 \\ \frac{-33}{16} & (1 - \frac{15}{32} h\lambda) & 0 \\ \frac{720}{347} + \frac{90}{347} h\lambda & -\frac{795}{347} & (1 - \frac{150}{347} h\lambda) \end{bmatrix}, \quad C_1 = \begin{bmatrix} \frac{2}{29} & \frac{-27}{29} & \frac{9}{29} h\lambda \\ \frac{-3}{32} & \frac{7}{16} & \frac{45}{32} h\lambda \\ \frac{27}{347} & \frac{-165}{347} & \frac{410}{347} \end{bmatrix}$$

The characteristic polynomial of the method is obtained by evaluating

$$\pi(u, \bar{h}) = det(C_0 u - C_1) = 0,$$

(32)

we obtain:

$$\pi(u, \bar{h}) = \begin{bmatrix} (1 - \frac{15}{29} h\lambda) & 0 & 0 \\ \frac{-33}{16} & (1 - \frac{15}{32} h\lambda) & 0 \\ \frac{720}{347} + \frac{90}{347} h\lambda & -\frac{795}{347} & (1 - \frac{150}{347} h\lambda) \end{bmatrix} u - \begin{bmatrix} \frac{2}{29} & \frac{-27}{29} & \frac{9}{29} h\lambda \\ \frac{-3}{32} & \frac{7}{16} & \frac{45}{32} h\lambda \\ \frac{27}{347} & \frac{-165}{347} & \frac{410}{347} \end{bmatrix} = 0,$$

$$\begin{array}{c}
\frac{8289}{11104} u^3 - \frac{476961}{322016} u^2 - \frac{144531}{161008} u - \frac{397575}{322016} u^3 h^2 - \frac{456705}{322016} u^2 h + \frac{7425}{11104} u h^2 - \frac{16875}{161008} u h^3 - \frac{108}{10063} u\hfill \\
+ \frac{9843}{20126} u - \frac{2543}{322016} u^3 - \frac{3645}{161008} u^2 + \frac{142965}{322016} u^3 h^2 - \frac{2187}{322016} u^2 h + u^3 = 0
\end{array}$$

(33)

https://scientificum.umy.edu.ng/
For absolute stability of the method, the stability region is obtained by substituting $u = e^{i\theta}$ into (33). The graph of stability region for the method is given below:

![Stability Region Graph](image)

**Figure 1:** Stability region of the method.

Following the definition of $A$-stability, the method (14) is nearly $A$-stable, as its stability region encompasses the entire negative half-plane. Hence, the method is suitable for the numerical integration of stiff ordinary differential equations.

For zero stability, we set $\bar{h} = 0$ in equation (33) to obtain

$$
\pi(u,0) = \det \left[ \begin{array}{ccc}
1 & 0 & 0 \\
\frac{1}{33} & 1 & 0 \\
\frac{1}{16} & \frac{1}{720} & 1
\end{array} \right] u - \left[ \begin{array}{ccc}
\frac{2}{29} & -\frac{27}{29} & \frac{54}{29} \\
\frac{3}{7} & \frac{16}{32} & \frac{45}{32} \\
\frac{27}{347} & -\frac{165}{347} & \frac{410}{347}
\end{array} \right] = 0,
$$

(34)

Evaluating the above determinant leads to the first characteristic polynomial as:

$$
-\frac{476961}{322016} u^2 + \frac{9843}{20126} u - \frac{2543}{322016} u^3 = 0
$$

(35)

By solving the cubic equation (35), we obtained the roots of the first characteristic polynomial as:

$$u = 0.0170138731, u = 0.4641578699, u = 1$$

And whose modulus are; 0.0170138731, 0.4641578699 and 1. Hence, from definition (3), we conclude that the method (14) is zero stable.

**IMPLEMENTATION OF THE (NFDIBBDF) METHOD**

The NFDIBBDF method is implemented by applying the idea of Newton’s iteration. We start by writing the formula (14) in the form:
The implementation of the equation (43) will be carried out using a C programming language code.
**TEST PROBLEMS USED**

The following stiff initial value problems show the performance of the method developed.

**Problem 1:**

\[ y' = -9y, \quad y(0) = e, \quad 0 \leq x \leq 1 \]

**Exact Solution:**

\[ y(x) = e^{(1-9x)} \]

Source: (Musa et al., 2012)

**Problem 2:**

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

**Exact Solution:**

\[ y(x) = x - e^{-5x} \]

Source: (Lee et al., 2002)

**Problem 3:**

\[ y' = -20y + 20\sin x + \cos x, \quad y(0) = 1, \quad 0 \leq x \leq 2 \]

**Exact Solution:**

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

Source: (Musa et al., 2015)

**Problem 4:**

\[ y'_1 = -20y_1 - 19y_2, \quad y_1(0) = 2 \quad 0 \leq x \leq 20 \]

\[ y'_2 = -19y_1 - 20y_2, \quad y_2(0) = 0 \]

**Exact Solution:**

\[ y_1(x) = e^{-39x} + e^{-x} \]

\[ y_2(x) = e^{-39x} - e^{-x} \]

Eigenvalues: -1 and -39

Source: (Musa et al., 2014)

**Problem 5:**

\[ y'_1 = 198y_1 + 199y_2, \quad y_1(0) = 1 \quad 0 \leq x \leq 10 \]

\[ y'_2 = -398y_1 - 399y_2, \quad y_2(0) = -1 \]

**Exact Solution:**

\[ y_1(x) = e^{-x} \]

\[ y_2(x) = -e^{-x} \]

Eigenvalues: -1 and -200

Source: (Musa et al., 2014)

**NUMERICAL COMPUTATIONS**

The three (3) selected test problems will now be solved numerically using the 3-point BBDF, 3-point Extended SBBDF, and NFDBBBDF methods. This is to compare the efficiency and accuracy of the methods; the maximum absolute errors obtained from different step lengths \( H \) are given in each problem. The tables below also show the number of steps taken to solve each problem and computational time. For easy referencing, the existing 3-point block backward differentiation formula developed by Ibrahim et al. (2007a) is given by:
\begin{align}
\frac{y_{n+1}}{10}y_{n-2} - &\frac{3}{4}y_{n-1} + 3y_n - \frac{3}{2}y_{n+2} + \frac{1}{20}y_{n+3} + 3hf_{n+1} \\
\frac{y_{n+2}}{65}y_{n-2} + &\frac{4}{13}y_{n-1} - \frac{12}{17}y_n + \frac{24}{17}y_{n+1} - \frac{12}{65}y_{n+3} + \frac{12}{17}hf_{n+2} \\
\frac{y_{n+3}}{137}y_{n-2} - &\frac{75}{137}y_{n-1} + \frac{200}{137}y_n - \frac{300}{137}y_{n+1} + \frac{300}{137}y_{n+2} + \frac{60}{137}hf_{n+1}
\end{align}

However, the extended 3-point super class of block backward differentiation formula developed by Musa et al. (2019) is given by:

\begin{align}
\frac{y_{n+1}}{70}y_{n-2} - &\frac{37}{45}y_{n-1} + \frac{9}{7}y_n + \frac{23}{14}y_{n+2} - \frac{27}{140}y_{n+3} - \frac{15}{7}hf_{n+1} - \frac{12}{7}hf_{n-1} \\
\frac{y_{n+2}}{265}y_{n-2} + &\frac{44}{53}y_{n-1} - \frac{44}{53}y_n + \frac{72}{53}y_{n+2} - \frac{68}{265}y_{n+3} + \frac{48}{53}hf_{n+2} + \frac{48}{53}hf_n \\
\frac{y_{n+3}}{673}y_{n-2} - &\frac{435}{673}y_{n-1} + \frac{1240}{673}y_n - \frac{1580}{673}y_{n+1} + \frac{1380}{673}y_{n+2} + \frac{300}{673}hf_{n+3} + \frac{240}{673}hf_{n+1}
\end{align}

Tables 3, 4, 5, 6, through 7 below give the numerical results. The following notations are used in the tables:

- H: Step length/size
- METHOD: Methods used
- NS: Number of steps
- 3BBDF: 3-point Block BDF method
- 3ESBBDF: 3-point Extended Superclass of Block BDF method
- NFDIBBDF: A New Fixed Coefficient Diagonally Implicit Block BDF method
- MAXE: Maximum Error
- CPU TIME: Computation Time (in seconds).

### Table 3: Numerical Result for Problem 1.

<table>
<thead>
<tr>
<th>H</th>
<th>METHOD</th>
<th>NS</th>
<th>MAXE</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>3BBDF</td>
<td>33</td>
<td>1.75664e-001</td>
<td>2.00833e-004</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>33</td>
<td>6.50071e-002</td>
<td>6.56100e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>33</td>
<td>3.52244e-002</td>
<td>2.70300e-002</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>3BBDF</td>
<td>333</td>
<td>2.63192e-002</td>
<td>1.36950e-003</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>333</td>
<td>6.50122e-004</td>
<td>1.88100e-001</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>333</td>
<td>6.19415e-004</td>
<td>2.53900e-002</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>3BBDF</td>
<td>3,333</td>
<td>2.69331e-003</td>
<td>1.29261e-002</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>3,333</td>
<td>6.50122e-006</td>
<td>1.13100e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>3,333</td>
<td>6.93783e-006</td>
<td>1.10700e-002</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>3BBDF</td>
<td>33,333</td>
<td>2.69933e-004</td>
<td>1.28720e-001</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>33,333</td>
<td>6.50123e-008</td>
<td>9.13000e-000</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>33,333</td>
<td>7.06579e-008</td>
<td>1.80500e-001</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>3BBDF</td>
<td>333,333</td>
<td>2.69933e-005</td>
<td>1.30950e+000</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>333,333</td>
<td>6.50123e-010</td>
<td>9.62100e+001</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>333,333</td>
<td>7.08360e-010</td>
<td>1.13500e+001</td>
</tr>
</tbody>
</table>

### Table 4: Numerical Result for Problem 2.

<table>
<thead>
<tr>
<th>H</th>
<th>METHOD</th>
<th>NS</th>
<th>MAXE</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>3BBDF</td>
<td>33</td>
<td>2.80735e-002</td>
<td>2.76330e-004</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>33</td>
<td>4.83217e-003</td>
<td>6.23441e-005</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>33</td>
<td>4.16232e-003</td>
<td>6.20100e-005</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>3BBDF</td>
<td>333</td>
<td>3.71852e-003</td>
<td>1.81850e-003</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>333</td>
<td>5.95338e-005</td>
<td>1.88100e-001</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>333</td>
<td>7.01937e-005</td>
<td>1.79400e-001</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>3BBDF</td>
<td>3,333</td>
<td>3.74700e-005</td>
<td>1.71443e-002</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>3,333</td>
<td>5.95692e-007</td>
<td>6.48433e-003</td>
</tr>
</tbody>
</table>
Table 5: Numerical Result for Problem 3.

<table>
<thead>
<tr>
<th>H</th>
<th>METHOD</th>
<th>NS</th>
<th>MAXE</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-5}</td>
<td>3BBDF</td>
<td>33,333</td>
<td>3.74970e-005</td>
<td>1.70042e-001</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>33,333</td>
<td>5.95974e-009</td>
<td>6.58687e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>33,333</td>
<td>8.02119e-009</td>
<td>6.55600e-002</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>3BBDF</td>
<td>33,333</td>
<td>3.74997e-006</td>
<td>1.70308e+000</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>33,333</td>
<td>6.18636e-011</td>
<td>9.62100e+001</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>33,333</td>
<td>8.04294e-011</td>
<td>1.49600e+001</td>
</tr>
</tbody>
</table>

Table 6: Numerical Result for Problem 4.

<table>
<thead>
<tr>
<th>H</th>
<th>METHOD</th>
<th>NS</th>
<th>MAXE</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>3BBDF</td>
<td>666</td>
<td>9.15007e-002</td>
<td>5.69750e-004</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>666</td>
<td>4.49329e-002</td>
<td>3.37200e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>666</td>
<td>3.93265e-002</td>
<td>1.93500e-002</td>
</tr>
<tr>
<td>10^{-3}</td>
<td>3BBDF</td>
<td>6,666</td>
<td>2.08350e-002</td>
<td>4.54233e-003</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>6,666</td>
<td>9.23116e-004</td>
<td>1.05100e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>6,666</td>
<td>1.00816e-003</td>
<td>1.04300e-002</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>3BBDF</td>
<td>6,666</td>
<td>2.19484e-003</td>
<td>4.34752e-002</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>6,666</td>
<td>9.92032e-006</td>
<td>7.45800e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>6,666</td>
<td>1.23565e-005</td>
<td>7.12400e-002</td>
</tr>
<tr>
<td>10^{-5}</td>
<td>3BBDF</td>
<td>666,666</td>
<td>2.20579e-004</td>
<td>4.34533e-002</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>666,666</td>
<td>9.99200e-008</td>
<td>3.36900e+000</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>666,666</td>
<td>1.27985e-007</td>
<td>3.68200e-001</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>3BBDF</td>
<td>6,666,666</td>
<td>2.20688e-005</td>
<td>4.33535e+000</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>6,666,666</td>
<td>9.99920e-010</td>
<td>2.96100e+001</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>6,666,666</td>
<td>1.28640e-009</td>
<td>7.45500e+000</td>
</tr>
</tbody>
</table>

Table 7: Numerical Result for Problem 5.

<table>
<thead>
<tr>
<th>H</th>
<th>METHOD</th>
<th>NS</th>
<th>MAXE</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>3BBDF</td>
<td>333</td>
<td>1.07308e-002</td>
<td>1.37500e-002</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>333</td>
<td>1.83217e-002</td>
<td>7.36289e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>333</td>
<td>2.78034e-004</td>
<td>1.41100e-002</td>
</tr>
<tr>
<td>10^{-3}</td>
<td>3BBDF</td>
<td>3,333</td>
<td>1.10060e-002</td>
<td>2.72200e-002</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>3,333</td>
<td>8.05338e-002</td>
<td>5.81512e-002</td>
</tr>
<tr>
<td></td>
<td>NFDIBBDF</td>
<td>3,333</td>
<td>3.14469e-006</td>
<td>1.91300e-002</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>3BBDF</td>
<td>33,333</td>
<td>1.10333e-004</td>
<td>2.02700e-001</td>
</tr>
<tr>
<td></td>
<td>3ESBBDF</td>
<td>33,333</td>
<td>1.26692e-008</td>
<td>5.81491e-001</td>
</tr>
</tbody>
</table>
The visual impact on the performance of the method developed is presented below. $\log_{10}(MAXE)$ against $\log_{10}(H)$ each test problem is plotted with the aid of MATLAB (Figure 2-6).

<table>
<thead>
<tr>
<th>Method</th>
<th>$3BBDF$</th>
<th>$3ESBBDF$</th>
<th>$NFDIBBDF$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-5}$</td>
<td>$33,333$</td>
<td>$1.1036e-005$</td>
<td>$1.92600e+000$</td>
</tr>
<tr>
<td></td>
<td>$33,333$</td>
<td>$1.3274e-010$</td>
<td>$5.81122e+000$</td>
</tr>
<tr>
<td></td>
<td>$33,333$</td>
<td>$3.2171e-010$</td>
<td>$4.40900e+000$</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>$3,333,333$</td>
<td>$1.10363e-006$</td>
<td>$1.91700e+001$</td>
</tr>
<tr>
<td></td>
<td>$3,333,333$</td>
<td>$1.33362e-012$</td>
<td>$5.79987e+001$</td>
</tr>
<tr>
<td></td>
<td>$3,333,333$</td>
<td>$2.38012e-011$</td>
<td>$4.23200e+001$</td>
</tr>
</tbody>
</table>

Figure 2: Efficiency Curve for Problem 1

Figure 3: Efficiency Curve for problem 2
Figure 4: Efficiency Curve for problem 3

Figure 5: Efficiency Curve for problem 4

Figure 6: Efficiency Curve for problem 5
DISCUSSION:

From tables 3, 4, 5, 6, and 7 above, the comparative assessment of three numerical methods, 3-point BBDF, 3-point Extended BBDF, and the newly introduced NFDIBBD, is applied to five distinct test problems. The primary objective is to gauge the effectiveness and precision of these methods through an analysis of Maximum Errors (MAXE), the Number of Steps (NS), and Computational Time (CPU Time) for each problem. Key findings indicate that NFDIBBD consistently outperforms 3BBDF and 3ESBBDF across various step sizes in terms of maximum error and computational time, showcasing its overall superiority in solving the specified problems.

The graphical portrayal of $\log_{10} \text{MAXE}$ against $\log_{10} H$ further underscores the heightened performance of the NFDIBBD method in contrast to 3BBDF and 3ESBBDF. The downward trends in NFDIBBD's graphs signify enhanced scalability and accuracy as the step size varies.

CONCLUSION

The New Fixed coefficient Diagonally Implicit Block Backward Differentiation Formula (NFDIBBD) is developed to solve stiff initial value problems. The NFDIBBD method emerges as a resilient and effective approach for tackling stiff initial value problems, surpassing or competently matching existing methods in terms of accuracy and computational efficiency. The comprehensive analysis and visual representation reinforce the credibility of the NFDIBBD method as a valuable addition to numerical techniques for stiff ODEs.

REFERENCES


Ibrahim Z.B., Noor N.M., Othman K.I. (2019). Fixed coefficients $A()$ stable block backward differentiation formulas for stiff ordinary differential equations. Symmetry. 11, 864. [Crossref]

Ibrahim Z.B., Othman K.I., & Suleiman M.B. (2007a). Implicit $r$-point Block Backward Differentiation Formula for first order stiff ODEs. Applied Mathematics and Computational, 186:558-565. [Crossref]


