A New Numerical Method for Solving Stiff Initial Value Problems

Babangida B, Musa H, Ibrahim L

Department of Mathematics and Computer Sciences, Faculty of Natural and Applied Sciences, Umaru Musa Yar’adua University Katsina, Nigeria

Abstract

A new numerical method that computes 2-points simultaneously at each step of integration is derived. The numerical scheme is achieved by modifying an existing Di2BBDF method. The method is of order 2. The stability analysis of the new method indicates that it is both zero and A-stable, implying that it is suitable for stiff problems. The necessary and sufficient conditions for the convergence of the method are also established which proved the convergence of the method. Numerical results show that the method outperformed some existing algorithms in terms of accuracy.

Keywords: A-Stability; Order of a block method; Implicit block method; Stiff initial value problems; Convergence; Diagonally implicit; Zero stability

Introduction

Consider a system of first order stiff initial value problems (IVPs) of the form:

\[ y' = f(x, y) \quad x \in [a, b] \quad y(x_0) = y_0 \]  

System (1) can be regarded as stiff if its exact solution contains very fast as well as very slow components [1]. Stiff IVPs occur in any fields of engineering and physical sciences. They are particularly found in electrical circuits, vibrations, chemical reactions, kinetics, automatic control and combustion, theory of fluid mechanics etc. The solution is characterized by the presence of transient and steady state components, which restrict the step size of many numerical methods except methods with A-stability properties (Suleiman [2, 3]). This behaviour makes it difficult to develop suitable methods for solving stiff problems. However, efforts have been made by researchers, such as Abasi [4], Alt [5], Alvarez [6], Cash [7], Dahlquist [1], Ibrahim [8-10], Musa [11-14], Suleiman [2, 3], Yatim [15] and Zawawi [16] among others, to develop methods for stiff ODEs. The need to obtain an efficient numerical approximation in terms of accuracy and computational time has attracted some researchers such as Alexander [17] with diagonally implicit Runge-Kutta for stiff ODEs, Ababneh [18] with design of new diagonally implicit Runge-Kutta for stiff problems, Ismail [19] with embedded pair of diagonally implicit Runge-Kutta for solving ODEs, Zawawi [20] with diagonally implicit block backward differentiation formulas for solving ODEs. The motivation of this research is to modify the method developed by Zawawi [20] so as to improve its accuracy and stability properties.

Derivation of the Method

This section describes the derivation of the method. Consider the numerical scheme developed by Zawawi [20],

\[ \sum_{j=0}^{1} \alpha_{j} y_{n+k+j} = h \beta_{k} f_{n+k}, \quad k = i = 1, 2 \]  

To improve its accuracy and stability, the term \(-h \beta_{k} f_{n+k-1}\) is added to (2) to come up with new scheme as follows

\[ \sum_{j=0}^{1} \alpha_{j} y_{n+k+j} = h \beta_{k} f_{n+k} - \rho f_{n+k-1}, \quad k = i = 1, 2 \]  

Where, \(k = i = 1\) represents the first point, \(k = i = 2\) represents the second point and \(\rho \in (-1, 1)\). In this paper, the value \(\rho = \frac{1}{5}\) is used. The formula (3) is derived from Taylor’s series expansion.

A Linear operator \(L\), for the first and second point of the new method is defined by:

\[ L \left[ y(x_n), h \right] : y_{n+1} = \alpha_{1} y_{n} + \alpha_{2} y_{n+1} + \alpha_{3} y_{n+2} \]  

\[ (x_{n} + h) - h \beta_{1} f(x_n + h) - \rho \beta_{2} f(x_n + h) = 0, \quad k = i = 1 \]  

\[ L \left[ y(x_n), h \right] : y_{n+2} = \alpha_{3,1} y_{n+1} + \alpha_{3,2} y_{n+2} + \alpha_{3,3} y_{n+3} \]  

\[ (x_{n} + 2h) - h \beta_{3} f(x_n + 2h) - \rho \beta_{4} f(x_n + h) = 0, \quad k = i = 2 \]  

respectively.

Expanding (4) and (5) as Taylor’s series about \(x_n\), collect like terms and normalized the coefficient of the first point \(\alpha_{3,1}\) and second point \(\alpha_{3,2}\) to obtain the following implicit 2-point block formula:

\[ y_{n+1} = \frac{3}{7} y_{n+1} - \frac{10}{7} y_{n} - \frac{1}{7} f_{n} - \frac{5}{7} f_{n+1} \]  

\[ y_{n+2} = \frac{11}{53} y_{n+2} - \frac{51}{53} y_{n+1} + \frac{93}{53} y_{n} - \frac{195}{53} f_{n+2} + \frac{66}{53} f_{n+1} \]  

The error constant of the new method (6) is \(E_{n} = \left\{ \frac{11}{42} \right\}\) implying that is of order 2.

Throughout this paper, the method will be referred to New Diagonally Implicit Super Class of Block Backward Differentiation Formula (NDISBBDF).

Stability Analysis of the NDISBBDF

This section presents the stability analysis of the method (6). It begins by presenting the definition of zero and A-stability taken from Suleiman [2].

Definition 3.1: A linear multistep method (LMM) is said to be zero stable if no root of the first characteristics polynomial has modulus greater than one and that any root with modulus one is simple.
\textbf{Definition 3.2}: A linear multistep method (LMM) is said to be A-stable if its stability region covers the entire negative half-plane.

Formula (6) can be written in matrix form as follows
\begin{equation}
\begin{pmatrix}
1 & 0 \\
\frac{93}{37} & 1 \\
\frac{93}{55} & \frac{10}{37} \\
\frac{5}{7} & \frac{5}{37} \\
0 & \frac{5}{37} \\
\frac{5}{37} & \frac{5}{37} \\
\end{pmatrix}
\times
\begin{pmatrix}
Y_{n} \\
\frac{93}{37} Y_{n} - \frac{93}{55} Y_{n-1} + \frac{10}{37} Y_{n-2} \\
\frac{5}{7} Y_{n} - \frac{5}{37} Y_{n-1} + \frac{5}{37} Y_{n-2} \\
0 \\
\frac{5}{37} Y_{n} \\
\frac{5}{37} Y_{n} \\
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
\frac{93}{37} (1 + \lambda h) - \frac{93}{55} (1 + \lambda h) - \frac{10}{37} (1 + \lambda h) \\
\frac{5}{7} (1 + \lambda h) - \frac{5}{37} (1 + \lambda h) + \frac{5}{37} (1 + \lambda h) \\
1 \\
\frac{5}{37} \\
\frac{5}{37} \\
\end{pmatrix}
\times
\begin{pmatrix}
Y_{n} \\
Y_{n-1} \\
Y_{n-2} \\
0 \\
Y_{n} \\
Y_{n} \\
\end{pmatrix}
\tag{7}
\end{equation}

Equation (7) can be rewritten in the following form:
\[ A_{h} Y_{n} = A_{h} Y_{n-1} + h(B_{h} F_{n} + B_{h} F_{n-1}). \tag{8} \]

The boundary of the stability region of (6) is determined by substituting $\lambda = 0$ into (11). The graph of the stability region for (6) using maple is given in Figure 1.

Definition 4.1: Method (6) is consistent if and only if the following conditions are satisfied:
\[ \sum_{j=0}^{3} D_{j} = 0, \tag{15} \]
\[ \sum_{j=0}^{3} J_{j} = \sum_{j=0}^{3} G_{j}, \tag{16} \]

Where, $J_{j}$, and $G_{j}$ are defined above.

Equation (15) and (16) then become
\[ \sum_{j=0}^{3} D_{j} = D_{0} + D_{1} + D_{2} + D_{3} = \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
\end{pmatrix}. \tag{17} \]

Thus, the consistency conditions in (15) and (16) are therefore met. Hence, method (6) is consistent.

Since the method (6) is both consistent and zero stable, it is thus convergent.

**Implementation of the Method**

This section discussed the implementation of the method using Newton iteration and begin by defining the absolute and maximum error.

**Definition 5.1**: Let $y(x)$ and $y(x)$ be the approximate and exact solution of (1) respectively. Then the absolute error is given by
\[ (error) = |y(x) - y(x)|. \tag{19} \]

The maximum error is given by
\[ MAXE = \max_{1 \leq i \leq T} \left( \frac{\text{error}_{i}}{N} \right). \tag{20} \]

Where, $T$ is the total number of steps and $N$ is the number of equations.
Define;
\[ F_i = y_{n+1} + \frac{1}{7} h f_{e_{i+1}} - \frac{2}{7} h f_{e_i} - e_i \]
\[ F_2 = -\frac{93}{53} y_{n+1} + y_{n+2} + \frac{6}{56} h f_{e_{i+1}} - \frac{30}{53} h f_{e_i} - e_2 . \]

Where,
\[ e_i = -\frac{3}{53} y_{n+1} + \frac{10}{53} y_n \quad \text{and} \quad e_2 = \frac{11}{53} y_{n+1} - \frac{51}{53} y_n . \]

Let \( y^{(i+1)}_{e_i} \) denote the \((i+1)\)th iteration and

\[ y_{e_{i+j}}^{(i+1)} = y_{e_{i+j}}^{(i)} - \frac{1}{\omega} [F_j(y_{e_{i+j}}^{(i)})] \]

This can be written in the form:

\[ \left[ F_j\left(y_{e_{i+j}}^{(i)}\right)\right] y^{(i+1)}_{e_{i+j}} = -[F_j(y_{e_{i+j}}^{(i)})] . \]

Newton’s iteration for the new method takes the form:

\[ F_j\left(y_{e_{i+j}}^{(i+1)}\right) y^{(i+1)}_{e_{i+j+1}} = -[F_j(y_{e_{i+j}}^{(i+1)})] \]

In addition, in matrix form, equation (25) is equivalent to

\[ \begin{pmatrix} 1 & 0 & 0 & \frac{1}{\omega} \end{pmatrix} \begin{pmatrix} y_{e_1}^{(i+1)} \ y_{e_2}^{(i+1)} \ y_{e_3}^{(i+1)} \ y_{e_4}^{(i+1)} \end{pmatrix} = \begin{pmatrix} 0 \ 0 \ 0 \ 0 \end{pmatrix} \]

\[ \begin{pmatrix} 1 & 0 & 0 & \frac{1}{\omega} \end{pmatrix} \begin{pmatrix} y_{e_1}^{(i+1)} \ y_{e_2}^{(i+1)} \ y_{e_3}^{(i+1)} \ y_{e_4}^{(i+1)} \end{pmatrix} = \begin{pmatrix} 0 \ 0 \ 0 \ 0 \end{pmatrix} \]

Test Problems

The following problems are used to test the performance of the method.

**Problem 1** (Musa [13])

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

Exact solution:

\[ y_1(x) = e^{-10x} + e^x, \]
\[ y_2(x) = e^{-10x} - e^x \]

Eigenvalues: -1 and -39.

**Problem 2**

\[ y_1' = -y_1 - 95y_2, \quad y_1(0)=1, \quad 0 \leq x \leq 10 \]
\[ y_2' = -y_1 - 97y_2, \quad y_2(0)=1 \]

Exact solution:

\[ y_1(x) = \frac{1}{47}(95e^{-2x} - 48e^{-x}) , \]
\[ y_2(x) = \frac{1}{47}(48e^{-96x} - e^{-2x}) \]

Eigenvalues: -2 and -96.

**Problem 3**

\[ y_1' = y_2, \quad y_1(0)=1, \quad 0 \leq x \leq 10 \]
\[ y_2' = -200y_1 - 20y_2, \quad y_2(0)=-10 \]

Exact solution:

\[ y_1(x) = e^{-10x}\cos10x, \]
\[ y_2(x) = -10e^{-10x}\cos10x+\sin10x. \]

**Discussion**

From tables 1-3, it can be seen that the new method outperformed the existing 2-point diagonally implicit block backward differentiation formula in terms of accuracy. Convergence is evident by the decrease in error as the step length \( h \) tends to zero. Similarly, the solution at any fixed point improves as the step length reduce. This can be seen from the tables when \( h \) is reduced (from 0.01, 0.001, and 0.0001 to 0.00001). The maximum error indicates that the numerical result becomes closer to the exact solution. Thus, the computed solution tends to the exact solution as the step length tends to zero. Hence, the new method converges faster for all the problems tested in comparison with DI2BBDF.

<table>
<thead>
<tr>
<th>( h )</th>
<th>Method</th>
<th>NS</th>
<th>MAXE</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^-6</td>
<td>DI2BBDF</td>
<td>1000</td>
<td>6.85453e-002</td>
<td>2.47400e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>1000</td>
<td>7.15278e-002</td>
<td>1.57000e-001</td>
</tr>
<tr>
<td>10^-5</td>
<td>DI2BBDF</td>
<td>1000</td>
<td>2.60436e-002</td>
<td>1.58700e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>1000000</td>
<td>2.32062e-003</td>
<td>1.60900e-001</td>
</tr>
<tr>
<td>10^-4</td>
<td>DI2BBDF</td>
<td>1000000</td>
<td>2.86751e-005</td>
<td>3.23700e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>1000000</td>
<td>2.74330e-007</td>
<td>9.01200e-001</td>
</tr>
<tr>
<td>10^-3</td>
<td>DI2BBDF</td>
<td>10000000</td>
<td>2.87419e-005</td>
<td>2.47190e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>10000000</td>
<td>2.75064e-009</td>
<td>1.54470e-001</td>
</tr>
</tbody>
</table>

**Table 1:** Numerical result for problem 1.

<table>
<thead>
<tr>
<th>( h )</th>
<th>Method</th>
<th>NS</th>
<th>MAXE</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^-6</td>
<td>DI2BBDF</td>
<td>500</td>
<td>9.37034e+005</td>
<td>2.51600e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>5000</td>
<td>2.22919e-002</td>
<td>1.75400e-01</td>
</tr>
<tr>
<td>10^-5</td>
<td>DI2BBDF</td>
<td>5000</td>
<td>5.58100e-002</td>
<td>2.43700e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>50000</td>
<td>1.20900e-002</td>
<td>1.33100e-001</td>
</tr>
<tr>
<td>10^-4</td>
<td>DI2BBDF</td>
<td>500000</td>
<td>7.04582e-003</td>
<td>2.61300e-001</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>500000</td>
<td>1.61924e-004</td>
<td>3.05800e-001</td>
</tr>
<tr>
<td>10^-3</td>
<td>DI2BBDF</td>
<td>5000000</td>
<td>7.19560e-004</td>
<td>6.85100e+000</td>
</tr>
<tr>
<td></td>
<td>NDISBBDF</td>
<td>5000000</td>
<td>1.58717e-001</td>
<td>2.51800e+001</td>
</tr>
</tbody>
</table>

**Table 2:** Numerical result for problem 2.

**Numerical Results**

The numerical results for the test problems given in section 6 are tabulated in this section. The problems are solved using the new method developed and the existing 2-point diagonally implicit block backward differentiation formula developed by Zawawi [20]. The number of steps taken to complete the integration and the maximum error for the methods are presented and compared in Tables 1-3. In addition, the graph of \( \log_{10}(\text{MAXE}) \) against \( h \) for each problem is plotted. The notations used in the tables are listed below:

- DI2BBDF=Diagonally Implicit 2-Point Block Backward Differentiation Formula
- NDISBBDF=New Diagonally Implicit Super Class of Block Backward Differentiation Formula
- NS=Total Number of Steps
- MAXE=Maximum Error
- Time=Computational Time in Seconds
- h=Step Size
The graphs also show that the scaled errors for the new method are smaller when compared with that in the existing method.

**Conclusion**

A new method called New Diagonally Implicit Super Class of Block Backward Differentiation Formula (NDISBBDF) is developed. The order of the method is 2 and it is suitable for solving stiff IVPs. The stability analysis has shown that the method is both zero and A-stable. A comparison between the method and existing DI2BBDF is made and the results show that the method outperformed the existing DI2BBDF method in terms of accuracy.

**References**


---

<table>
<thead>
<tr>
<th>$h$</th>
<th>Method</th>
<th>NS</th>
<th>MAXE</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>DI2BBDF</td>
<td>500</td>
<td>1.61797e+000</td>
<td>1.17000e-001</td>
</tr>
<tr>
<td></td>
<td>DIS2BBDF</td>
<td>500</td>
<td>1.63063e-001</td>
<td>1.19600e-001</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>DI2BBDF</td>
<td>50000</td>
<td>1.45914e-001</td>
<td>1.09700e-001</td>
</tr>
<tr>
<td></td>
<td>DIS2BBDF</td>
<td>50000</td>
<td>3.27724e-003</td>
<td>1.50200e-001</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>DI2BBDF</td>
<td>500000</td>
<td>1.44486e-002</td>
<td>2.22200e-001</td>
</tr>
<tr>
<td></td>
<td>DIS2BBDF</td>
<td>500000</td>
<td>3.56827e-005</td>
<td>3.46700e-001</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>DI2BBDF</td>
<td>5000000</td>
<td>1.44348e-003</td>
<td>7.39800e-001</td>
</tr>
<tr>
<td></td>
<td>DIS2BBDF</td>
<td>5000000</td>
<td>3.60983e-007</td>
<td>2.89700e+000</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>DI2BBDF</td>
<td>50000000</td>
<td>1.44332e-004</td>
<td>5.54800e+000</td>
</tr>
<tr>
<td></td>
<td>DIS2BBDF</td>
<td>50000000</td>
<td>3.61514e-009</td>
<td>2.59900e+001</td>
</tr>
</tbody>
</table>

**Table 3:** Numerical result for problem 3.