Fluid Mechanics: Open Access

Babangida et al., Fluid Mech Open Acc 2016, 3:2
DOI: 10.4172/2476-2296.1000136

Research Article Open Access

A New Numerical Method for Solving Stiff Initial Value Problems

[1]B. Babangida*, [2]H. Musa and [3]L. K. Ibrahim

[1,2,3]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:

$$y' = \mathbf{f}(x, \mathbf{y}) \quad x \in [a,b] \quad y(x_0) = y_0 \tag{1}$$

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 have 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].

$$\sum_{j=0}^{1+k} \alpha_{j,i} y_{n+j-1} = h\beta_{k,i} f_{n+k}, k = i = 1,2 \tag{2}$$

To improve its accuracy and stability, the term $-h\beta(k,i)\,\rho f(n+k-1)$ is added to (2) to come up with new scheme as follows

$$\sum_{j=0}^{1+k} \alpha_{j,i} y_{n+j-1} = h\beta_{k,i}\left(f_{n+k} - \rho f_{n+k-1}\right), k = i = 1,2 \tag{3}$$

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_i for the first and second point of the new method is defined by:

$$L_i\left[y(x_n),h\right]: \alpha_{0,i} y(x_n - h) + \alpha_{1,i} y(x_n) + \alpha_{2,i} y(x_n + h) - h\beta_{k,i}\left(f(x_n + h) - \rho f(x_n)\right) = 0, \quad k = i = 1. \tag{4}$$

$$L_i\left[y(x_n),h\right]: \alpha_{0,i} y(x_n - h) + \alpha_{1,i} y(x_n) + \alpha_{2,i} y(x_n + h) + \alpha_{3,i} y(x_n + 2h) - h\beta_{k,i}\left(f(x_n + 2h) - \rho f(x_n + h)\right) = 0, \ k = i = 2 \tag{5}$$

respectively.

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

$$y_{n+1} = -\frac{3}{7} y_{n-1} + \frac{10}{7} y_n - \frac{1}{7} hf_n + \frac{5}{7} hf_{n+1} \tag{6}$$

$$y_{n+2} = \frac{11}{53} y_{n-1} - \frac{51}{53} y_n + \frac{93}{53} y_{n+1} - \frac{6}{53} hf_{n+1} + \frac{30}{53} hf_{n+2}.$$

The error constant of the new method (6) is $E_3 = \begin{pmatrix} -\frac{11}{42} \\ 0 \end{pmatrix}$ 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.

***Corresponding author:** B. Babangida, Department of Mathematics and Computer Sciences, Faculty of Natural and Applied Sciences, Umaru Musa Yar'adua University Katsina, Katsina State, Nigeria, Tel: +2347067704150; E-mail: bature.babangida@umyu.ed.ng

Received September 25, 2016; **Accepted** October 23, 2016; **Published** October 30, 2016

Citation: Babangida B, Musa H, Ibrahim L. K. (2016) A New Numerical Method for Solving Stiff Initial Value Problems. Fluid Mech Open Acc 3: 136. doi: 10.4172/2476-2296.1000136

Citation: Babangida B, Musa H, Ibrahim L. K. (2016) A New Numerical Method for Solving Stiff Initial Value Problems. Fluid Mech Open Acc 3: 136. doi: 10.4172/2476-2296.1000136

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

$$\begin{pmatrix} 1 & 0 \\ -\frac{93}{53} & 1 \end{pmatrix}\begin{pmatrix} y_{n+1} \\ y_{n+2} \end{pmatrix}=\begin{pmatrix} -\frac{3}{7} & \frac{10}{7} \\ \frac{11}{53} & -\frac{51}{53} \end{pmatrix}\begin{pmatrix} y_{n-1} \\ y_n \end{pmatrix}+h\begin{pmatrix} 0 & -\frac{1}{7} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} f_{n-1} \\ f_n \end{pmatrix}+h\begin{pmatrix} \frac{5}{7} & 0 \\ -\frac{6}{53} & \frac{30}{53} \end{pmatrix}\begin{pmatrix} f_{n+1} \\ f_{n+2} \end{pmatrix} \quad (7)$$

Equation (7) can be rewritten in the following form:

$$A_0Y_m = A_1Y_{m-1}+h(B_0F_{m1}+B_1F_m), \quad (8)$$

Where,

$$A_0=\begin{pmatrix} 1 & 0 \\ -\frac{93}{53} & 1 \end{pmatrix},\ A_1=\begin{pmatrix} -\frac{3}{7} & \frac{10}{7} \\ \frac{11}{53} & -\frac{51}{53} \end{pmatrix},\ B_0=\begin{pmatrix} 0 & -\frac{1}{7} \\ 0 & 0 \end{pmatrix},B_1=\begin{pmatrix} \frac{5}{7} & 0 \\ -\frac{6}{53} & \frac{30}{53} \end{pmatrix},\ Y_m=\begin{pmatrix} y_{n+1} \\ y_{n+2} \end{pmatrix}=\begin{pmatrix} y_{2m+1} \\ y_{2m+2} \end{pmatrix},$$

$$Y_{m-1}=\begin{pmatrix} y_{n-1} \\ y_n \end{pmatrix}=\begin{pmatrix} y_{2m-1} \\ y_{2m} \end{pmatrix}=\begin{pmatrix} y_{2(m-1)+1} \\ y_{2(m-1)+2} \end{pmatrix},\ F_{m-1}=\begin{pmatrix} f_{n-1} \\ f_n \end{pmatrix}=\begin{pmatrix} f_{2m-1} \\ f_{2m} \end{pmatrix}=\begin{pmatrix} f_{2(m-1)+1} \\ f_{2(m-1)+2} \end{pmatrix},\ F_m=\begin{pmatrix} f_{n+1} \\ f_{n+2} \end{pmatrix}=\begin{pmatrix} f_{2m+1} \\ f_{2m+2} \end{pmatrix}.$$

Substituting the scalar test equation $y'=\lambda y$ ($\lambda<0, \lambda$ is complex) into (8) and using $\lambda h=\bar{h}$ gives

$$A_0Y_m = A_1Y_{m-1}+\bar{h}(B_0Y_{m-1}+B_1Y_m) \quad (9)$$

The stability polynomial of (6) is obtained by evaluating:

$$Det\left[(A_0-\bar{h}B_1)t-(A_1+\bar{h}B_0)\right]=0 \quad (10)$$

to get,

$$R\left(t,\bar{h}\right)=t^2-\frac{475}{371}t^2\bar{h}-\frac{414}{371}t+\frac{150}{371}\bar{h}^2t^2-\frac{192}{371}t\bar{h}+\frac{43}{371}-\frac{6}{371}t\bar{h}^2+\frac{11}{371}\bar{h}=0 \quad (11)$$

To show that the method (6) is zero stable, we set $\bar{h}=0$ in (11) to get the first characteristics polynomial as follows:

$$t^2-\frac{414}{371}t+\frac{43}{371}=0 \quad (12)$$

Solving equation (12) for t gives the following roots:

t=0.1159 and t=1 (13)

From the definition 3.1, method (6) is zero-stable.

The boundary of the stability region of (6) is determined by substituting $t=e^{i\theta}$ into (11). The graph of stability region for (6) using maple is given in Figure 1.

The stability region covers the entire negative half plane indicating that the method (6) is A-stable.

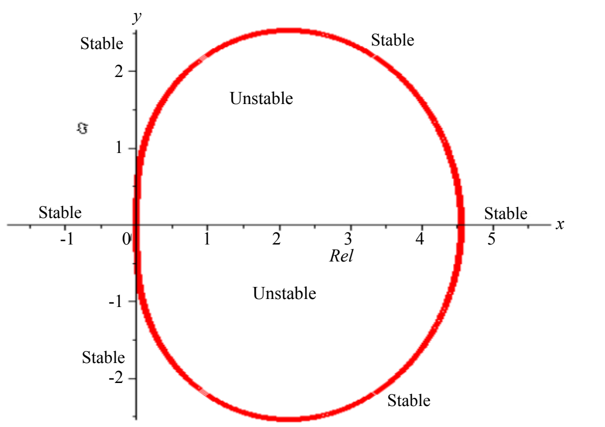

Figure 1: Stability region of the method.

Convergence of the Method

Convergence is an essential feature that every acceptable linear multistep method (LMM) must possess. This section discussed the convergence of the method (6). Consistency and zero stability are the necessary and sufficient conditions for the convergence of any numerical scheme. In section 3, it was shown that method (6) is zero stable. It is now remain to show that method (6) is consistent.

This discussion will be based on matrix form of (6) which can be written as:

$$\begin{pmatrix} \frac{3}{7} & -\frac{10}{7} \\ -\frac{11}{53} & \frac{51}{53} \end{pmatrix}\begin{pmatrix} \mathrm{y}_{n-1} \\ \mathrm{y}_n \end{pmatrix}+\begin{pmatrix} 1 & 0 \\ -\frac{93}{53} & 1 \end{pmatrix}\begin{pmatrix} \mathrm{y}_{n+1} \\ \mathrm{y}_{n+2} \end{pmatrix}=\mathrm{h}\begin{pmatrix} 0 & -\frac{1}{7} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathrm{f}_{n-1} \\ \mathrm{f}_n \end{pmatrix}+\mathrm{h}\begin{pmatrix} 0 & -\frac{1}{7} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathrm{f}_{n-1} \\ \mathrm{f}_n \end{pmatrix} \quad (14)$$

With $\mathrm{D}_{0,}=\begin{pmatrix} \frac{3}{7} \\ -\frac{11}{53} \end{pmatrix}$,

$$\mathrm{D}_1=\begin{pmatrix} -\frac{10}{7} \\ \frac{51}{53} \end{pmatrix},\mathrm{D}_2=\begin{pmatrix} 1 \\ -\frac{93}{53} \end{pmatrix},\ \mathrm{D}_3=\begin{pmatrix} 0 \\ 1 \end{pmatrix},\mathrm{G}_0=\begin{pmatrix} 0 \\ 0 \end{pmatrix},\ \mathrm{G}_1=\begin{pmatrix} -\frac{1}{7} \\ 0 \end{pmatrix},\mathrm{G}_2=\begin{pmatrix} \frac{5}{7} \\ -\frac{6}{53} \end{pmatrix},\ \mathrm{G}_3=\begin{pmatrix} 0 \\ \frac{30}{53} \end{pmatrix}.$$

Definition 4.1: Method (6) is consistent if and only if the following conditions are satisfied:

$$\sum_{j=0}^{3}D_j=0\ , \quad (15)$$

$$\sum_{j=0}^{3}jD_j=\sum_{j=0}^{3}G_j \quad (16)$$

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

Equation (15) and (16) then become

$$\sum_{j=0}^{3}D_j=D_0+D_1+D_2+D_3=\begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (17)$$

$$\sum_{j=0}^{3}jD_j=\sum_{j=0}^{3}G_j=\begin{pmatrix} \frac{4}{7} \\ \frac{24}{53} \end{pmatrix}. \quad (18)$$

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 converges.

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_i and $y(x_i)$ be the approximate and exact solution of (1) respectively. Then the absolute error is given by

$$(error_i)_t=|(y_i)_t-(y(x_i))_t| \quad (19)$$

The maximum error is given by

$$MAXE=\max_{1\le i\le T}\underbrace{\left(\max\left(error_i\right)_t\right)}_{1\le i\le N} \quad (20)$$

Where, T is the total number of steps and N is the number of equations.

Citation: Babangida B, Musa H, Ibrahim L. K. (2016) A New Numerical Method for Solving Stiff Initial Value Problems. Fluid Mech Open Acc 3: 136. doi: 10.4172/2476-2296.1000136

Define;

$$F_1 = y_{n+1} + \frac{1}{7}hf_n - \frac{5}{7}hf_{n+1} - \varepsilon_1$$

$$F_2 = -\frac{93}{53}y_{n+1} + y_{n+2} + \frac{6}{56}hf_{n+1} - \frac{30}{53}hf_{n+2} - \varepsilon_2 \cdot$$

Where,

$$\varepsilon_1 = -\frac{3}{7}y_{n-1} + \frac{10}{7}y_n \text{ and } \varepsilon_2 = \frac{11}{53}y_{n-1} - \frac{51}{53}y_n .$$

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

$$e_{n+j}^{(i+1)} = y_{n+j}^{(i+1)} - y_{n+j}^{(i)}, \text{ j=1, 2.} \tag{22}$$

$$e_{n+j}^{(i+1)} = y_{n+j}^{(i+1)} - y_{n+j}^{(i)}, \left[F_j\left(y_{n+j}^{(i)}\right)\right] \tag{23}$$

This can be written in the form:

$$[\left[F_j^{'}\left(y_{n+j}^{(i)}\right)\right]e_{n+j}^{(i+1)} = -[F_j(y_{n+j}^{(i)})]]. \tag{24}$$

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

$$\left[F_j^{'}\left(y_{n+1,n+2}^{(i)}\right)\right]e_{n+1,n+2}^{(i+1)} = -[F_j(y_{n+1,n+2}^{(i)})] \tag{25}$$

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

$$\begin{pmatrix} 1-\frac{5}{7}h\frac{\partial f_{n+1}}{\partial y_{n+1}} & 0 \\ -\frac{93}{53}+\frac{6}{53}h\frac{\partial f_{n+1}}{\partial y_{n+1}} & 1-\frac{30}{53}h\frac{\partial f_{n+2}}{\partial y_{n+2}} \end{pmatrix}\begin{pmatrix} e_{n+1}^{(i+1)} \\ e_{n+2}^{(i+1)} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ \frac{93}{53} & -1 \end{pmatrix}\begin{pmatrix} y_{n+1}^{(i)} \\ y_{n+2}^{(i)} \end{pmatrix} + h\begin{pmatrix} 0 & -\frac{1}{7} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} f_{n-1}^{(i)} \\ f_n^{(i)} \end{pmatrix} + h\begin{pmatrix} \frac{5}{7} & 0 \\ -\frac{6}{53} & \frac{30}{53} \end{pmatrix}\begin{pmatrix} f_{n+1}^{(i)} \\ f_{n+2}^{(i)} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} \tag{26}$$

Test Problems

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

Problem 1 (Musa [13])

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

$y_2' = -19y_1 - 20y_2,\ y_2(0)=0$

Exact solution: $y_1(x) = e^{-39x} + e^{-x}$,

$y_2(x) = e^{-39x} - e^{-x}$

Eigenvalues: -1 and -39.

Problem 2

$y_1' = -y_1 - 95y_2,\ y_1(0)=1,\ 0 \le x \le 10$

$y_2' = -y_1 - 97y_2,\ y_2(0)=1$

Exact solution: $y_1(x) = \frac{1}{47}\left(95e^{-2x} - 48e^{-2x}\right)$,

$y_2(x) = \frac{1}{47}\left(48e^{-96x} - e^{-2x}\right)$

Eigenvalues: – 2 and – 96.

Problem 3

$y_1' = y_2,\ y_1(0)=1,\ 0 \le x \le 10$

$y_2' = -200y_1 - 20y_2.\ y_2(0) = -10$

Exact solution: $y_1(x) = e^{-10x}\cos 10x$,

$y_2(x) = -10e^{-10x}(\cos 10x + \sin 10x)$

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

To give the visual impact on the performance of the new method, the graphs of Log10 (MAXE) against h for the problems tested are plotted in figures 2-4.

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., 0.0001, and 0.00001 to 0.000001). 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.

h	Method	NS	MAXE	Time
10^{-2}	DI2BBDF NDISBBDF	1000 1000	6.85453e-002 7.15278e-002	2.47400e-001 1.57000e-001
10^{-3}	DI2BBDF NDISBBDF	1000 1000	2.60436e-002 2.32062e-003	1.58700e-001 1.609000e-001
10^{-4}	DI2BBDF NDISBBDF	100000 100000	2.84730e-003 2.68751e-005	3.23700e-001 9.01200e-001
10^{-5}	DI2BBDF NDISBBDF	1000000 1000000	2.87174e-004 2.74330e-007	1.14900e+000 5.11600e+000
10^{-6}	DI2BBDF NDISBBDF	10000000 10000000	2.87419e-005 2.75064e-009	9.85100e+000 5.44700e+001

Table 1: Numerical result for problem 1.

h	Method	NS	MAXE	Time
10^{-2}	DI2BBDF NDISBBDF	500 500	9.37034e+005 2.22919e-002	2.51600e-001 1.75400e-001
10^{-3}	DI2BBDF NDISBBDF	5000 5000	5.58180e-002 1.20098e-002	2.43700e-001 1.331000e-001
10^{-4}	DI2BBDF NDISBBDF	50000 50000	7.04562e-003 1.61824e-004	2.61300e-001 3.05800e-001
10^{-5}	DI2BBDF NDISBBDF	500000 500000	7.19659e-004 1.69025e-006	7.92400e-001 2.3500e+000
10^{-6}	DI2BBDF NDISBBDF	5000000 5000000	7.21171e-005 1.70019e-008	5.05000e+000 2.58100e+001

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} (*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

Citation: Babangida B, Musa H, Ibrahim L. K. (2016) A New Numerical Method for Solving Stiff Initial Value Problems. Fluid Mech Open Acc 3: 136. doi: 10.4172/2476-2296.1000136

h	Method	NS	MAXE	Time
10^{-2}	DI2BBDF	500	1.61797e+000	1.17000e-001
	DIS2BBDF	500	1.63063e-001	1.19600e-001
10^{-3}	DI2BBDF	5000	1.45914e-001	1.09700e-001
	DIS2BBDF	5000	3.27724e-003	1.50200e-001
10^{-4}	DI2BBDF	50000	1.44486e-002	2.22200e-001
	DIS2BBDF	50000	3.56827e-005	3.46700e-001
10^{-5}	DI2BBDF	500000	1.44346e-003	7.39800e-001
	DIS2BBDF	500000	3.60983e-007	2.89700e+000
10^{-6}	DI2BBDF	5000000	1.44332e-004	5.54800e+000
	DIS2BBDF	5000000	3.61514e-009	2.559000e+001

Table 3: Numerical result for problem 3.

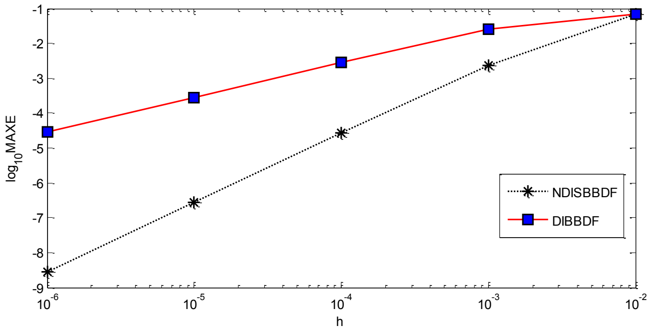

Figure 2: Graph of Log_{10} (*MAXE*) against h for problem 1.

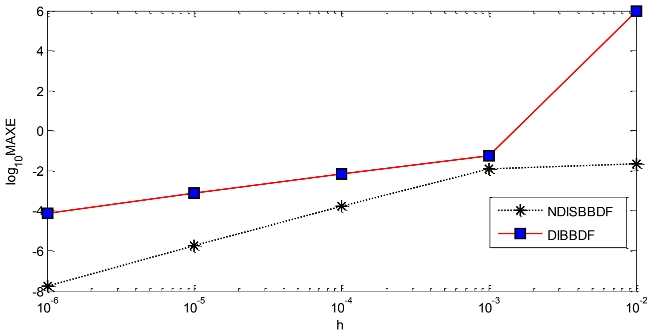

Figure 3: Graph of Log_{10} (*MAXE*) against h for problem 2.

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

1. Dahlquist G (1974) Problems related to the numerical treatment of stiff differential equations.
2. Suleiman MB, Musa H, Ismail F, Senu N, Ibrahim ZB (2014) A new super class of block backward differentiation formulas for stiff ODEs. Asian -European J Math.

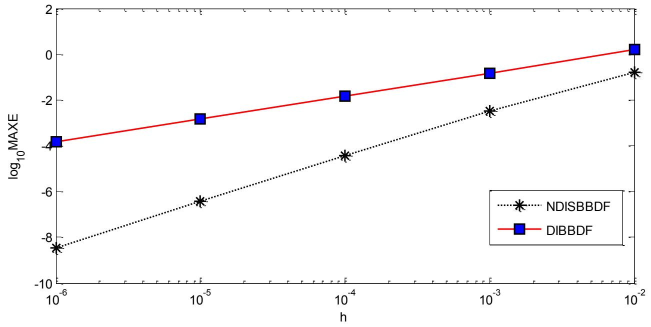

Figure 4: Graph of Log_{10} (*MAXE*) against h for problem 3.

3. Suleiman MB, Musa H, Ismail F, Senu N (2013) A new variable step size block backward differentiation formula for solving stiff IVPs. Int J Comput Math 90: 2391-2408.
4. Abasi N, Suleiman MB, Abbasi N, Musa H (2014) 2-point block BDF method with off-step points for solving stiff ODEs. Journal of Soft Computing and Applications 2014: 1-15.
5. Alt R (1978) A-stable one-step methods with step-size control for stiff systems of ordinary differential equations. J Comput Appl Math 4: 29-35.
6. Alvarez J, Rojo J (2002) An improved class of generalized Runge-Kutta methods for stiff problems. Part I: The scalar case. Appl Math Comput 130: 537-560.
7. Cash JR (1980) On the integration of stiff systems of ODEs using extended backward differentiation formulae. Numer Math 34: 235-246.
8. Ibrahim ZB, Othman KI, Suleiman MB (2008) Fixed coefficients block backward differentiation formulas for the numerical solution of stiff ordinary differential equations. Eur J Sci Res 21: 508-520.
9. Ibrahim ZB, Othman KI, Suleiman MB (2007) Implicit r-point block backward differentiation formula for first order stiff ODEs. Appl Math Comput 186: 558-565.
10. Ibrahim ZB, Isk K, Othman A, Suleiman MB (2007) Variable step block backward differentiation formula for first order stiff ODEs. Proceedings of the World Congress on Engineering 2: 2-6.
11. Musa H, Suleiman MB, Ismail F, Senu N, Ibrahim ZB (2013) An accurate block solver for stiff IVPs. ISRN Applied mathematics.
12. Musa H, Suleiman MB, Ismail F (2015) An implicit 2-point block extended backward differentiation formulas for solving stiff IVPs. Malaysian J Math Sci 9: 33-51.
13. Musa H, Suleiman MB, Senu N (2012) A-stable 2-point block extended backward differentiation formulas for solving stiff ODEs. AIP Conference Proceedings 1450: 254-258.
14. Musa H, Suleiman MB, Ismail F, Senu N, Ibrahim ZB (2013) An improved 2-point block backward differentiation formula for solving stiff initial value problems. AIP Conference Proceedings 1522: 211 -220.
15. Yatim SAM, Ibrahim ZB, Othman KI, Suleiman MB (2011) A quantitative comparison of numerical method for solving stiff ordinary differential equations. Math Probl Eng 2011: 193691.
16. Zawawi ISM, Ibrahim ZB, Othman KI (2015) Derivation of diagonally implicit block backward differentiation formulas for solving stiff IVPs. Math Probl Eng 2015: 179231.
17. Alexander R (1977) Diagonally implicit Runge-Kutta for stiff ordinary differential equations. SIAM J Numer Anal 14: 1006-1021.
18. Ababneh OY, Ahmad R, Ismail ES (2009) Design of new diagonally implicit Runge-Kutta method for stiff problems. Appl Math Sci 3: 2241-2253.
19. Ismail F, Al-khasawneh AR, Suleiman MB, Malik Hassan ABU (2010) Embedded pair of diagonally implicit Runge-Kutta method for solving ordinary differential equations. Sains Malaysiana 39: 1049-1054.

20. Zawawi ISM, Ibrahim ZB, Ismail F, Majid ZA (2012) Diagonally implicit block backward differentiation formulas for solving ODEs. Internat J Math Math Sci 2012: 767328.