RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2014.40.375

ScienceAsia 40 (2014): **375–378**

Variable step 2-point block backward differentiation formula for index-1 differential algebraic equations

Naghmeh Abasi[a,*], **Mohamed Bin Suleiman**[a], **Zarina Bibi Ibrahim**[b], **Hamisu Musa**[b], **Faranak Rabiei**[a]

[a] Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia
[b] Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

*Corresponding author, e-mail: naghmeh.abasiupm@yahoo.com

Received 25 Apr 2013
Accepted 20 Dec 2013

ABSTRACT: In this paper, index-1 differential algebraic equations have been solved via a block backward differentiation formula (BDF) using variable step size. Two solution values are obtained simultaneously based on the method in the block. The strategy of controlling the step size is proposed. The method is compared with the existing variable step BDF method. Numerical results are given to support the enhancement of the method in terms of accuracy.

KEYWORDS: block method, variable step size

INTRODUCTION

Differential algebraic equations (DAEs) arise in many applications of engineering and science such as power systems, circuit analysis, simulation of mechanical systems, and optimal control problems[1–3]. The most general form of a DAE is given by

$$F(x, y, y') = 0 \tag{1}$$

where $\partial F/\partial y'$ is assumed to be singular. If it is nonsingular, (1) is considered as an implicit ODE and can be reformulated as $y' = f(x, y)$.

A semi-explicit DAE or an ODE with constraints is defined as

$$\begin{aligned} y' &= f(x, y, z) \\ 0 &= g(x, y, z) \end{aligned} \tag{2}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$. This is a special case of (1). The index is 1 if $\partial g/\partial z$ is nonsingular and one differentiation of (2) yields z'. The index of a DAE is defined as the number of differentiations required to transform the DAE system to its related ODE. The unknowns, y and z, are differential variables and algebraic variables, respectively. Problems with higher index are more difficult to solve. Fortunately, most DAEs encountered in applications are of index 1 and if the problems are of higher index, they can be reduced to a combination of Hessenberg systems.

Block methods for ODEs have been developed by many researchers, e.g., see Refs. 4–7. In recent years, considerable attempts have been made to solve systems of DAEs numerically. Many numerical methods have been developed. The most common for low index are the backward differentiation formula (BDF)[1,8,9] and implicit Runge-Kutta[1,10] methods. Implicit methods for index-1 DAEs converge with the same order as for ODEs[1]. Numerical methods for solving ODEs are known to work well with DAEs. Recently, block BDF methods using constant step size for solving index-1 DAEs were presented[11]. The solution of index-1 DAEs using a block BDF of variable step size have not been considered before. This work is an extension of the work in Ref. 11 to implement a block BDF for the solution of DAEs. The aim of this paper is to obtain two solution values of index-1 DAEs simultaneously using a variable step size block BDF method. A method for ODEs is used[12]. Unlike the conventional BDF methods, the proposed method has the advantage of computing more than one solution value per step, using three previously computed back values.

BLOCK BDF METHOD

First, the block BDF method for ODEs[12] is presented and then the extension of the method for semi-explicit index-1 DAEs is explained.

Block BDF method for ODEs

In the derivation of the method for ODEs[12], three back values, namely, y_{n-2}, y_{n-1} and y_n, are used to compute two new values, y_{n+1} and y_{n+2}, simultaneously at each step. The step size between the back values and current values are qh and q, respectively, where q is the step size ratio. The step size control is organized as follows. First, a constant step size is used and a test is conducted to determine the magnitude of the local truncation error in relation to the prescribed tolerance used at each step. If the error is small enough to allow for step size increase, the step size is increased by a multiple of 1.6, otherwise it is halved. Hence, the corresponding values of q that are used for the step size changes are $q = 1$, 2, and $\frac{5}{8}$. The motivation behind the choice of each value of q is to optimize the total number of steps and make each value used give rise to a zero stable formula. For full details of the analysis of zero stability see Ref. 12.

The interpolating polynomial $P_k(x)$ of degree k which interpolates the points $(x_{n-2}, y_{n-2}), \dots, (x_{n+2}, y_{n+2})$ is defined as

$$P_k(x) = \sum_{j=0}^{k} L_{k,j}(x)\, y(x_{n+2-j}) \tag{3}$$

where

$$L_{k,j}(x) = \prod_{\substack{i=0\\ i\neq j}} \frac{x - x_{n+2-i}}{x_{n+1-j} - x_{n+2-i}} \quad j = 0, 1, \dots, k.$$

Define $s = (x - x_{n+1})/h$ and substitute $x = x_{n+1} + sh$ in the associated polynomial for (3). Hence

$$\begin{aligned} p(x) &= p(x_{n+1} + sh) \\ &= \frac{(s+1+2q)(s+1+q)(s+1)s}{4(q+1)(q+2)}\, y_{n+2} \\ &\quad - \frac{(s+1+2q)(s+1+q)(s^2-1)}{(q+1)(2q+1)}\, y_{n+1} \\ &\quad + \frac{(s+1+2q)(s+1+q)s(s-1)}{4q^2}\, y_n \\ &\quad - \frac{(s+1+2q)s(s^2-1)}{q^2(q+1)(q+2)}\, y_{n-1} \\ &\quad + \frac{(s+1+q)s(s^2-1)}{4q^2(2q+1)(q+1)}\, y_{n-2}. \end{aligned} \tag{4}$$

The polynomial (4) is differentiated with respect to s at both 0 and 1. Differentiating at $s = 0$ gives the formula for the first point and differentiating at $s = 1$ gives the formula for the second point. For further details of the derivation see Ref. 12. Substituting $q = 1, 2$ and $q = \frac{5}{8}$ in the resulting derivatives, we obtain the 2-point block formulae as follows. For $q = 1$:

$$\begin{aligned} y_{n+1} &= \tfrac{1}{10} y_{n-2} - \tfrac{3}{5} y_{n-1} + \tfrac{9}{5} y_n - \tfrac{3}{10} y_{n+2} \\ &\quad + \tfrac{6}{5} h f_{n+1} \\ y_{n+2} &= -\tfrac{3}{25} y_{n-2} + \tfrac{16}{25} y_{n-1} - \tfrac{36}{25} y_n + \tfrac{48}{25} y_{n+1} \\ &\quad + \tfrac{12}{25} h f_{n+2}. \end{aligned} \tag{5}$$

For $q = 2$:

$$\begin{aligned} y_{n+1} &= \tfrac{3}{128} y_{n-2} - \tfrac{25}{128} y_{n-1} + \tfrac{225}{128} y_n - \tfrac{75}{128} y_{n+2} \\ &\quad + \tfrac{15}{8} h f_{n+1} \\ y_{n+2} &= -\tfrac{2}{115} y_{n-2} + \tfrac{3}{23} y_{n-1} - \tfrac{18}{23} y_n + \tfrac{192}{115} y_{n+1} \\ &\quad + \tfrac{12}{23} h f_{n+2}. \end{aligned} \tag{6}$$

For $q = \frac{5}{8}$:

$$\begin{aligned} y_{n+1} &= \tfrac{208}{775} y_{n-2} - \tfrac{6912}{5425} y_{n-1} + \tfrac{13689}{6200} y_n \\ &\quad - \tfrac{351}{1736} y_{n+2} + \tfrac{117}{124} h f_{n+1} \\ y_{n+2} &= -\tfrac{12544}{29875} y_{n-2} + \tfrac{53248}{29875} y_{n-1} - \tfrac{74529}{29875} y_n \\ &\quad + \tfrac{2548}{1195} y_{n+1} + \tfrac{546}{1195} h f_{n+2}. \end{aligned} \tag{7}$$

Block BDF method for DAEs

We define the 2-point block BDF of (5), (6) and (7) for the DAE (2) as

$$\begin{aligned} y_{n+1} &= h\alpha_1 f(x_{n+1},\ y_{n+1},\ z_{n+1}) + \beta_1 y_{n+2} + \zeta_1, \\ y_{n+2} &= h\alpha_2 f(x_{n+2},\ y_{n+2},\ z_{n+2}) + \beta_2 y_{n+1} + \zeta_2, \\ 0 &= g(x_{n+1},\ y_{n+1},\ z_{n+1}), \\ 0 &= g(x_{n+2},\ y_{n+2},\ z_{n+2}), \end{aligned} \tag{8}$$

where ζ_1 and ζ_2 represent the back values for the first and second points, respectively. β_1 and β_2 represent the coefficient of y_{n+2} and y_{n+1}, respectively. α_1 is the coefficient of f_{n+1} while α_2 is the coefficient of f_{n+2}.

IMPLEMENTATION OF THE METHOD

Newton's iteration is applied for the implementation of the method. First, we define the error in the ith iteration for y and z as $\mathrm{err}^{(i)}$ which is equivalent to

$$\max(|y^{(i)}_{\text{exact}} - y^{(i)}_{\text{approx}}|, |z^{(i)}_{\text{exact}} - z^{(i)}_{\text{approx}}|)$$

and the maximum global error is given by

$$\mathrm{MAXE} = \max_{1\leqslant i\leqslant \mathrm{TNS}} (\mathrm{err}^{(i)})$$

where TNS is the total number of steps. Let $y^{(i+1)}_{n+j}$ and $z^{(i+1)}_{n+j}$, $j = 1, 2$ denote the $(i+1)$th iterative values of y_{n+j} and z_{n+j}, respectively. Define

$$e_{n+j}^{(i+1)} = y_{n+j}^{(i+1)} - y_{n+j}^{(i)},$$
$$\hat{e}_{n+j}^{(i+1)} = z_{n+j}^{(i+1)} - z_{n+j}^{(i)}, \quad j = 1, 2. \tag{9}$$

Let

$$F_1 = y_{n+1} - \beta_1 y_{n+2} - h\alpha_1 f_{n+1} - \zeta_1,$$
$$F_2 = -\beta_2 y_{n+1} + y_{n+2} - \alpha_2 h f_{n+2} - \zeta_2,$$
$$g_1 = g(x_{n+1},\ y_{n+1},\ z_{n+1}),$$
$$g_2 = g(x_{n+2},\ y_{n+2},\ z_{n+2}).$$

Newton's iteration then takes the form

$$\begin{bmatrix} y_{n+j}^{(i+1)} \\ \\ z_{n+j}^{(i+1)} \end{bmatrix} = \begin{bmatrix} y_{n+j}^{(i)} \\ \\ z_{n+j}^{(i)} \end{bmatrix} - \begin{bmatrix} \frac{\partial F_j^{(i)}}{\partial y_{n+j}^{(i)}} & \frac{\partial F_j^{(i)}}{\partial z_{n+j}^{(i)}} \\ \\ \frac{\partial g_j^{(i)}}{\partial y_{n+j}^{(i)}} & \frac{\partial g_j^{(i)}}{\partial z_{n+j}^{(i)}} \end{bmatrix}^{-1} \cdot \begin{bmatrix} F_j^{(i)} \\ \\ g_j^{(i)} \end{bmatrix}$$

where $j = 1, 2$. Hence $e_{n+j}^{(i+1)}$ and $\hat{e}_{n+j}^{(i+1)}$, $j = 1, 2$ can be approximated and then the solution values $y_{n+j}^{(i+1)}$ and $z_{n+j}^{(i+1)}$ are computed from (9).

Choosing the step size

Choosing the step size is an important factor in the reduction of the number of iterations. The step size selection falls into three strategies. Using a prescribed tolerance value (TOL) an initial step size is determined. A test is conducted to compare the local truncation error (LTE) with TOL where

$$\text{LTEY} = \left| y_{n+2}^{(k)} - y_{n+2}^{(k-1)} \right|, \quad k = 4,$$
$$\text{LTEZ} = \left| z_{n+2}^{(k)} - z_{n+2}^{(k-1)} \right|,$$
$$\text{LTE} = \max\,(\text{LTEY}, \text{LTEZ}).$$

If LTE $<$ TOL the step is considered as successful. At this step, the previous step size is maintained (corresponding to using $q = 1$) and the following test will be conducted:

$$h_{\text{new}} = c h_{\text{old}} \left(\frac{\text{TOL}}{\text{LTE}} \right)^{1/k}, \tag{10}$$

where c is the safety factor, and k is the order of the method and is equal to 4. The h_{new} and h_{old} in (10) are the step size for the current and previous blocks, respectively. Here $c = 0.5$.

If $h_{\text{new}} > 1.6 h_{\text{old}}$ then $h_{\text{new}} = 1.6 h_{\text{old}}$. This corresponds to using the formula $q = \frac{5}{8}$. On the other hand, if LTE $>$ TOL, the step size is halved and we regard this step as a failed step (corresponding to the formula when $q = 2$).

NUMERICAL RESULTS

The performance of the variable step block BDF method on index-1 DAEs is examined using the following examples representing models of various DAEs occurring in engineering. Different tolerance values (10^{-2}, 10^{-4} and 10^{-6}) are used. The examples are also solved using variable step BDF for comparison purposes. The maximum error, successful steps, failed steps, total number of steps, and the time of each example are given and compared.

Example 1 [Ref. 13] $y' = f(x, y, z) = z$, $y(0) = 1$, $0 = g(x, y, z) = z^3 - y^2$, $z(0) = 1$, $0 \leqslant x \leqslant 10$. Exact solution: $y = (1 + \frac{1}{3}x)^3$, $z = (1 + \frac{1}{3}x)^2$.

Example 2 [Ref. 1] $y' = f(x, y, z) = x\cos x - y + (1+x)z$, $y(0) = 1$, $0 = g(x, y, z) = \sin x - z$, $z(0) = 0$, $0 \leqslant x \leqslant 10$. Exact solution: $y = f(x, y, z) = \mathrm{e}^{-x} + x\sin x$, $z = \sin x$.

Example 3 $y_1' = -xy_2 - (1+x)z_1$, $y_1(0) = 5$, $y_2' = xy_1 - (1+x)z_2$, $y_2(0) = 1$, $0 = \frac{1}{5}(y_1 - z_2) - \cos(\frac{1}{2}x^2)$, $z_1(0) = -1$, $0 = \frac{1}{5}(y_2 + z_1) - \sin(\frac{1}{2}x^2)$, $z_2(0) = 0$, $0 \leqslant x \leqslant 10$. Exact solution: $y_1 = \sin x + 5\cos(\frac{1}{2}x^2)$, $y_2 = \cos x + 5\sin(\frac{1}{2}x^2)$, $z_1 = -\cos x$, $z_2 = \sin x$.

It can be seen from Tables 1–3 that the 2BBDF method is more accurate than BDF method with the same order for all the examples tested. It is also observed that the 2BBDF has smaller execution time and fewer steps.

Table 1 Performance for Example 1.

TOL	Method	IFST	IST	TNS	MAXE	TIME
10^{-2}	BDF	1	76	77	3.0×10^{-2}	2.5×10^{-3}
	2BBDF	0	18	18	4.0×10^{-4}	7.4×10^{-4}
10^{-4}	BDF	1	98	99	3.6×10^{-4}	7.2×10^{-3}
	2BBDF	0	23	23	6.5×10^{-5}	9.1×10^{-4}
10^{-6}	BDF	1	136	137	3.6×10^{-5}	6.0×10^{-2}
	2BBDF	0	31	31	4.2×10^{-6}	1.2×10^{-3}

BDF = variable step BDF method; 2BBDF = variable step 2-point block BDF method; IST = the total number of successful steps; IFST = the total number of failed steps; TIME = the execution time

Table 2 Performance for Example 2.

TOL	Method	IFST	IST	TNS	MAXE	TIME
10^{-2}	BDF	9	106	115	7.9×10^{-3}	1.5×10^{-3}
	2BBDF	0	26	26	6.6×10^{-5}	3.0×10^{-4}
10^{-4}	BDF	12	179	191	1.4×10^{-4}	3.9×10^{-3}
	2BBDF	1	55	56	3.1×10^{-6}	7.8×10^{-4}
10^{-6}	BDF	18	326	344	2.3×10^{-6}	3.7×10^{-2}
	2BBDF	1	110	111	5.3×10^{-8}	3.0×10^{-3}

Table 3 Performance for Example 3.

TOL	Method	IFST	IST	TNS	MAXE	TIME
10^{-2}	BDF	5	102	107	2.7×10^{-1}	1.0×10^{-2}
	2BBDF	2	64	66	1.0×10^{-3}	2.6×10^{-3}
10^{-4}	BDF	4	231	235	2.5×10^{-3}	9.7×10^{-2}
	2BBDF	2	191	193	3.0×10^{-6}	7.5×10^{-3}
10^{-6}	BDF	7	665	672	2.8×10^{-5}	1.2×10^{-1}
	2BBDF	2	554	556	9.5×10^{-9}	2.1×10^{-2}

CONCLUSIONS

In this paper, a variable step 2-point block BDF method is applied to solve semi-explicit index-1 DAEs. The numerical results obtained for the test problems indicate that the maximum errors of the 2-point are lower than the existing variable step BDF method.

Acknowledgements: The authors would like to thank Institute for Mathematical Research, Universiti Putra Malaysia for supporting this study.

REFERENCES

1. Brennan KE, Campbell SL, Petzold L (1989) *Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, Elsevier, New York.
2. Kee RJ, Petzold LR, Smooke MD, Grcar JF (1985) Implicit methods in combustion and chemical kinetics modeling. In: Brackill W (ed) *Multiple Time Scales*, Academic, New York, pp 113-44.
3. McClamroch NH (1986) Singular system of differential equations as dynamic models for constrained robot systems. Technical report RSD TR-2-86, Robot Systems Division, Univ of Michigan.
4. Brita LG, Rabia OA (1987) Parallel block predictor-corrector methods for ODEs. *IEEE Trans Comput* **36**, 299–311.
5. Chu MT, Hamiton H (1987) Parallel solution of ODE's by multi-block methods. *SIAM J Sci Stat Comput* **8**, 342–53.
6. Fatunla SO (1991) Block methods for second order ODEs. *Int J Comput Math* **41**, 55–63.
7. Shampine LF, Watts HA (1969) Block implicit one-step methods. *Math Comput* **23**, 731–40.
8. Ascher UM, Petzold LR (1998) *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA.
9. Gear CW (1971) Simultaneous numerical solution of differential-algebraic equations. *IEEE Trans Circ Theor* **18**, 89–95.
10. Ascher UM, Petzold LR (1991) Projected implicit Runge-Kutta methods for differential algebraic equations. *SIAM J Numer Anal* **28**, 1097–120.
11. Abasi N, Suleiman M, Ibrahim Z, Ismail F (2012) 2-point and 3-point BBDF methods for solving semi-explicit index-1 DAEs. *Appl Math Sci* **6**, 6679–89.
12. Ibrahim Z, Othman KI, Suleiman M (2008) Fixed coefficients block backward differentiation formulas for the numerical solution of stiff ordinary differential equations. *Eur J Sci Res* **21**, 508–20.
13. Vijalapura PK, Strain J, Govinindjee S (2005) Fractional step methods for index-1 differential-algebraic equations. *J Comput Phys* **203**, 305–20.