Computers and Mathematics with Applications 65 (2013) 996–1005

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Continuous block backward differentiation formula for solving stiff ordinary differential equations☆

O.A. Akinfenwa [a,*], S.N. Jator [b], N.M. Yao [a]

[a] College of Computer Science and Technology, Harbin Engineering University, Harbin 15001, PR China

[b] Department of Mathematics, Austin Peay State University, Clarksville, TN 37044, USA

ARTICLE INFO

Article history:
Received 30 August 2011
Received in revised form 13 March 2012
Accepted 14 March 2012

Keywords:
Stiff problems
Continuous block BDF
Ordinary differential equations
Collocation and interpolation
Stability

ABSTRACT

In this paper, we consider an implicit Continuous Block Backward Differentiation formula (CBBDF) for solving Ordinary Differential Equations (ODEs). A block of p new values at each step which simultaneously provide the approximate solutions for the ODEs is derived, where p is the number of points. A performance comparison of the continuous block methods is made with existing methods. Numerical results indicate that the CBBDF is more efficient in improving the number of integration steps with better accuracy.

1. Introduction

The most popular class of implicit multistep methods for solving stiff ODEs is the Backward Differentiation Formula (BDF). These methods were first used for the solution of stiff problem by Curtis and Hirschfelder [1]. Over the years several implicit methods have been developed and discussed extensively in literature, see [2–8].

In this paper, the concern has to do with implicit BDFs for the numerical solution of Initial Value Problems (IVPs) for first-order ODEs of the form

$$y' = f(t, y), \quad t \in [t_0, T_n], \qquad y(t_0) = y_0, \tag{1}$$

which is generally written as

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\beta_k f_{n+\mathbf{k}}, \tag{2}$$

where h is the step size, $\alpha_k = 1$, α_j, $j = 1, \ldots, k$, β_k are unknown constants which are uniquely determined such that the formula is of order k. The objective of this paper is to develop a set of self-starting implicit block BDFs that will circumvent the conventional search for single step or lower ordered methods required to obtain starting values. It would be observed that block methods were first introduced by Milne [9] and since then several block methods have been developed by researchers such as [5,10–14] and the references therein. Most of the block methods in the literature use a predictor–corrector approach and some require a starting value through the use of Runge–Kutta or Taylor series for their

☆ This paper has been evaluated according to old Aims and Scope of the journal.

* Corresponding author.
E-mail addresses: akinolu35@yahoo.com, akin.sheye35@gmail.com (O.A. Akinfenwa), yaonianmin@hrbeu.edu.cn (S.N. Jator), Jators@apsu.edu (N.M. Yao).

0898-1221/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.03.111

implementation. However, we note that some of those methods have performed excellently well [15–17]. Our method as in [18,19] preserves the Runge–Kutta traditional advantage of being self-starting. The block algorithm proposed in this paper is based on interpolation and collocation, see Atkinson [20], Onumanyi et al. [21], and Gladwell and Sayers [22]. The continuous representation of the algorithm generates a main discrete collocation method to render the approximate solution y_{n+j} to the solution of (1) at points $t_{n+j},\ j = 1, \dots, k$.

Definition 1.1. A block-by-block method is a method for computing vectors $Y_0, Y_1, \dots$ in sequence (see Baker and Keech [23]). Let the ν-vector (ν is the number of points within the block) Y_μ, F_μ, and G_μ, for $n = m\nu, m = 0, 1, \dots$ be given as $Y_\omega = (y_{n+1}, \dots, y_{n+\nu})^T, F_\omega = (f_{n+1}, \dots, f_{n+\nu})^T$, then the l-block ν-point methods for (1) are given by

$$Y_\omega = \sum_{i=1}^{\ell} A^{(i)} Y_{\omega-i} + h \sum_{i=0}^{\ell} B^{(i)} F_{\omega-i}, \tag{3}$$

where $A^{(i)}, B^{(i)},\ i = 0, \dots, \ell$ are ν by ν matrices (see Fatunla [24]).

From the above definition a block method has the advantage that in each application, the solution is approximated at more than one point. The number of points depends on the structure of the block method. Therefore, applying these methods can give faster solutions to the problem which can be managed to produce a desired accuracy.

The rest of this paper is presented as follows: in Section 2 the basic idea behind the algorithm is discussed and a continuous representation $Y(t)$ for the exact solution $y(t)$ which is used to generate a main discrete block method for solving (1) is derived. In Section 3 the order of accuracy of the methods is presented. In Section 4 the stability regions of the implicit block BDFs are presented. In Section 5 we show the accuracy of the methods with some numerical *experiments*. Finally, in Section 6 some concluding remarks are presented.

2. Derivation of the method

In this section, the aim is to derive the main block method of the form (2). We proceed by seeking an approximation of the exact solution $y(t)$ by assuming a continuous solution $Y(t)$ of the form

$$Y(t) = \sum_{j=0}^{q+r-1} m_j \varphi_j(t), \tag{4}$$

such that $t \in [t_0, T_n]$, m_j are unknown coefficients and $\varphi_j(t)$ are polynomial basis functions of degree $q + r - 1$, where the number of interpolation points q and the collocation point r are respectively chosen to satisfy $q = k$ and $r = 1$. The integer $k \geq 1$ denotes the step number of the method. We thus construct a k-step block method with $\varphi_j(t) = t^j_{n+i}$ by imposing the following conditions

$$\sum_{j=0}^{q} m_j t^j_{n+i} = y_{n+i}, \quad i = 0, \dots, q-1, \tag{5}$$

$$\sum_{j=0}^{q} m_j j t^j_{n+i} - 1 = f_{n+i}, \quad i = k, \tag{6}$$

where y_{n+j} is the approximation for the exact solution $y(t_{n+j}), f_{n+j} = f(t_{n+j}, y_{n+j})$, n is the grid index and $t_{n+j} = t_n + jh$. It should be noted that Eqs. (5) and (6) leads to a system of $q + 1$ equations of the form $AM = C$ where

$$A = \begin{pmatrix} t_n^0 & t_n & t_n^2 & \cdots & t_n^q \\ t_{n+1}^0 & t_{n+1} & t_{n+1}^2 & \cdots & t_{n+1}^q \\ t_{n+2}^0 & t_{n+2} & t_{n+2}^2 & \cdots & t_{n+2}^q \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n+q-1}^0 & t_{n+q-1} & t_{n+q-1}^2 & \cdots & t_{q-1}^q \\ 0 & 1 & 2t_{n+k} & \cdots & qt_{n+k}^{q-1} \end{pmatrix}$$

$$M = (m_0, m_1, m_2, \dots, m_k)^T,$$

$$C = (y_n, y_{n+1}, y_{n+2}, \dots, y_{n+k-1}, f_n + k)^T,$$

which must be solved by using matrix inversion to obtain the coefficients m_j. The k-step continuous block BDF is then obtained by substituting these values of m_j into Eq. (4). After some algebraic computation, our method yields the expression in the form

$$Y(t) = -\sum_{j=0}^{q-1} \alpha_j(t) y_{n+j} + h\beta_k(t) f_{n+k}, \tag{7}$$

where $\alpha_j(t)$ and $\beta_k(t)$ are continuous coefficients. The method (6) is then used to generate the standard BDF (2) of order k at the desired point $t = t_{n+i},\ i = 1, 2, \ldots, q$. The additional methods are then obtained by evaluating the first derivative of (7) given by (8) at the points $t = (t_{n+i}),\ i = 1, 2, \ldots, q-1$

$$Y'(t) = \frac{1}{h}\left(\sum_{j=0}^{q-1} \alpha_j'(t) y_{n+j} + h\beta_k'(t) f_{n+k}\right). \tag{8}$$

These additional integrators are combined with the standard BDF (7) and implemented as a block method for any desired step of (7). For $k = 4$ taking $q = k$, $\varphi_j(t) = t_{n+i}^j$, $i = 0, 1, \ldots, 4$ and thus evaluating (7) at $t = t_{n+4}$, and combined with (8) at $t = [t_{n+1}, t_{n+2}, t_{n+3}]$ we generate the block method (9):

$$\left.\begin{aligned}
f_{n+1} &= \frac{1}{50h}[2hf_{n+4} - 13y_n - 39y_{n+1} + 69y_{n+2} - 17y_{n+3}]\\
f_{n+2} &= \frac{1}{75h}[-3hf_{n+4} + 7y_n - 54y_{n+1} + 9y_{n+2} + 38y_{n+3}]\\
f_{n+3} &= \frac{1}{150h}[18hf_{n+4} - 17y_n + 99y_{n+1} - 279y_{n+2} + 197y_{n+3}]\\
y_{n+4} &= \frac{1}{25}[12hf_{n+4} - 3y_n + 16y_{n+1} - 36y_{n+2} + 48y_{n+3}]
\end{aligned}\right\}. \tag{9}$$

Similarly, specifying $k = 6$, $q = k$, $\varphi_j(t) = t_{n+i}^j$, $i = 0, 1, \ldots, 6$ and thus evaluating (6) at $t = t_{n+6}$, together with (8) at $t = [t_{n+1}, t_{n+2}, t_{n+3}, t_{n+4}, t_{n+5}]$ yield the block method (10)

$$\left.\begin{aligned}
f_{n+1} &= \frac{1}{1764h}[24hf_{n+6} - 298y_n - 2235y_{n+1} + 4320y_{n+2} - 2780y_{n+3} + 1290y_{n+4} - 297y_{n+5}]\\
f_{n+2} &= \frac{1}{2205h}[-15hf_{n+6} + 76y_n - 900y_{n+1} - 1230y_{n+2} + 2840y_{n+3} - 990y_{n+4} + 204y_{n+5}]\\
f_{n+3} &= \frac{1}{8820h}[60hf_{n+6} - 157y_n + 1395y_{n+1} - 6840y_{n+2} + 400y_{n+3} + 6165y_{n+4} - 963y_{n+5}]\\
f_{n+4} &= \frac{1}{8820h}[-120hf_{n+6} + 167y_n - 1320y_{n+1} + 4860y_{n+2} - 12\,560y_{n+3} + 6045y_{n+4} + 2808y_{n+5}]\\
f_{n+5} &= \frac{1}{8820h}[600hf_{n+6} - 394y_n + 2925y_{n+1} - 9600y_{n+2} + 18\,700y_{n+3} - 26\,550y_{n+4} + 14\,919y_{n+5}]\\
y_{n+6} &= \frac{1}{147}[60hf_{n+6} - 10y_n + 72y_{n+1} - 225y_{n+2} + 400y_{n+3} - 450y_{n+4} + 360y_{n+5}]
\end{aligned}\right\}. \tag{10}$$

3. Order of accuracy

Following Fatunla [24] and Lambert [5] we define the local truncation error associated with (9) and (10) to be the linear difference operator

$$L[y(t); h] = \sum_{j=0}^{k} \alpha_j y_{n+j} - h\beta_k f_{n+k}. \tag{11}$$

Assuming that $y(t)$ is sufficiently differentiable, we can write the terms in (11) as a Taylor series expression of $y(t_{n+j})$ and $f(t_{n+j}) = y'(t_{n+j})$ as

$$y(t_{n+j}) = \sum_{p=0}^{\infty} \frac{(jh)}{p!} y^{(p)}(t_n) \quad \text{and} \quad y'(t_{n+j}) = \sum_{p=0}^{\infty} \frac{(jh)}{p!} y^{(p+1)}(t_n). \tag{12}$$

Substituting these into Eqs. (9) and (10) we obtain the expression

$$L[y(t); h] = C_0 y(t) + C_1 h y'(t) + C_2 h^2 y''(t) + \cdots, + C_p h^p y^p(t) + \cdots, \tag{13}$$

Table 1
Orders and error constants for block methods (9) and (10).

Method	Order p	Error constant C_{p+1}
(9)	4	$-\frac{29}{500}$
	4	$\frac{31}{750}$
	4	$-\frac{37}{500}$
	4	$-\frac{12}{125}$
(10)	6	$-\frac{53}{2085}$
	6	$\frac{18}{1715}$
	6	$-\frac{167}{20\,580}$
	6	$\frac{59}{5145}$
	6	$-\frac{23}{686}$
	6	$-\frac{20}{343}$

where the constant coefficients $C_p,\ p = 0, 1, 2, \ldots, l = 1, 2, \ldots, k$ are given as follows:

$$C_0 = \sum_{j=0}^{k} \alpha_j$$

$$C_1 = \sum_{j=1}^{k} j\alpha_j - \beta_k + \eta_l$$

$$C_2 = \frac{1}{2!}\left(\sum_{j=1}^{k} j^2\alpha_j - 2k\beta_k + 2l\eta_l\right)$$

$$\vdots$$

$$C_p = \frac{1}{p!}\left(\sum_{j=1}^{k} j^p\alpha_j - pk^{p-1}\beta_k + pl^{p-1}\eta_l\right)$$

where $\eta_k = 0$ and $\eta_l = 1, l = 1, \ldots, k-1$.

According to Henrici [25], we say that the method (2) has order p if

$$L[y(t); h] = O(h^{p+1}), \qquad C_0 = C_1 = \cdots = C_p = 0, \qquad C_{p+1} \neq 0. \tag{14}$$

Therefore, C_{p+1} is the error constant and $C_{p+1}h^{p+1}y^{(p+1)}(t_n)$ is the principal local truncation error at the point t_n. It was established from our calculations that the block methods (9) and (10) have order and error constants as displayed in Table 1.

4. Stability analysis

In what follows, the new k-step CBBDF can be generally rearranged and rewritten as a matrix finite difference equation of the form

$$A^{(1)}Y_{\omega+1} = A^{(0)}Y_{\omega} + hB^{(1)}F_{\omega}, \tag{15}$$

where

$$Y_{\omega+1} = (y_{n+1}, y_{n+2}, y_{n+3}, \ldots, y_{n+k-1}, y_{n+})^T,$$
$$Y_{\omega} = (y_{n-k+1}, y_{n-k+2}, y_{n-k+3}, \ldots, y_{n-1}, y_n)^T$$
$$F_{\omega} = (f_{n+1}, f_{n+2}, f_{n+3}, \ldots, f_{n+k})^T,$$

for $\omega = 0, \ldots$ and $n = 0, k, \ldots, N-k$.
The matrices $A^{(1)}, A^{(0)}, B^{(1)}$ are K by K matrices whose entries are given by the combined coefficients of (7) and (8) evaluated at the $t = t_{n+k}$ point and $t = (t_{n+1}, t_{n+2}, \ldots, t_{n+k-1})$ respectively. We thus define the matrices as follows:

$$A^{(1)} = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \cdots & \mu_{1k-1} & \mu_{1k} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} & \cdots & \mu_{2k-1} & \mu_{2k} \\ \mu_{31} & \mu_{32} & \mu_{13} & \mu_{14} & \cdots & \mu_{3k-1} & \mu_{3k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{k-11} & \mu_{k-12} & \mu_{k-13} & \mu_{k-14} & \cdots & \mu_{k-1k-1} & \mu_{k-1k} \\ \mu_{k1} & \mu_{k2} & \mu_{k3} & \mu_{k4} & \cdots & \mu_{kk-1} & \mu_{kk} \end{pmatrix}$$

where $\mu_{1k} = \mu_{2k} = \mu_{3k} = \cdots = \mu_{k-1k} = 0$ and $\mu_{kk} = 1$.

$$A^{(0)} = \begin{pmatrix} 0 & 0 & 0 & \cdots & \mu 0_{1k} \\ 0 & 0 & 0 & \cdots & \mu 0_{2k} \\ 0 & 0 & 0 & \cdots & \mu 0_{3k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mu 0_{k-1k} \\ 0 & 0 & 0 & \cdots & \mu 0_{kk} \end{pmatrix}$$

$$B^{(1)} = \begin{pmatrix} -1 & 0 & 0 & 0 & \cdots & b_{1k} \\ 0 & -1 & 0 & 0 & \cdots & b_{2k} \\ 0 & 0 & -1 & 0 & \cdots & b_{3k} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & b_{k-1k} \\ 0 & 0 & 0 & \cdots & 0 & b_{kk} \end{pmatrix}.$$

4.1. Linear stability

The linear stability properties of the continuous block BDFs are determined by expressing them in the form (15) and applying the test problem

$$y' = \lambda y, \quad \lambda < 0$$

to yield

$$Y_{\omega+1} = D(z)Y_{\omega}, \quad z = \lambda h, \tag{16}$$

where the matrix $D(z)$ is given by

$$D(z) = (A^{(1)} - zB^{(1)})^{-1}A^{(0)}. \tag{17}$$

The matrix $D(z)$ has eigenvalues $\{\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_k\} = \{0, 0, 0, \ldots, \gamma_k\}$, where the dominant eigenvalue γ_k is the stability function $R(z) : \mathbb{C} \to \mathbb{C}$ which is a rational function with real coefficients.

In particular taking $k = 4$ we have that

$$A^{(1)} = \begin{pmatrix} \frac{39}{50} & -\frac{69}{50} & \frac{17}{50} & 0 \\ \frac{18}{25} & -\frac{3}{25} & -\frac{38}{75} & 0 \\ -\frac{33}{50} & \frac{93}{50} & -\frac{197}{150} & 0 \\ -\frac{16}{25} & \frac{36}{25} & -\frac{48}{25} & 1 \end{pmatrix}$$

$$A^{(0)} = \begin{pmatrix} 0 & 0 & 0 & -\frac{13}{50} \\ 0 & 0 & 0 & \frac{7}{75} \\ 0 & 0 & 0 & -\frac{17}{150} \\ 0 & 0 & 0 & -\frac{3}{25} \end{pmatrix}$$

$$B^{(1)} = \begin{pmatrix} -1 & 0 & 0 & \frac{2}{50} \\ 0 & -1 & 0 & -\frac{3}{75} \\ 0 & 0 & -1 & \frac{3}{25} \\ 0 & 0 & 0 & \frac{12}{25} \end{pmatrix}.$$

Applying the test equation with $z = \lambda h$, from (17) the stability function $R(z)$ is given by

$$R(z) = \frac{12 + 18z + 11z^2 + 3z^3}{12 - 30z + 35z^2 - 25z^3 + 12z^4}. \tag{18}$$

For $k = 6$ the matrices $A^{(1)}, A^{(0)}, B^{(1)}$ are given as

$$A^{(1)} = \begin{pmatrix} \frac{745}{588} & -\frac{120}{49} & \frac{695}{441} & -\frac{215}{294} & \frac{33}{196} & 0 \\ \frac{20}{49} & \frac{246}{441} & -\frac{568}{441} & \frac{22}{49} & -\frac{68}{735} & 0 \\ -\frac{93}{588} & \frac{342}{441} & -\frac{20}{441} & -\frac{1233}{1764} & \frac{107}{980} & 0 \\ \frac{22}{147} & -\frac{27}{49} & \frac{1256}{882} & -\frac{403}{588} & -\frac{702}{2205} & 0 \\ -\frac{585}{1764} & \frac{160}{147} & -\frac{1870}{882} & \frac{2655}{882} & -\frac{4973}{2940} & 0 \\ -\frac{24}{49} & \frac{75}{49} & -\frac{400}{147} & \frac{150}{49} & -\frac{120}{49} & 1 \end{pmatrix}$$

$$A^{(0)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\frac{149}{882} \\ 0 & 0 & 0 & 0 & 0 & \frac{76}{2205} \\ 0 & 0 & 0 & 0 & 0 & -\frac{157}{8820} \\ 0 & 0 & 0 & 0 & 0 & \frac{167}{8820} \\ 0 & 0 & 0 & 0 & 0 & -\frac{197}{4410} \\ 0 & 0 & 0 & 0 & 0 & -\frac{10}{147} \end{pmatrix}$$

$$B^{(1)} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & \frac{6}{441} \\ 0 & -1 & 0 & 0 & 0 & -\frac{3}{441} \\ 0 & 0 & -1 & 0 & 0 & \frac{1}{147} \\ 0 & 0 & 0 & -1 & 0 & -\frac{12}{882} \\ 0 & 0 & 0 & 0 & -1 & \frac{10}{147} \\ 0 & 0 & 0 & 0 & 0 & \frac{280}{147} \end{pmatrix}.$$

Also by applying the test equation with $z = \lambda h$, from (17) we obtained the stability function $R(z)$ for $k = 6$ as

$$R(z) = \frac{360 + 900z + 1020z^2 + 675z^3 + 274z^4 + 60z^5}{360 - 1260z + 2100z^2 - 2205z^3 + 1624z^4 - 882z^5 + 360z^6}. \tag{19}$$

In the spirit of Hairer and Wanner [26], the stability regions for both $k = 4$ and $k = 6$ are drawn using Eqs. (18) and (19) as shown in Figs. 1 and 2 respectively.

The stability region for the method $k = 4$ lies outside the bounded region. Thus for $k = 4$ the method is A-stable, although for $k = 6$, it is obvious from Fig. 2 that it is not A-stable since part of the bounded region lies inside the left half complex plane. However, the method is L_0-stable, see Cash [3] since (19) satisfies the requirement that:

$$\operatorname*{Max}_{z \le 0} |R(z)| \le 1, \quad z \in R \quad \text{and} \quad \lim_{z \to -\infty} R(z) = 0.$$

Implementation. The two newly derived methods are implemented more efficiently as a k-step block numerical integrator for (1) without requiring starting values and predictors by explicitly obtaining initial conditions at t_{n+k}, $n = 0, k, \ldots, N-k$, using the computed values $H(t_{n+k}) = y_{n+k}$ over sub-intervals $[t_0, t_k], \ldots, [t_{N-k}, t_N]$. In particular, for $k = 4$, the computation using the CBBDF (15) is as follows.

For $n = 0, \omega = 0$, $(y_{n+1}, y_{n+2}, y_{n+3}, y_{n+4})^T$, are simultaneously obtained over the sub-interval $[t_0, t_4]$, as y_0 is known from (1).

For $n = 1, \omega = 1$, $(y_{n+5}, y_{n+6}, y_{n+7}, y_{n+8})^T$ are simultaneously obtained over the sub-interval $[t_4, t_8]$, as y_4 is known from the previous block, and so on. Similarly, we apply the above for $k = 6$.

Hence, the sub-intervals do not overlap and the solutions obtained in this manner are more accurate than those obtained in the conventional way. It should be noted that for a linear problem, the resulting $k \times k$ matrix in each block is solved with

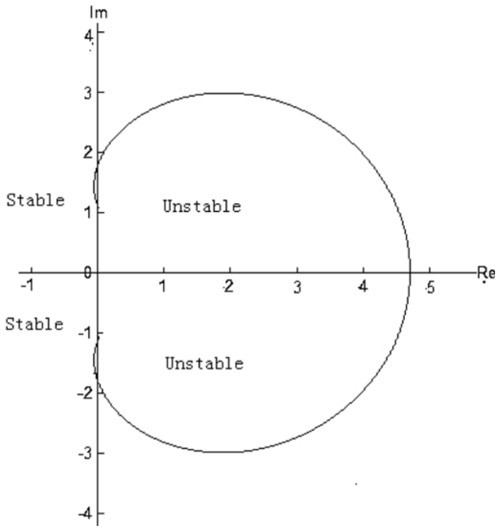

Fig. 1. Stability region for $k = 4$.

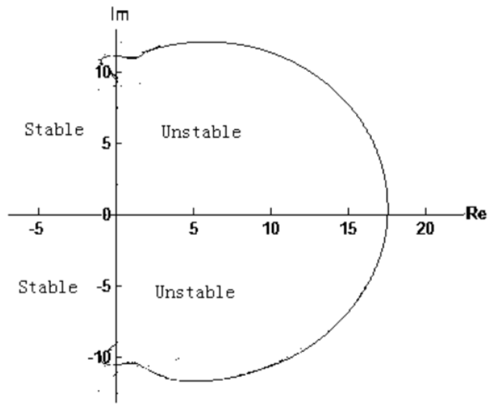

Fig. 2. Stability region for $k = 6$.

our written Matlab code while for a nonlinear problem the code uses the Newton iteration. The following notation is used to specify the iteration: y_{n+i}^{j+1} denotes the $(j+1)th$ iterative value of y_{n+i} and $\delta_{n+i}^{j+1} = y_{n+i}^{j+1} - y_{n+i}^{j}$ for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots$ Thus the Newton iteration of the k-point continuous block BDF for (16) takes the form

$$y_{n+i}^{(j+1)} = y_{n+i}^{(j)} - \frac{f_{n+i}^{(j)}}{f_{n+i}^{\prime(j)}}, \tag{20}$$

$$y_{n+1}^{(j+1)} - y_{n+1}^{(j)} = \frac{a_1 y_{n+1}^{(j)} + a_2 y_{n+2}^{(j)} + \cdots + a_{k-1} y_{n+k-1}^{(j)} + hf_{n+1}^{(j)} + h\beta_k f_{n+k}^{(j)}}{1 + h\frac{\delta f_{n+1}}{\delta y_{n+1}} + h\beta_k \frac{\delta f_{n+k}}{\delta y_{n+k}}} + D_1$$

$$y_{n+2}^{(j+1)} - y_{n+2}^{(j)} = \frac{c_1 y_{n+1}^{(j)} + c_2 y_{n+2}^{(j)} + \cdots + c_{k-1} y_{n+k-1}^{(j)} + hf_{n+2}^{(j)} + hV_k f_{n+k}^{(j)}}{1 + h\frac{\delta f_{n+2}}{\delta y_{n+2}} + hV_k \frac{\delta f_{n+k}}{\delta y_{n+k}}} + D_2$$

$$y_{n+3}^{(j+1)} - y_{n+3}^{(j)} = \frac{d_1 y_{n+1}^{(j)} + d_2 y_{n+2}^{(j)} + \cdots + c_{k-1} y_{n+k-1}^{(j)} + hf_{n+3}^{(j)} + h\nu_k f_{n+k}^{(j)}}{1 + h\frac{\delta f_{n+3}}{\delta y_{n+2}} + h\nu_k \frac{\delta f_{n+k}}{\delta y_{n+k}}} + D_3$$

$$\ldots$$

$$y_{n+k}^{(j+1)} - y_{n+k}^{(j)} = \frac{g_1 y_{n+1}^{(j)} + g_2 y_{n+2}^{(j)} + \cdots + g_{k-1} y_{n+k}^{(j)} + y_{n+k}^{(j)} + h\psi_k f_{n+8}^{(j)}}{1 + h\psi_k \frac{\delta f_{n+k}}{\delta y_{n+k}}} + D_k.$$

Table 2
Relative error for continuous block BDF of order 4 and 6 for Experiment 5.1.

Step	$k = 4$		$k = 6$	
	Relative error	Rate	Relative error	Rate
20	3.8×10^{-2}	–	4.7×10^{-2}	–
40	2.1×10^{-2}	1.2	2.1×10^{-3}	3.6
80	3.3×10^{-3}	2.7	1.4×10^{-4}	3.9
160	4.2×10^{-4}	3.2	7.5×10^{-6}	4.1
320	2.2×10^{-5}	3.9	1.7×10^{-7}	5.7
640	1.4×10^{-6}	4.0	3.0×10^{-9}	5.9

Table 3
Relative error for BDF of order 4 and 6 for Experiment 5.1.

Step	$k = 4$		$k = 6$	
	Relative error	Rate	Relative error	Rate
20	8.7×10^{-2}	–	2.0×10^{-1}	–
40	6.8×10^{-2}	0.4	2.6×10^{-1}	$*$
80	1.1×10^{-2}	2.6	2.6×10^{-3}	$*$
160	8.0×10^{-4}	3.8	9.1×10^{-5}	4.8
320	6.8×10^{-5}	3.6	1.8×10^{-6}	5.6
640	5.3×10^{-6}	3.7	3.3×10^{-8}	5.8

Put in matrix form it then becomes:

$$J^{(1)}\delta^{(1)} = \alpha^{(0)}Y^{(1)} + h\beta^{(0)}F^{(1)} + D, \tag{21}$$

where $J^{(1)}$ is the Jacobian matrix, and $D = D_1, D_2, \dots, D_k$ are known from the initial value of the problem. Thus we obtain the approximated values of $y_{n+1}, y_{n+2}, \dots, y_{n+k}$ as

$$\begin{aligned} y_{n+1}^{(j+1)} &= y_{n+1}^{(j)} + \delta_{n+1}^{(j+1)} \\ y_{n+2}^{(j+1)} &= y_{n+2}^{(j)} + \delta_{n+2}^{(j+1)} \\ &\vdots \\ y_{n+k}^{(j+1)} &= y_{n+k}^{(j)} + \delta_{n+k}^{(j+1)}. \end{aligned}$$

5. Numerical experiments

This section deals with some numerical experiments, executed in Matlab language with double precision arithmetic, which illustrate the result derived in the previous sections.

Experiment 5.1. Consider the linear problem

$$y'(t) = \begin{bmatrix} -21 & 19 & -20 \\ 19 & -21 & 20 \\ 40 & -40 & -40 \end{bmatrix} \mathbf{y(t)}, \qquad y(0) = (1, 0, -1)^T,$$

with theoretical solution

$$\begin{aligned} y_1(t) &= \frac{1}{2}(e^{-2t} + e^{-40t}(\cos(40t) + \sin(40t))) \\ y_2(t) &= \frac{1}{2}(e^{-2t} - e^{-40t}(\cos(40t) + \sin(40t))) \\ y_3(t) &= -\frac{1}{2}(2e^{-40t}(\cos(40t) - \sin(40t))). \end{aligned}$$

The main aim is to show the good stability properties and accuracy of the Continuous Block BDF (CBBDF) in comparison with the BDF used with the same sequence of step sizes. For different choices of the constant step size h the relative error $\max_i \frac{|y_i - y(t_i)|}{|(1+y(t_i))|}$ and the Rate Of Convergence (ROC) which is calculated using the formula $\mathrm{ROC} = \log_2\left(\frac{e^{2h}}{eh}\right)$, e^h is the maximum absolute error for h. In all cases the rate of convergence is consistent with the methods. Thus, for this example, our method is superior in terms of accuracy (smaller errors) and efficiency (smaller number of function evaluations, NFEs) (see Tables 2 and 3).

Table 4
A comparison of NFEs for Experiment 5.1.

Steps	BDF(NFEs)		CBBDF(NFEs)	
	$k = 4$	$k = 6$	$k = 4$	$k = 6$
20	29	35	20	24
40	49	55	40	42
80	89	95	80	82
160	169	175	160	162
320	329	335	320	322
640	649	655	640	642

Table 5
Absolute error for our method $k = 6$ for Experiment 5.2.

t	h	N	Y	Theoretical	Akinfenwa, Jator, and Yao numerical $k = 6$	Absolute error
1	0.02	50	y_1	1.353352832366127e−1	1.353352832375237e−1	9.1102e−13
			y_2	3.678794411714423e−1	3.678794411726950e−1	1.2527e−12
10	0.02	500	y_1	2.061153622416581e−9	2.061153622438558e−9	2.1977e−20
			y_2	4.539992976383902e−5	4.53999297624848e−5	1.3542e−15

Table 6
Absolute error for our method $k = 4$ for Experiment 5.2.

t	h	N	Y	Theoretical	Akinfenwa, Jator, and Yao numerical $k = 4$	Absolute error
1	0.02	50	y_1	1.353352832366127e−1	1.35335286619327e−1	3.3827e−9
			y_2	3.678794411714423e−1	3.678794457979147e−1	4.6265e−9
10	0.02	500	y_1	2.061153622416581e−9	2.061154110095654e−9	4.8766e−16
			y_2	4.539992976383902e−5	4.539993515208483e−5	5.38966e−12

Table 7
Absolute error for Wu and Xia [16] for Experiment 5.2.

t	h	N	Y	Theoretical	Wu and Xia [16] numerical [16]	Absolute error
1	0.002	500	y_1	1.353352832366127e−1	1.353350271728111e−1	2.5606e−7
			y_2	3.678794411714423e−1	3.678795213211519e−1	8.0150e−8
10	0.001	10 000	y_1	2.061153622416581e−9	2.061154177118385e−9	5.5468e−16
			y_2	4.539992976383902e−5	4.539993585613384e−5	6.0936e−12

Experiment 5.2. Consider the nonlinear problem;

$$y_1' = -1002y_1 + 1000y_2^2, \qquad y_1(0) = 1.$$
$$y_2' = y_1 - y_2(1 + y_2), \qquad y_2(0) = 1.$$

This problem has also been solved by Wu and Xia [16] using Two Low Accuracy Explicit Methods in vector form. Their result is here reproduced in Table 7 and compared with our methods in Tables 5 and 6 with values of t the independent variable, h the step size, N the number of computation steps, the theoretical solutions, our numerical solutions and the absolute error.

6. Conclusion

A continuous block BDF has been proposed and implemented as a self-starting method which requires only the initial value in (1) for the solution of ordinary differential equations. The proposed method is not only accurate but also has reduced computational cost as evident in Tables 4 and 5 respectively with fewer numbers of function evaluations and computational steps. The good *convergence* and stability properties of our method make it suitable for numerical solution of stiff problems. The results demonstrate its efficiency and good accuracy over non-block methods.

Acknowledgments

The authors would like to thank the State Key Laboratory of High-end Server and Storage Technology, No. 2009HSSA08 and the Fundamental Research Funds for the Central Universities China, Nos. HEUCFT1007, and HEUCF100607, for the their financial support in this research work. Also, we thank the referees whose useful suggestions greatly improved the quality of this paper.

References

[1] C.F. Curtis, J.D. Hirschfelder, Integration of stiff equations, Proc. Natl. Acad. Sci. 38 (1952) 235–243.
[2] J.R. Cash, On integration on stiff system of ordinary differential equations using extended BDF, Numer. Math. 34 (1980) 235–246.
[3] J.R. Cash, Review paper. Efficient numerical methods for the solution of stiff initial-value problems and differential algebraic equations, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 459 (2032) (2003) 797–815.
[4] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, NJ, 1971.
[5] J.D. Lambert, Numerical Methods For Ordinary Differential Systems, The Initial Value Problem, Wiley, Chichester, New York, 1991.
[6] L.F. Shampine, Numerical Solution of Ordinary Differential Equations, Chapman & Hall, New York, NY, 1994.
[7] S. Skelboe, B. Christiansen, Backward differentiation formulas with extended region of absolute stability, BIT 21 (1981) 221–231.
[8] R.D. Skeel, A.K. Kong, Blended linear multistep methods, UIUCDCS-R-76-800, Department of Computer Science, University of Illinois, Urbana, IL, 1976.
[9] W.E. Milne, Numerical Solution of Differential Equations, John Wiley and Sons, 1953.
[10] M.G. Lee, R.W. Song, Computation of stability region of some block methods and their application to numerical solutions of ODEs, in: Proceedings of XIV Baikal International Conference, Baikal, Siberia, Russia, June 2008.
[11] M.G. Lee, R.W. Song, A new block method for stiff differential equations, in: International Conference on Scientific Computation and Differential Equations, SciCADE2009, Beijing, China.
[12] G. Psihoyios, A Block implicit advanced step-point (BIAS) algorithm for stiff differential systems, Comput. Lett. 2 (1–2) (2006) 51–58.
[13] J.E. Bond, J.R. Cash, A Block method for the numerical integration of stiff systems of ODEs, BIT 19 (1979) 429–447.
[14] D. Voss, S. Abbas, Block predictor–corrector schemes for the parallel solution of ODEs, Comput. Math. Appl. 33 (1997) 65–72.
[15] Z. Majid, Parallel block methods for solving ordinary differential equations, Ph.D. Thesis, Universiti Putra Malaysia, 2004.
[16] X.Y. Wu, J.L. Xia, Two low accuracy methods for stiff system, Appl. Math. Comput. 123 (2001) 141–153.
[17] L. Brugnano, D. Trigiante, Block implicit methods for ODEs, in: Recent Trends in Numerical Analysis, Nova Science Publishers Inc., Commack, NY, 2000.
[18] S.N. Jator, On the hybrid method with three-off step points for initial value problems, Int. J. Math. Educ. Sci. Technol. 41 (1) (2010) 110–118. 1464–5211.
[19] D. Sarafyan, Multistep methods for the numerical solution of ordinary differential equations made self-starting, Tech. Report 495, Math. Res. Center, Madison, 1965.
[20] K.E. Atkinson, An Introduction to Numerical Analysis, second ed., John Wiley and Sons, New York, 1989.
[21] P. Onumanyi, D.O. Awoyemi, S.N. Jator, U.W. Sirisena, New linear mutlistep methods with continuous coefficients for first order initial value problems, J. Nigerian Math. Soc. 13 (1994) 37–51.
[22] I. Gladwell, D.K. Sayers (Eds.), Computational Techniques for Ordinary Differential Equations, Academic Press, New York, 1976.
[23] C.T.H. Baker, M.S. Keech, Stability regions in the numerical treatment of Volterra integral equations, SIAM J. Numer. Anal. 15 (1978) 394–417.
[24] S.O. Fatunla, Block methods for second order IVPs, Int. J. Comput. Math. 41 (1991) 55–63.
[25] P. Henrici, Discrete Variable Methods in ODEs, John Wiley, New York, 1962.
[26] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer, New York, 1996.