Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 3, No. 2, 2015, pp. 99-110

Application of the block backward differential formula for numerical solution of Volterra integro-differential equations

Somayyeh Fazeli
Marand Faculty of Engineering,
University of Tabriz, Tabriz-Iran.
E-mail: fazeli@tabrizu.ac.ir

Abstract In this paper, we consider an implicit block backward differentiation formula (BBDF) for solving Volterra Integro-Differential Equations (VIDEs). The approach given in this paper leads to numerical methods for solving VIDEs which avoid the need for special starting procedures. Convergence order and linear stability properties of the methods are analyzed. Also, methods with extensive stability region of orders 2, 3 and 4 are constructed which are suitable for solving stiff VIDEs.

Keywords. Volterra integro-differential equations, Block methods, Backward differential formula.

2010 Mathematics Subject Classification. 65R20.

1. Introduction

Consider Volterra Integro-Differential Equations (VIDEs) of the form

$$y'(t) = g(t, y(t)) + \int_0^t K(t, \tau, y(\tau))d\tau, \qquad t \in I := [0, T], \qquad y(0) = y_0, \qquad (1.1)$$

where $g \in C(S)$ and the kernel $K \in C(\Omega)$ with $S = \{(t, y) : \ t \in I, y \in \mathbb{R}\}$, $\Omega = \{(t, \tau, y) : \ 0 \le \tau \le t \le T, \ y \in \mathbb{R}\}$, denote given functions which are (at least) continuous on their respective domain and satisfy a uniform Lipschitz condition with respect to y. In these hypotheses there exists a unique solution $y \in C^1(I)$ (see [5]). It is convenient to rewrite this equation in the form

$$y'(t) = f(t, y(t)),$$

where

$$f(t, y(t)) = g(t, y(t)) + \int_0^t K(t, \tau, y(\tau))d\tau.$$

VIDEs arise as mathematical models of many physical and biological phenomena with memory, such as population dynamics, viscoelasticity in materials with memory, fluid dynamics (see [8, 15] and references therein contained).

Several numerical methods have been proposed in the literature for the solution of (1.1), such as linear multistep methods, Runge-Kutta methods, collocation methods [4, 5], and Galerkin type methods for linear VIDEs [10]. In [16] construction of the

Received: 27 October 2015 ; Accepted: 23 February 2016.

quadrature rules generated by the backward differentiation formulae is discussed in detail and their linear stability properties are analyzed. In some literature, hybrid methods are used for numerical solution of VIDEs [11,13].

In this paper we are concerned with the backward differentiation formula (BDF) for solving VIDEs which is generally written as

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\beta_k f_{n+k}, \tag{1.2}$$

where h is the step size, $\alpha_k = 1$ and α_j, $j = 1, 2, \cdots, k-1$, β_k are unknown constants which are uniquely determined such that the formula is of order k. The implementation of the BDF methods for solving stiff ODEs was discussed by Gear in [7]. The block methods were first introduced by Milne [14] and several block methods for numerical integration of ordinary differential equations have been introduced in [3]. Recently continuous block BDF have been used for solving siff ODEs [1]. The aim of these methods is to develop of self-starting implicit block BDFs where the starting values are not computed by other methods. Here, we use this technique for numerical treatment of VIDEs in order to construct high order methods with extensive stability regions. In many of numerical approaches, one or more starting values are required which must be found by other methods. The method which we now describe, gives starting values directly.

Next sections of this paper are organized as follows. In Section 2, we describe the construction of CBBDF for VIDEs. In Section 3, we determine convergence orders of the methods and in Section 4, we analyze the linear stability properties of the method. Some examples of methods are described in Section 5. In Section 6, efficiency of the methods are shown by some numerical experiments.

2. Construction of the method

In this section, we describe construction of the main block method of the form (1.2) where the solution of (1.1) is approximated by assuming a continuous solution of the form

$$Y(t) = \sum_{j=0}^{k} m_j \varphi_j(t), \tag{2.1}$$

where $t \in [0, T]$, the coefficients m_j are unknown, the functions $\varphi_j(t)$ are polynomial basic functions and the integer $k \geq 1$ denotes the step number of the method. Let us define a uniform partition of $[0, T]$ in the form $0 = t_0 < t_1 < \cdots < t_N = T$, such that $t_n = nh, \quad n = 0, \cdots, N$, contrained that $N = kr$ for some $r \in \mathbb{N}$. By setting $\bar{n} = nk$, we construct the k-step method with $\varphi_j(t) = t^{j-1}$ where imposing the interpolation condition for unknown function at the points $t_{\bar{n}+i}$, $i = 0, 1, \cdots, k-1$ and the interpolation condition for derivative of unknown function at the point $t_{\bar{n}+k}$ lead to $k+1$ equations for determination of m_j in the form

$$\sum_{j=0}^{k} m_j t_{\bar{n}+i}^{j} = y_{\bar{n}+i}, \qquad i = 0, \cdots, k-1, \tag{2.2}$$

$$\sum_{j=0}^{k} m_j j t_{\bar{n}+i} = f_{\bar{n}+i}, \qquad i = k. \tag{2.3}$$

Let us define

$$A = (t^{j-1}_{\bar{n}+i-1})_{i,j} \in R^{(k+1)\times(k+1)}, \quad M = [m_0, m_1, \cdots, m_k]^T,$$
$$C = [y_{\bar{n}}, y_{\bar{n}+1}, \cdots, y_{\bar{n}+k-1}, f_{\bar{n}+k}]^T.$$

The equations (2.2) and (2.3) lead to a system of $k+1$ equations of the form $AM = C$ to obtain the coefficients m_j in terms of $y_{\bar{n}}$, $y_{\bar{n}+1}$, $\cdots$, $y_{\bar{n}+k-1}$ and $f_{\bar{n}+k}$. Then the $k-$step block BDF method is obtained by substituting the values of m_j in (2.1) which yields the expression in the form

$$Y(t) = -\sum_{j=0}^{k-1} \alpha_j(t) y_{\bar{n}+j} + h\beta_k(t) f_{\bar{n}+k}, \tag{2.4}$$

where $\alpha_j(t)$ and $\beta_k(t)$ are continuous functions. Infact, the approximation given by (2.4) is the Hermite interpolation polynomial which is obtained by the values of $y(t)$ in the points $t_{\bar{n}+j}$, $j = 0, 1, \cdots, k-1$ and the value of $y'(t)$ in $t_{\overline{n+1}}$. By differentiating from (2.4) and evaluating it at the point $t_{\overline{n+1}}$, and also evaluating (2.4) at the points $t_{\overline{n}+1}, \cdots, t_{\bar{n}+k-1}$, the block method is obtained in the form

$$\begin{aligned}
&f_{\overline{n}+1} = \left(\beta_{1,k} h f_{\overline{n}+k} + \alpha_{1,0} y_{\overline{n}} - \alpha_{1,1} y_{\overline{n}+1} - \cdots - \alpha_{1,k-1} y_{\overline{n}+k-1}\right)/h,\\
&\vdots\\
&f_{\overline{n}+k-1} = \left(\beta_{k-1,k} h f_{\overline{n}+k} + \alpha_{k-1,0} y_{\overline{n}} - \alpha_{k-1,1} y_{\overline{n}+1} - \cdots - \alpha_{k-1,k-1} y_{\overline{n}+k-1}\right)/h,\\
&y_{\overline{n}+k} = \beta_{k,k} h f_{\overline{n}+k} + \alpha_{k,0} y_{\overline{n}} - \alpha_{k,1} y_{\overline{n}+1} - \cdots - \alpha_{k,k-1} y_{\overline{n}+k-1}.
\end{aligned} \tag{2.5}$$

These methods can be represented in the matrix form as

$$A^{(1)} Y_{\overline{n+1}} = A^{(0)} Y_{\overline{n}} + h B^{(1)} F_{\overline{n+1}}, \tag{2.6}$$

where

$$Y_{\overline{n+1}} = [y_{\overline{n}+1}, y_{\overline{n}+2}, \cdots, y_{\overline{n}+k-1}, y_{\overline{n}+k}]^T \in \mathbb{R}^k,$$
$$Y_{\overline{n}} = [y_{\overline{n}-k+1}, y_{\overline{n}-k+2}, \cdots, y_{\overline{n}-1}, y_{\overline{n}}]^T \in \mathbb{R}^k,$$
$$F_{\overline{n+1}} = [f_{\overline{n}+1}, f_{\overline{n}+2}, \cdots, f_{\overline{n}+k-1}, f_{\overline{n}+k}]^T \in \mathbb{R}^k,$$

$$A^{(0)} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \alpha_{1,0} \\ 0 & 0 & \cdots & 0 & \alpha_{2,0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \alpha_{k,0} \end{bmatrix}, \quad A^{(1)} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,k-1} & 0 \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,k-1} & 0 \\ \vdots & \vdots & \dots & \vdots & 0 \\ \alpha_{k,1} & \alpha_{k,2} & \cdots & \alpha_{k,k-1} & 1 \end{bmatrix},$$

$$B^{(1)} = \begin{bmatrix} -1 & 0 & \cdots & 0 & \beta_{1,k} \\ 0 & -1 & \cdots & 0 & \beta_{2,k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & \beta_{k-1,k} \\ 0 & 0 & \cdots & 0 & \beta_{k,k} \end{bmatrix},$$

By solving the nonlinear system (2.5) for unknowns $y_{\overline{n}+1}, \cdots, y_{\overline{n}+k}$, the method is obtained. In practice, we need to compute $f_{\overline{n}+i}$ which is in the form

$$\begin{aligned}
&f_{\overline{n}+i} := f(t_{\overline{n}+i}, y_{\overline{n}+i}) = g_{\overline{n}+i}\\
&+ \int_0^{t_{\overline{n}+i}} K(t_{\overline{n}+i}, \tau, y(\tau))d\tau, \quad i = 1, \cdots, k,\\
&= g_{\overline{n}+i} + \sum_{j=1}^{n} \int_{t_{(j-1)k}}^{t_{jk}} K(t_{\overline{n}+i}, \tau, y(\tau))d\tau\\
&+ \int_{t_{\overline{n}}}^{t_{\overline{n}+i}} K(t_{\overline{n}+i}, \tau, y(\tau))d\tau\\
&= g_{\overline{n}+i} + h\sum_{j=1}^{n} \int_0^k K(t_{\overline{n}+i}, t_{\overline{j}} + sh, y(t_{\overline{j}} + sh))ds\\
&+h \int_0^i K(t_{\overline{n}+i}, t_{\overline{n}+sh}, y(t_{\overline{n}} + sh)ds,
\end{aligned}$$

where $g_{\overline{n}+i} := g(t_{\overline{n}+i}, y_{\overline{n}+i})$. The integrals on the subintervals $[0, k]$ and $[0, i]$ are approximated by the integration formula with the weights b_ν, $\omega_{i,\nu}$ $\nu, i = 1, 2, \cdots, k$ for integrations in the subintervals $[0, k]$ and $[0, i]$, respectively and nodes $1, 2, \cdots, k$ in the form

$$\begin{aligned}
\textstyle\int_0^k p(s)ds &= \sum_{l=1}^{k} b_l p(l),\\
\textstyle\int_0^i p(s)ds &= \sum_{l=1}^{k} \omega_{i,l} p(l).
\end{aligned} \tag{2.7}$$

These quadrature formulas are specified by the vector and matrix of weights

$$W = \begin{bmatrix} \omega_{11} & \cdots & \omega_{1k} \\ \vdots & \ddots & \vdots \\ \omega_{k1} & \cdots & \omega_{kk} \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}.$$

Thus the approximation $\hat{f}_{\overline{n}+i}$ to $f_{\overline{n}+i}$ takes the following form

$$\begin{aligned}
\hat{f}_{\overline{n}+i} = g_{\overline{n}+i} &+ h \sum_{j=1}^{n} \sum_{l=1}^{k} b_l K(t_{\overline{n}+i}, t_{\overline{j}+l}, y_{\overline{j}+l})\\
&+h \sum_{l=1}^{k} \omega_{i,l} K(t_{\overline{n}+i}, t_{\overline{n}+l}, y_{\overline{n}+l}).
\end{aligned}$$

3. Derivation of the order condition

In this section we derive order conditions for the method (2.6) with k steps, assuming the order p. We assume that the components of the known vector $Y_{\overline{n}}$ satisfy

$$(Y_{\overline{n}})_i = y(t_{\overline{n}-k+i}) + O(h^{p+1}), \qquad i = 1, 2, \cdots, k. \tag{3.1}$$

We then request that

$$(Y_{\overline{n+1}})_i = y(t_{\overline{n}+i}) + O(h^{p+1}), \qquad i = 1, 2, \cdots, k. \tag{3.2}$$

We also assume that the quadrature formula (2.7) are of order $p-1$ and the components of the $\hat{f}_{\overline{n}}$ satisfy in

$$(\hat{f}_{\overline{n}})_i = f(t_{\overline{n}+i}) + O(h^{p+1}), \qquad i = 1, 2, \cdots, k. \tag{3.3}$$

Let us define the matrices

$$V_1^{(p)} = \begin{bmatrix} 1 & 1 & \frac{1}{2!} & \cdots & \frac{1}{p!} \\ 1 & 2 & \frac{2^2}{2!} & \cdots & \frac{2^p}{p!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & k & \frac{k^2}{2!} & \cdots & \frac{k^p}{p!} \end{bmatrix} \in \mathbb{R}^{k\times(p+1)},$$

$$V_2^{(p)} = \begin{bmatrix} 1 & (-k+1) & \frac{(-k+1)^2}{2!} & \cdots & \frac{(-k+1)^p}{p!} \\ 1 & (-k+2) & \frac{(-k+2)^2}{2!} & \cdots & \frac{(-k+2)^p}{p!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & -1 & \frac{(-1)^2}{2!} & \cdots & \frac{(-1)^p}{p!} \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{k\times(p+1)},$$

$$V_3^{(p)} = \begin{bmatrix} 0 & 1 & 1 & \frac{1}{2!} & \cdots & \frac{1}{(p-1)!} \\ 0 & 1 & 2 & \frac{2^2}{2!} & \cdots & \frac{2^{p-1}}{(p-1)!} \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 1 & k & \frac{k^2}{2!} & \cdots & \frac{k^{p-1}}{(p-1)!} \end{bmatrix} \in \mathbb{R}^{k\times(p+1)}.$$

Now we have the following theorem:

Theorem 3.1. *Assume that $Y_{\overline{n}}$ satisfy (3.1) and quadrature formulas are such that (3.3) is satisfied. Then BBDF satisfy (3.2) if and only if*

$$A^{(1)}V_1^{(p)} = A^{(0)}V_2^{(p)} + B^{(1)}V_3^{(p)}.$$

Proof. Substituting (3.1)-(3.3) in (2.6), we obtain

$$A^{(1)}Y(t_{\overline{n+1}}) = A^{(0)}Y(t_{\overline{n}}) + hB^{(1)}F(t_{\overline{n+1}}) + O(h^{p+1}),$$

where

$$Y(t_{\overline{n+1}}) = [y(t_{\overline{n}+1}), y(t_{\overline{n}+2}), \cdots, y(t_{\overline{n}+k-1}), y(t_{\overline{n}+k})]^T \in \mathbb{R}^k,$$

$$F_{\overline{n+1}} = [f(t_{\overline{n}+1}), f(t_{\overline{n}+2}), \cdots, f(t_{\overline{n}+k-1}), f(t_{\overline{n}+k})]^T \in \mathbb{R}^k.$$

Expanding all entries of $Y(t_{\overline{n+1}})$, $Y(t_{\overline{n}})$ and $F(t_{\overline{n+1}})$ as Taylor series about the point t_n yields the result. □

Now, we analyze the condition on the quadrature formulas which guarantee the required accuracy.

Theorem 3.2. *Suppose that $K(t,\tau,y)$ is sufficiently smooth. Then (3.3) is satisfied if*

$$\begin{aligned} &\sum_{l=1}^{k} b_l l^j = \tfrac{k^{j+1}}{j+1}, \\ &\sum_{l=1}^{k} \omega_{i,l} l^j = \tfrac{i^{j+1}}{j+1}, \quad i,l = 1,2,\cdots,k \end{aligned} \tag{3.4}$$

Proof. The condition (3.3) will be satisfied if

$$\int_0^k p(s)ds = \sum_{l=1}^{k} b_l p(l) + O(h^p),$$

$$\int_0^i p(s)ds = \sum_{l=1}^{k} \omega_{i,l} p(l) + O(h^p), \quad i = 1,\cdots,k,$$

for sufficiently smooth function p. Expanding the functions p and p as Taylor series around $s_0 = 0$ and comparing corresponding terms up to order $p = k$, we obtain the system (3.4). This completes the proof. □

4. Linear stability analysis

In this section, we analyze the stability properties of the introduced methods with respect to the basic test equation [5,6,12]

$$y'(t) = g(t) + \xi y(t) + \eta \int_0^t y(\tau)d\tau, \quad t > 0, \qquad y(0) = y_0, \tag{4.1}$$

where $\xi, \eta \in \mathbb{C}$. The solution of (4.1) is stable if $Re(r_1) < 0$ and $Re(r_2) < 0$ where $r_{1,2} = (\xi \pm \sqrt{\xi^2 + 4\eta})/2$ (see [2]). We observe that, particularly for real ξ and η, these conditions reduce to $\xi < 0$ and $\eta < 0$. As usual, we look for sufficient conditions for the stability of the numerical solution of (4.1).

Definition 4.1. We set $w = \xi h$ and $z = \eta h^2$. The absolute stability region is the set R of all the pairs $(z; w) \in \mathbb{C}^- \times \mathbb{C}^-$ such that the numerical solution y_n of test equation (4.1) with a fixed stepsize h, tends to zero as $n \to \infty$. The method is A_0-stable if $R \supseteq \mathbb{R}^- \times \mathbb{R}^-$ and is A-stable if it is stable for any value of (z,w) such that $Re(r_1) < 0$ and $Re(r_2) < 0$. An A-stable method is A_0-stable too.

Theorem 4.2. *The discretized BBDF, applied to the test equation (4.1), leads to the following recurrence relation*

$$\begin{bmatrix} \boldsymbol{Y}_{\overline{n+1}} \\ \boldsymbol{Z}_{\overline{n+1}} \end{bmatrix} = R(z,w) \begin{bmatrix} \boldsymbol{Y}_{\overline{n}} \\ \boldsymbol{Z}_{\overline{n}} \end{bmatrix} + h\overline{\boldsymbol{G}}_{\overline{n+1}},$$

where

$$R(z,w) = [Q(z,w)]^{-1}M(z,w),$$

and

$$Q(z,w) = \left[\begin{array}{c|c} A^{(1)} - zB^{(1)}W - wB^{(1)} & \boldsymbol{0}_{k\times k} \\ \hline -\boldsymbol{I}_k & \boldsymbol{I}_k \end{array}\right],$$

$$M(z,w) = \left[\begin{array}{c|c} A^{(0)} & zB^{(1)}Q \\ \hline \boldsymbol{0}_{k\times k} & \boldsymbol{I}_k \end{array}\right]$$

and

$$\overline{\boldsymbol{G}}_{\overline{n+1}} = [Q(z,w)]^{-1}\left[\begin{array}{c} G_{\overline{n+1}} \\ \hline \boldsymbol{0}_k \end{array}\right],$$

with

$$Q = \begin{bmatrix} b^T \\ \vdots \\ b^T \end{bmatrix} \in R^{k\times k}.$$

Proof. By considering the test problem (4.1) and $\hat{F}_{\overline{n+1}}$ as the approximation of the $F_{\overline{n+1}}$ and by using the quadrature formulas we have

$$\hat{f}_{\overline{n}+i} = g_{\overline{n}+i} + \xi y_{\overline{n}+i} + h\eta \sum_{j=0}^{n-1}\sum_{l=1}^{k} b_l y_{\overline{j}+l} + h\eta \sum_{l=1}^{k} \omega_{i,l} y_{\overline{n}+l},$$

which can be represented in the matrix form

$$\hat{F}_{\overline{n+1}} = G_{\overline{n+1}} + \xi Y_{\overline{n+1}} + h\eta Q \sum_{j=1}^{n} Y_{\overline{j}} + h\eta W Y_{\overline{n+1}}. \tag{4.2}$$

Now, by setting $Z_{\overline{n}} = \sum_{j=1}^{n} Y_{\overline{j}}$, $\xi h = w$ and $\eta h^2 = z$ and substituting (4.2) in (2.6) we obtain

$$\begin{cases} A^{(1)}Y_{\overline{n+1}} = A^{(0)}Y_{\overline{n}} + B^{(1)}\left(hG_{\overline{n+1}} + wY_{\overline{n+1}} + zQZ_{\overline{n}} + zWY_{\overline{n+1}}\right), \\ Z_{\overline{n+1}} = Y_{\overline{n+1}} + Z_{\overline{n}}. \end{cases}$$

These relations can be written in the matrix form

$$\left[\begin{array}{c|c} A^{(1)} - wB^{(1)} - zB^{(1)}W & \mathbf{0}_{k\times k} \\ \hline -\mathbf{I}_k & \mathbf{I}_k \end{array}\right]\begin{bmatrix} Y_{\overline{n+1}} \\ \\ Z_{\overline{n+1}} \end{bmatrix} = \left[\begin{array}{c|c} A^{(0)} & zB^{(1)}Q \\ \hline \mathbf{0}_{k\times k} & \mathbf{I}_k \end{array}\right]\begin{bmatrix} Y_{\overline{n}} \\ \\ Z_{\overline{n}} \end{bmatrix} + \left[\begin{array}{c} hG_{\overline{n+1}} \\ \hline \mathbf{0}_k \end{array}\right],$$

and this completes the proof. $\square$

$R(z,w)$ is called the stability matrix of the method. Now, the method is stable if $\rho(R(z,w)) < 1$. Hence, the stability region of the method is $R = \{(z,w) \in \mathbb{C}\times\mathbb{C} : \rho(R(z,w) < 1\}$. Here, the term $\overline{G}_n$ does not influence stability. The stability function of the method with respect to (4.1) is then defined as

$$p(z,w;\lambda) = \det(\lambda\mathbf{I}_{2k} - R(z,w)). \tag{4.3}$$

To investigate the stability properties of the BBDF, it is more convenient to work with the polynomial obtained by multiplying the stability function (4.3) by its denominator. The resulting polynomial will be denoted by the same symbol $p(z, w; \lambda)$. This polynomial takes the form

$$p(z, w; \lambda) = \sum_{i=0}^{2k} p_i(z, w)\lambda^i, \tag{4.4}$$

where $p_i(z, w)$, $i = 0, 1, \dots, 2k$ are polynomials of degree less than or equal to k. Denoting the roots of the polynomial $p(z, w; \lambda)$ by $\lambda_1, \lambda_2, \dots, \lambda_{2k}$, the absolute stability region of the method is then defined by

$$\mathcal{R} = \{(z, w) \in \mathbb{C}^- \times \mathbb{C}^- : |\lambda_i(z, w)| < 1, \quad i = 1, 2, \dots, 2k\}.$$

5. Examples of methods

Now we describe some classes of BBDF methods. We analyze the order conditions (3.4) and the stability properties with respect to test equation (4.1) with $z < 0$, $w < 0$ and find classes of A_0-stable methods.

Example 1. Two-step method with $k = 2$. The method is defined by

$$\begin{bmatrix} \frac{-2}{3} & 0 \\ \frac{-4}{3} & 1 \end{bmatrix} \begin{bmatrix} y_{2n+1} \\ y_{2n+2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-2}{3} \\ 0 & \frac{-1}{3} \end{bmatrix} \begin{bmatrix} y_{2n-1} \\ y_{2n} \end{bmatrix} + h \begin{bmatrix} -1 & \frac{+1}{3} \\ 0 & \frac{2}{3} \end{bmatrix} \begin{bmatrix} f_{2n+1} \\ f_{2n+2} \end{bmatrix}$$

This method is A_0-stable method of order 2 and the weights of the numerical integration formula of order 2 are in the form

$$W = \begin{bmatrix} \frac{3}{2} & \frac{-1}{2} \\ 2 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

Example 2. Three-step method with $k = 3$. The method is defined by

$$\begin{bmatrix} \frac{4}{11} & \frac{-8}{11} & 0 \\ \frac{28}{22} & \frac{-23}{22} & 0 \\ \frac{-8}{11} & \frac{-6}{11} & 1 \end{bmatrix} \begin{bmatrix} y_{3n+1} \\ y_{3n+2} \\ y_{3n+3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{-4}{11} \\ 0 & 0 & \frac{5}{22} \\ 0 & 0 & \frac{-3}{11} \end{bmatrix} \begin{bmatrix} y_{3n-2} \\ y_{3n-1} \\ y_{3n} \end{bmatrix} + h \begin{bmatrix} -1 & 0 & \frac{-1}{11} \\ 0 & -1 & \frac{4}{22} \\ 0 & 0 & \frac{24}{11} \end{bmatrix} \begin{bmatrix} f_{3n+1} \\ f_{3n+2} \\ f_{3n+3} \end{bmatrix}.$$

This method is of order 3 with extensive stability region and the stability region of 3-step methods is plotted in Figure 1. The weights of the numerical integration formula of order 3 are

$$W = \begin{bmatrix} \frac{23}{12} & \frac{-4}{3} & \frac{5}{12} \\ \frac{7}{3} & \frac{-2}{3} & \frac{1}{3} \\ \frac{9}{4} & 0 & \frac{3}{4} \end{bmatrix}, \quad b = \begin{bmatrix} \frac{9}{4} \\ 0 \\ \frac{3}{4} \end{bmatrix}.$$

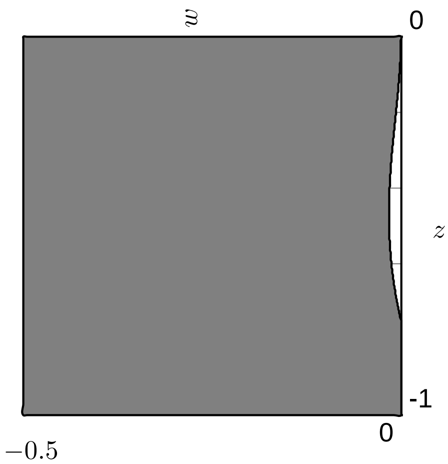

FIGURE 1. The stability region for 3-step method.

Example 3. four-step method with $k = 4$. The method is defined by

$$\begin{bmatrix} \frac{39}{50} & \frac{-69}{50} & \frac{17}{50} & 0 \\ \frac{18}{25} & \frac{-3}{25} & \frac{-38}{75} & 0 \\ \frac{-33}{50} & \frac{93}{50} & \frac{-197}{150} & 0 \\ \frac{-16}{25} & \frac{36}{25} & \frac{-48}{25} & 1 \end{bmatrix} \begin{bmatrix} y_{\overline{n}+1} \\ y_{\overline{n}+2} \\ y_{\overline{n}+3} \\ y_{\overline{n}+4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{-13}{50} \\ 0 & 0 & 0 & \frac{7}{75} \\ 0 & 0 & 0 & \frac{-17}{150} \\ 0 & 0 & 0 & \frac{-3}{25} \end{bmatrix} \begin{bmatrix} y_{4n-3} \\ y_{4n-2} \\ y_{4n-1} \\ y_{4n} \end{bmatrix}$$
$$+h \begin{bmatrix} -1 & 0 & 0 & \frac{2}{50} \\ 0 & -1 & 0 & \frac{-3}{75} \\ 0 & 0 & -1 & \frac{3}{25} \\ 0 & 0 & 0 & \frac{12}{25} \end{bmatrix} \begin{bmatrix} f_{4n+1} \\ f_{4n+2} \\ f_{4n+3} \\ f_{4n+4} \end{bmatrix}.$$

The integration formula are declared by the weights

$$W = \begin{bmatrix} \frac{55}{24} & \frac{-59}{24} & \frac{37}{24} & \frac{-3}{8} \\ \frac{8}{3} & \frac{-5}{3} & \frac{4}{3} & \frac{-1}{3} \\ \frac{21}{8} & \frac{-9}{8} & \frac{15}{8} & \frac{-3}{8} \\ \frac{8}{3} & \frac{-4}{3} & \frac{8}{3} & 0 \end{bmatrix}, \quad b = \begin{bmatrix} \frac{8}{3} \\ \frac{-4}{3} \\ \frac{8}{3} \\ 0 \end{bmatrix}.$$

6. Numerical examples

We illustrate the performances of discretized BBDF by means of some test examples. We verify the theoretical order of convergence established in Section 3.

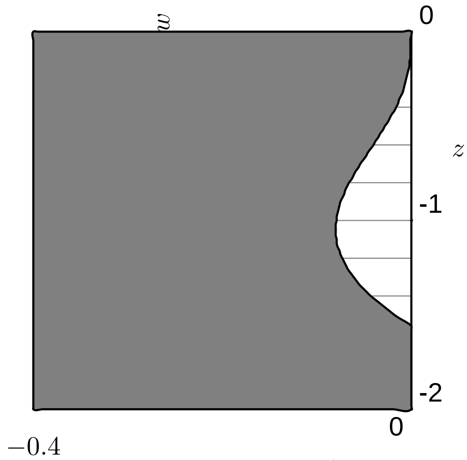

FIGURE 2. The stability region of 4-step method.

We consider the following test equations:
I. linear test equation

$$y'(t) = 1 + 2t - y(t) + \int_0^t \tau(1+2\tau)e^{\tau(t-\tau)}y(\tau)d\tau, \qquad t \in [0,2],$$
$$y(0) = 1,$$

with the exact solution $y(t) = e^{t^2}$.
II. nonlinear test equation

$$y'(t) = -\sin(t) - 2\frac{t}{e} + 2te^{-y} + \int_0^t -2t\sin(\tau)\ e^{-y}d\tau, \qquad t \in [0,1],$$
$$y(0) = 1,$$

with the exact solution $y(t) = \cos(t)$.
III. stiff test equation

$$y'(t) = \lambda(y(t) - \sin(t)) + 1 - \int_0^t y(\tau)d\tau, \qquad t \in [0, \frac{3\pi}{4}],$$
$$y(0) = 0,$$

where the exact solution $y(t) = \sin(t)$ and with $\lambda < 0$. Differentiating of this problem leads to the second order ODE which can be written in the form of system of first order ODE with eigenvalues λ_1, λ_2. It is equivalent to a system of ODEs of Prothero-Robinson type, with $|\frac{\lambda_1}{\lambda_2}| = O(\lambda^2)$, and it is stiff for large values of $|\lambda|$. We handle it with $\lambda = -10^6$. We have implemented the methods with a fixed stepsize $h = \frac{T}{2^m}$, with several integer values of k. In the following tables, the maximal end point error is written as 10^{-cd}, where cd is the number of correct significant digits. Also, a numerical estimation of the order of convergence of the methods is computed by the formula $p(h) = \log_2(\frac{e(2h)}{e(h)})$, where $e(h)$ is the maximal absolute end point error.

The results in Tables 1-3 confirm the proved convergence order.

TABLE 1. The results of problem I.

	m	2	3	4	5	6	7
3-step method	cd	0.21	1.64	2.74	3.75	4.71	5.63
	$p(h)$		4.76	3.68	3.33	3.17	3.08
4-step method	cd	1.02	2.66	4.11	5.43	6.71	5.63
	$p(h)$		5.42	4.79	4.41	4.21	4.11
6-step method	cd	2.37	4.19	6.00	7.81	9.62	12.36
	$p(h)$		6.05	6.01	6.00	6.00	6.00

TABLE 2. The results of problem II.

	m	2	3	4	5	6	7
3-step method	cd	4.44	5.38	6.31	7.22	8.12	9.03
	$p(h)$		3.12	3.05	3.03	3.02	3.01
4-step method	cd	3.91	5.19	6.41	7.64	8.86	10.08
	$p(h)$		4.25	4.06	4.09	4.07	4.04

TABLE 3. The results of problem III.

	m	2	3	4	5	6
4-step method	cd	1.21	2.33	3.49	4.96	5.89
	$p(h)$		3.72	3.85	3.92	3.96
6-step method	cd	2.81	4.62	6.43	8.22	10.03
	$p(h)$		6.05	6.01	6.00	6.00

REFERENCES

[1] O.A. Akinfenwa, S.N. Jator and N.M. Yao, *Continuous block backward differentiation formula for solving ordinary differential equations*, Comput. and math. with appl. *65*, (2013), 996–1005.

[2] C.T.H. Baker, A. Makroglou and E. Short,*Regions of stability in the numerical treatment of Volterra integro-differential equations*, SIAM. J. Numer. Anal. *16*, (1979), 890–910.

[3] J.E. Bond and J.R. Cash, *A block method for the numerical integration of stiff sytems of ODEs*, BIT, *19*, (1979), 429–447.

[4] H. Brunner and P. J. Van der Houwen, *The Numerical solution of Volterra equations*, CWI monographs, *3*, North Holland, Amesterdam, 1986.

[5] H. Brunner, *Collocation methods for Volterra integral and related functional equations*, Cambridge University Press, 2004.

[6] H. Brunner and J. D. Lambert, *Stability of numerical methods for Volterra integro-differential equations*, Computing (Arch. Elektron. Rechnen) *12*, no. 1, (1974), 75–89.

[7] C. W. Gear, *Numerical initial value problems in ordinary differential equations,* COMM. ACM, *14*, (1971), 185–190.

[8] A. Makroglou, *Integral equations and actuarial risk management: Some models and numerics*, Mathematical Modelling and Analysis, *8*, (2003), 143-154.

[9] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, Wiley, New York, 1991.
[10] T. Lin, Y. Lin, M. Rao and S. Zhang, *Petrov-Galerkin methods for linear Volterra integro-differential equations*, SIAM J. Numer. Anal. *38*, (2000), 937-963.
[11] P. Linz, *Linear multistep methods for Volterra integro-differential equations*, J. Ass. comput. Mach. *16*, (1969), 295–301.
[12] J. Matthys, *A-stable linear multistep methods for Volterra integro-differential equations*, Numer. Math. *27*, (1976/77), 85–94.
[13] G. Mehdiyeva, M. Imanova and V. Ibrahimov, *Application of the Hybrid methods to solving Volterra integro-differential equations,* World Academy of Science, Engineering and Technology *77*, 2011.
[14] W. E. Milne, *Numerical solution of differential equations*, John Wiley and Sons, 1953.
[15] V. Volterra, *Theory of functional and of integral and integro-differential equations,* Moscow, Nauka, 1982.
[16] P. H. M. Wolkenfelt, *The Construction of Reducible Quadrature Rules for Volterra Integral and Integro-differential Equations,* IMA Journal of Numerical Analysis, *2*, (1982), 131–152.