Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 4, No. 3, 2016, pp. 217-229

A hybrid method with optimal stability properties for the numerical solution of stiff differential systems

Akram Movahedinejad
Faculty of Mathematical Sciences,
University of Tabriz, Tabriz, Iran.
E-mail: a_movahedinejad@tabrizu.ac.ir

Ali Abdi*
Faculty of Mathematical Sciences,
University of Tabriz, Tabriz, Iran.
E-mail: a_abdi@tabrizu.ac.ir

Gholamreza Hojjati
Faculty of Mathematical Sciences,
University of Tabriz, Tabriz, Iran.
E-mail: ghojjati@tabrizu.ac.ir

Abstract In this paper, we consider the construction of a new class of numerical methods based on the backward differentiation formulas (BDFs) that be equipped by including two off–step points. We represent these methods from general linear methods (GLMs) point of view which provides an easy process to improve their stability properties and implementation in a variable stepsize mode. These superiorities are confirmed by the numerical examples.

Keywords. Backward differentiation formula, Hybrid methods, General linear methods, A– and $A(\alpha)$–stability, Variable stepsize implementation.

2010 Mathematics Subject Classification. 65L05.

1. Introduction

We shall be concerned with the construction of numerical methods for solving stiff autonomous ordinary differential equations (ODEs) of the form

$$y'(x) = f(y(x)), \quad y(x_0) = y_0, \tag{1.1}$$

on the finite interval $I := [x_0, \overline{x}]$ where $y : I \to \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^m$ are continuous and differentiable, and m is the dimensionality of the system. Backward differentiation formulas (BDFs) [12] are the standard and popular methods in the class of multivalue methods with the form

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

Received: 23 November 2016 ; Accepted: 3 January 2017.
* Corresponding author.

Theses methods are $A(\alpha)$–stable up to order $p = k = 6$ [21]. Many codes have been introduced for solving stiff initial value problems based on BDFs with good accuracy and reasonably wide region of absolute stability. Some extensions of BDFs were introduced by using the future points technique such as EBDF (extended BDF) [6], MEBDF (modified EBDF) [4], MF–MEBDF (Matrix free MEBDF) [18], A–BDF method [11] and A–EBDF method [17], by using the higher derivatives of the solutions technique such as SDBDF [5] and MESDMM [16], and also by using the off–step points such as methods introduced in [8, 9, 13].

In this paper, we first construct a modification of BDF which applies two off–step points technique and represent it from general linear methods (GLMs) point of view with the aim of obtaining extensive absolute stability regions. Also, we implement A–stable methods of this category for solving stiff ODEs in a variable stepsize environment.

The rest of the paper is organized as follows: In Section 2, we construct hybrid BDFs (HBDFs) including two off-step points with the aim of achieving the maximum value of α of $A(\alpha)$-stability. In section 3, implementation of the methods using variable stepsize technique are investigated. Finally, in Section 4, some results of numerical experiments are presented to confirm the theoretical results.

2. Construction of HBDFs with extended stability region

In this section, we first consider a new class of numerical methods for ODEs which are based on BDFs by using two off-step points. Then, we present these methods in the form of general linear methods to obtain extensive absolute stability regions.

2.1. **HBDFs with two off-step points.**

Assume that the solution of (1.1) has desired continuous derivatives in the interval $[x_0, \overline{x}]$. We introduce a new class of HBDFs with two off-step points as follows

$$y_{n+k} = \sum_{j=1}^{k} \bar{\alpha}_j y_{n+k-j} + h\bar{\beta}_1 f_{n+k-\theta} + h\bar{\beta}_2 f_{n+k+\eta} + h\bar{\beta}_3 f_{n+k}, \tag{2.1}$$

where $x_n = x_0 + nh$, $n = 1, 2, \cdots$, h is the stepsize, $0 < \theta < 1$, $\eta > 0$, $f_{n+k} = f(y_{n+k})$, $f_{n+k-\theta} = f(y_{n+k-\theta})$ and $f_{n+k+\eta} = f(y_{n+k+\eta})$. The coefficients $\bar{\alpha}_j$, $j = 1, 2, \dots, k$ and $\bar{\beta}_j$, $j = 1, 2, 3$, are computed by solving the appropriate order conditions for the order $p = k + 1$. In these methods, we need the first derivative of the solution $y(x)$ in two off–step points, i.e., $x_{n+k-\theta}$ and $x_{n+k+\eta}$. Assuming that the values of y_n, y_{n+1}, ..., y_{n+k-1} are available, the method (2.1) is used in practice with applying two predictors that take the forms

$$y_{n+k-\theta} = \sum_{j=1}^{k} \tilde{\alpha}_j y_{n+k-j} + h\tilde{\beta}_1 f_{n+k-\theta}, \tag{2.2}$$

and

$$y_{n+k+\eta} = \sum_{j=1}^{k} \hat{\alpha}_j y_{n+k-j} + h\hat{\beta}_1 f_{n+k-\theta} + h\hat{\beta}_2 f_{n+k+\eta}, \tag{2.3}$$

where the coefficients are chosen such that equations (2.2) and (2.3) have order k. Then the approach goes as follows:

stage 1.: Use the predicator (2.2) to compute $\bar{y}_{n+k-\theta}$

$$\bar{y}_{n+k-\theta} = \sum_{j=1}^{k} \tilde{\alpha}_j y_{n+k-j} + h\tilde{\beta}_1 \bar{f}_{n+k-\theta}, \tag{2.4}$$

where $\bar{f}_{n+k-\theta} = f(\bar{y}_{n+k-\theta})$.

stage 2.: Use the predicator (2.3) to compute $\bar{y}_{n+k+\eta}$

$$\bar{y}_{n+k+\eta} = \sum_{j=1}^{k} \hat{\alpha}_j y_{n+k-j} + h\hat{\beta}_1 \bar{f}_{n+k-\theta} + h\hat{\beta}_2 \bar{f}_{n+k+\eta}, \tag{2.5}$$

where $\bar{f}_{n+k+\eta} = f(\bar{y}_{n+k+\eta})$.

stage 3.: Compute y_{n+k} as the solution of

$$y_{n+k} = \sum_{j=1}^{k} \bar{\alpha}_j y_{n+k-j} + h\bar{\beta}_1 \bar{f}_{n+k-\theta} + h\bar{\beta}_2 \bar{f}_{n+k+\eta} + h\bar{\beta}_3 f_{n+k}. \tag{2.6}$$

We note that the parameters $\hat{\beta}_2$, $\bar{\beta}_3$ and two parameters which determine the position of two off step points, i.e. θ and η, remain as free parameters.

Now, we are going to prove that the new method (2.4)-(2.6) has order $p = k + 1$. We assume that the local truncation errors for (2.4), (2.3) and (2.1) are

$$y(x_{n+k-\theta}) - \bar{y}_{n+k-\theta} = \widetilde{C}_{k+1} h^{k+1} y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}), \tag{2.7}$$

$$y(x_{n+k+\eta}) - \bar{y}_{n+k+\eta} = \widehat{C}_{k+1} h^{k+1} y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}), \tag{2.8}$$

and

$$y(x_{n+k}) - y_{n+k} = C_{k+2} h^{k+2} y^{(k+2)}(x_n) + \mathcal{O}(h^{k+3}), \tag{2.9}$$

with the error constants $\widetilde{C}_{k+1}$, $\widehat{C}_{k+1}$ and C_{k+2}, respectively. Thus, we have the following theorem.

Theorem 2.1. *Assume that*

(1) *formula* (2.4) *is of order* k,
(2) *formula* (2.5) *is of order* k,
(3) *formula* (2.6) *is of order* $k+1$,

then, the method (2.4)–(2.6) *has order* $k+1$.

Proof. Suppose that the values y_n, y_{n+1}, $\ldots$, y_{n+k-1} be exact. From (2.4) we have

$$y(x_{n+k-\theta}) - \bar{y}_{n+k-\theta} = \widetilde{C}_{k+1} h^{k+1} y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}). \tag{2.10}$$

Since in (2.5) we apply $\bar{y}_{n+k-\theta}$, the error of $y(x_{n+k-\theta}) - \bar{y}_{n+k-\theta}$ must be added to the (2.8). Hence, using the mean value theorem and (2.10) yield

$$\begin{aligned} y(x_{n+k+\eta}) - \bar{y}_{n+k+\eta} &= h\hat{\beta}_1\big(f(y(x_{n+k-\theta})) - f(\bar{y}_{n+k-\theta})\big) \\ &\quad + \widehat{C}_{k+1}h^{k+1}y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}) \\ &= h\hat{\beta}_1\frac{\partial f}{\partial y}(\tau)\big(y(x_{n+k-\theta}) - \bar{y}_{n+k-\theta}\big) \\ &\quad + \widehat{C}_{k+1}h^{k+1}y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}) \\ &= h\hat{\beta}_1\frac{\partial f}{\partial y}(\tau)(\widetilde{C}_{k+1}h^{k+1}y^{(k+1)}(x_n)) \\ &\quad + \widehat{C}_{k+1}h^{k+1}y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}) \\ &= \widehat{C}_{k+1}h^{k+1}y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}), \end{aligned}$$

where τ is a point in the interval whose endpoints are $\bar{y}_{n+k-\theta}$ and $y(x_{n+k-\theta})$. Similarly, from (2.6) and by considering the errors of $y(x_{n+k-\theta})-\bar{y}_{n+k-\theta}$ and $y(x_{n+k+\eta})-\bar{y}_{n+k+\eta}$ to the expression of (2.9), we have

$$\begin{aligned} y(x_{n+k}) - y_{n+k} &= h\bar{\beta}_1\Big(f(y(x_{n+k-\theta})) - f(\bar{y}_{n+k-\theta})\Big) + h\bar{\beta}_2\Big(f(y(x_{n+k+\eta})) \\ &\quad - f(\bar{y}_{n+k+\eta})\Big) + C_{k+2}h^{k+2}y^{(k+2)}(x_n) + \mathcal{O}(h^{k+3}) \\ &= h\bar{\beta}_1\frac{\partial f}{\partial y}(\tau_1)\big(y(x_{n+k-\theta}) - \bar{y}_{n+k-\theta}\big) + h\bar{\beta}_2\frac{\partial f}{\partial y}(\tau_2)\Big(y(x_{n+k+\eta}) \\ &\quad - \bar{y}_{n+k+\eta}\Big) + C_{k+2}h^{k+2}y^{(k+2)}(x_n) + \mathcal{O}(h^{k+3}) \\ &= h^{k+2}\Big(C_{k+2}y^{(k+2)}(x_n) + \big(\widetilde{C}_{k+1}\bar{\beta}_1\frac{\partial f}{\partial y}(\tau_1) + \\ &\quad \widehat{C}_{k+1}\bar{\beta}_2\frac{\partial f}{\partial y}(\tau_2)\big)y^{(k+1)}(x_n)\Big) + \mathcal{O}(h^{k+3}). \end{aligned}$$

Here, τ_1 and τ_2 are points in the interval whose endpoints are $y_{n+k-\theta}$ and $y(x_{n+k-\theta})$, for τ_1, and $y_{n+k+\eta}$ and $y(x_{n+k+\eta})$, for τ_2. So the order of overall method is $k+1$. □

2.2. **HBDFs with two off–step points as GLMs.** In this subsection, at first, we express HBDFs with two off–step points from GLMs point of view. Then by using this representation, we obtain the free parameters such that the method has low implementation cost and more extensive absolute stability region.

GLMs [1, 2, 3, 20] in each step import r quantities from the previous step and export the same number of quantities of order p to use in the following step that denoted by $y^{[n-1]} = [y_i^{[n-1]}]_{i=1}^r$ and $y^{[n]} = [y_i^{[n]}]_{i=1}^r$, respectively. GLMs have s stages that these values are collected in a vector denoted by $Y^{[n]} = [Y_i^{[n]}]_{i=1}^s$ and the first derivative of stage vector is $f(Y^{[n]}) = [f(Y_i^{[n]})]_{i=1}^s$. The stage values are approximations to the solution at $x_{n-1}+c_ih$ of order q, i.e.,

$$Y_i = y(x_{n-1}+c_ih)+\mathcal{O}(h^{q+1}),$$

where $c=[c_1,\, c_2,\, \dots,\, c_s]^T$ is abscissae vector. The quantities imported and evaluated in step number n are related to

$$\begin{aligned} Y^{[n]} &= h(A\otimes I_m)f(Y^{[n]})+(U\otimes I_m)y^{[n-1]},\\ y^{[n]} &= h(B\otimes I_m)f(Y^{[n]})+(V\otimes I_m)y^{[n-1]}, \end{aligned} \tag{2.11}$$

where $n=1,\,2,\,...,\,N$, $Nh=\overline{x}-x_0$, and $\otimes$ is the Kronecker product of two matrices. Here, $A\in\mathbb{R}^{s\times s}$, $U\in\mathbb{R}^{s\times r}$, $B\in\mathbb{R}^{r\times s}$ and $V\in\mathbb{R}^{r\times r}$ are coefficients matrices of a GLM.

Now it can be verified that the algorithm based on formulas (2.4)–(2.6) can be written as a GLM of the form (2.11) with $s=3$, $r=k$ and with the vectors $Y^{[n]}$, $f(Y^{[n]})$ and $y^{[n]}$ as

$$Y^{[n]}=\begin{bmatrix}\bar{y}_{n+k-\theta}\\ \bar{y}_{n+k+\eta}\\ y_{n+k}\end{bmatrix},\quad f(Y^{[n]})=\begin{bmatrix}\bar{f}_{n+k-\theta}\\ \bar{f}_{n+k+\eta}\\ f_{n+k}\end{bmatrix},\quad y^{[n]}=\begin{bmatrix}y_{n+k}\\ y_{n+k-1}\\ \vdots\\ y_{n+1}\end{bmatrix}.$$

The coefficient matrices A, U, B and V are given by

$$A=\begin{bmatrix}\tilde{\beta}_1 & 0 & 0\\ \hat{\beta}_1 & \hat{\beta}_2 & 0\\ \bar{\beta}_1 & \bar{\beta}_2 & \bar{\beta}_3\end{bmatrix},\quad U=\begin{bmatrix}\tilde{\alpha}_1 & \tilde{\alpha}_2 & \cdots & \tilde{\alpha}_k\\ \hat{\alpha}_1 & \hat{\alpha}_2 & \cdots & \hat{\alpha}_k\\ \bar{\alpha}_1 & \bar{\alpha}_2 & \cdots & \bar{\alpha}_k\end{bmatrix},$$

$$B=\begin{bmatrix}\bar{\beta}_1 & \bar{\beta}_2 & \bar{\beta}_3\\ 0 & 0 & 0\\ \vdots & \vdots & \vdots\\ 0 & 0 & 0\end{bmatrix},\quad V=\begin{bmatrix}\bar{\alpha}_1 & \bar{\alpha}_2 & \cdots & \bar{\alpha}_{k-1} & \bar{\alpha}_k\\ 1 & 0 & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & 0\end{bmatrix},$$

where $A\in\mathbb{R}^{3\times3}$, $U\in\mathbb{R}^{3\times k}$, $B\in\mathbb{R}^{k\times3}$, $V\in\mathbb{R}^{k\times k}$ and $c=[1-\theta,1+\eta,1]^T$.

The coefficient matrix A plays main role in determining the implementation cost of GLMs. So, to reduce implementation cost, we assume that $\hat{\beta}_2=\bar{\beta}_3=\tilde{\beta}_1$.

The stability behavior of GLMs is considered using the standard linear test problem $y'=\lambda y$ [7], where λ is a complex parameter with negative real part. Applying test problem to (2.11) and assuming the matrix $I-zA$, with $z=\lambda h$, is nonsingular, we get

$$y^{[n]}=M(z)y^{[n-1]},$$

where $M(z) = V + zB(I - zA)^{-1}U$ is the stability matrix and $p(w, z) = \det(wI - M(z))$ is the stability function of the method. The region of absolute stability of the method (2.11) is the subset of the complex plane

$$S = \{z \in \mathbb{C} : \text{all roots } w_i(z) \text{ of } p(w, z) \text{ are in the unit circle}\}.$$

To obtain the absolute stability region of the presented methods, the boundary locus method is used. We set $w = e^{i\nu}$, where i is the imaginary unit, with $\nu \in [0, 2\pi]$ and then by using $p(w, z) = 0$, we obtain three roots, $z_j(\theta, \eta)$, $j = 1, 2, 3$, that give us the boundary of the stability region for the given θ and η. We next define an objective function which approximates the negative value of the angle α for specific choices of parameters θ and η. Then by using fminsearch command from MATLAB, we minimize this objective function. The optimal values of θ and η (preserving zero stability) and the corresponding maximum value for α are listed and compared with those in BDFs in Table 1. For these values of θ and η, HBDFs are A–stable up to order 4 and $A(\alpha)$–stable up to order 10.

Table 1. The values of optimal θ, η to obtain maximum value of α.

	HBDF method				BDF method	
k	θ	η	p	α_{max}	p	α
1	0.5	0.5	2	90°	1	90°
2	0.1	0.5	3	90°	2	90°
3	0.01	1.8	4	90°	3	88°
4	0.1210	1.6030	5	88.56°	4	73°
5	0.1320	1.4155	6	84.05°	5	51°
6	0.1677	0.8321	7	77.31°	6	18°
7	0.0904	0.9	8	66.78°	–	–
8	0.2	0.9	9	61.45°	–	–
9	0.19	0.9	10	45.52°	–	–

3. Implementation of the methods in a variable stepsize mode

Deriving the representation of hybrid methods in the form of GLMs makes more convenient to turn them into Nordsieck form than previous representation. In this section, we derive a representation of these methods in the Nordsieck form and discuss about the implementation of them in a variable step size environment.

3.1. **Nordsieck representation of the HBDFs.** In this subsection, we derive the Nordsieck representation of HBDFs which makes easy to implement in a variable step size environment. Nordsieck representation is achieved by forcing the vectors of incoming and outgoing to directly approximate of the Nordsieck vector

$$z(x_n, h) := \begin{bmatrix} y(x_n) \\ hy'(x_n) \\ \vdots \\ h^p y^{(p)}(x_n) \end{bmatrix},$$

of order p. The number of incoming and outgoing approximations in the HBDFs in the GLMs form is $r = k$. For the vector $y^{[n]}$ of external vector approximates directly the Nordsieck vector $z(x_n, h)$, we add two extra independent components $hf_{n+k-\theta}$ and $hf_{n+k+\eta}$ to the output vector, i.e.,

$$y^{[n]} = \begin{bmatrix} y_{n+k} \\ y_{n+k-1} \\ \vdots \\ y_{n+1} \\ hf_{n+k-\theta} \\ hf_{n+k+\eta} \end{bmatrix}.$$

With this change the coefficient matrices U, B, and V take the following forms

$$U = \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 & \cdots & \tilde{\alpha}_k & 0 & 0 \\ \hat{\alpha}_1 & \hat{\alpha}_2 & \cdots & \hat{\alpha}_k & 0 & 0 \\ \bar{\alpha}_1 & \bar{\alpha}_2 & \cdots & \bar{\alpha}_k & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} \bar{\beta}_1 & \bar{\beta}_2 & \bar{\beta}_3 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} \bar{\alpha}_1 & \bar{\alpha}_2 & \cdots & \bar{\alpha}_{k-1} & \bar{\alpha}_k & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Here, $U \in \mathbb{R}^{3\times(k+2)}$, $B \in \mathbb{R}^{(k+2)\times 3}$ and $V \in \mathbb{R}^{(k+2)\times(k+2)}$.

For transforming the external vector to the Nordsieck vector, we define the relation

$$y_i^{[n]} = \sum_{j=1}^{p+1} t_{ij} z_i^{[n]},$$

where $z_i^{[n]}$, $i = 1, 2, ..., p+1$, denote the ith component of the Nordsieck vector. This transformation can be written more compactly as

$$y^{[n]} = Tz^{[n]}, \tag{3.1}$$

where

$$T = \begin{bmatrix} 1 & k & \frac{k^2}{2!} & \cdots & \frac{k^p}{p!} \\ 1 & k-1 & \frac{(k-1)^2}{2!} & \cdots & \frac{(k-1)^p}{p!} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & 1 & \frac{1}{2} & \cdots & \frac{1}{p!} \\ 0 & 1 & k-\theta & \cdots & \frac{(k-\theta)^{p-1}}{(p-1)!} \\ 0 & 1 & k+\eta & \cdots & \frac{(k+\eta)^{p-1}}{(p-1)!} \end{bmatrix}. \tag{3.2}$$

By substituting (3.1) into (2.11), we obtain

$$\begin{aligned} Y^{[n]} &= h(A \otimes I_m) f(Y^{[n]}) + (\mathbf{P} \otimes I_m) z^{[n-1]}, \\ z^{[n]} &= h(\mathbf{G} \otimes I_m) f(Y^{[n]}) + (\mathbf{Q} \otimes I_m) z^{[n-1]}, \end{aligned} \tag{3.3}$$

where the new coefficient matrices $\mathbf{P}$, $\mathbf{G}$ and $\mathbf{Q}$ are defined by

$$\mathbf{P} = UT, \qquad \mathbf{G} = T^{-1}B, \qquad \mathbf{Q} = T^{-1}VT.$$

So, (3.3) is the desired Nordsieck representation of the method.

3.2. **Variable stepsize formulation of HBDFs.** For the variable step formulation of the methods on a nonuniform grid

$$x_0 < x_1 < \cdots < x_N, \quad x_N \geq \overline{x},$$

the Nordsieck representation of the method takes the form

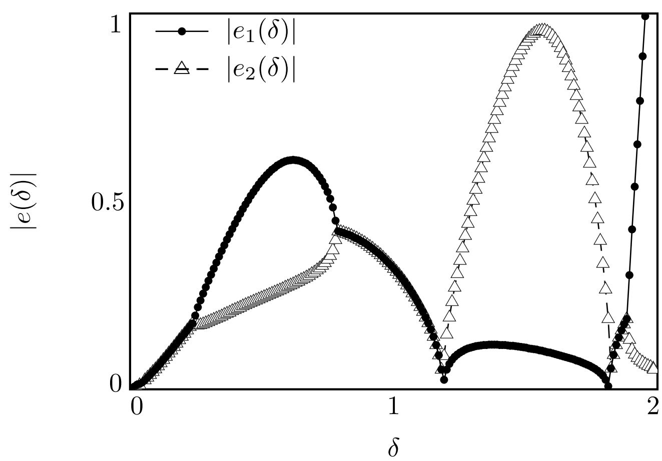
$$\begin{aligned} Y^{[n]} &= h_n(A \otimes I_m) f(Y^{[n]}) + (\mathbf{P}D(\delta_n) \otimes I_m) z^{[n-1]}, \\ z^{[n]} &= h_n(\mathbf{G} \otimes I_m) f(Y^{[n]}) + (\mathbf{Q}D(\delta_n) \otimes I_m) z^{[n-1]}, \end{aligned} \tag{3.4}$$

$n = 1, 2, ..., N$, where $h_n = x_n - x_{n-1}$. Here $Y^{[n]}$ is an approximation of $y(x_{n-1} + ch_n) = [y(x_{n-1} + c_i h_n)]_{i=1}^{s}$, $y^{[n]}$ is an approximation of order p to the Nordsieck vector $[h_n^{i-1} y^{(i-1)}(x_n)]_{i=1}^{p+1}$, and $D(\delta_n)$ is the rescaling matrix defined by

$$D(\delta_n) := \text{diag}\left(1, \delta_n, \delta_n^2, \ldots, \delta_n^p\right), \qquad \delta_n = h_n / h_{n-1}.$$

It follows from (3.4) that the zero–stability properties of method are determined by the eigenvalues of the matrix $\mathbf{Q}D(\delta)$. So, it is important to find a suitable bound for δ such that all the eigenvalues of $\mathbf{Q}D(\delta)$ in this bound be less than one. For method with $k = 1$, the eigenvalues of matrix $\mathbf{Q}D(\delta)$ are $1, 0, 0$ and so the method is zero stable for any variable step size pattern. For $k = 2$, the eigenvalues of $\mathbf{Q}D(\delta)$ are 1, 0, 0 and $e(\delta)$ where $e(\delta) = 1.7514771121\delta - 2.4336313368\delta^2 + 0.8603747259\delta^3$. For any given δ in the interval [0, 2.128764453], eigenvalue $e(\delta)$ is smaller than one. And for $k = 3$, the eigenvalues of $\mathbf{Q}D(\delta)$ are 1, 0, 0, $e_1(\delta)$ and $e_2(\delta)$. We have plotted the values of $e_1(\delta)$ and $e_2(\delta)$ for $\delta \in [0, 2]$ in Figure 1. For every δ in the interval [0, 1.949578008], eigenvalues $e_1(\delta)$ and $e_2(\delta)$ are smaller than one.

Figure 1. The plot of $|e(\delta)|$ versus δ.



For any method to be implemented in variable stepsize mode it must be estimated the local truncation error, as this allows a measure of how accurate the approximations are, and how much the stepsize should be varied. The local truncation error is given by

$$E_{p+1}h^{p+1}y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}),$$

where E_{p+1} is the error constant for the method of order p and determine with

$$p(\exp(z), z) = E_{p+1}z^{p+1} + \mathcal{O}(z^{p+2}). \tag{3.5}$$

Assuming h is sufficiently small, we approximate $h^{p+1}y^{(p+1)}(x_n)$ so that

$$\text{LTE}(x_n) = E_{p+1}h^{p+1}y^{(p+1)}(x_n), \tag{3.6}$$

can be calculated as an approximation to the local truncation error. It is possible to approximate the estimation of $h^{p+1}y^{(p+1)}(x_n)$ with a linear combination of the known stage derivatives, $hf(Y_i^{[n]})$, $i = 1, 2, 3$ and some components of the input vector. So an approximation to (3.6) can be represented by

$$est(x_n) := E_{p+1}\left(\sum_{i=1}^{3}\eta_{pi}hf(Y_i^{[n]}) + \sum_{i=1}^{p-2}\gamma_{pi}y_{i+3}^{[n-1]}\right).$$

For the method with $k = 1$ and $p = 2$, we have $E_3 = \frac{1}{24}$ and

$$\eta_{21} = 4, \quad \eta_{22} = 4, \quad \eta_{23} = -8.$$

For the method with $k = 2$ and $p = 3$, we have $E_4 = \frac{2608179043}{22786176000}$ and

$$\eta_{31} = \frac{500}{17}, \quad \eta_{32} = \frac{100}{17}, \quad \eta_{33} = -\frac{600}{17}, \quad \gamma_{31} = -\frac{15}{17}.$$

For the method with $k = 3$ and $p = 4$, we have $E_5 = 0.0564$ and

$$\eta_{41} = \frac{800000000}{9883867}, \quad \eta_{42} = \frac{40000000}{88954803}, \quad \eta_{43} = -\frac{40000000}{491463},$$

$$\gamma_{41} = -\frac{40000}{54607}, \quad \gamma_{42} = -\frac{191600}{163821}.$$

Now, we recall the implementation strategies which have been investigated in [19] to apply in our numerical experiments.

The used strategy to control the stepsize in the advancing from the step n to the step $n+1$ is according to the following control

$$\text{est}(x_n) \leq Rtol \cdot \max\{\|y_n\|, \|y_{n+1}\|\} + Atol, \tag{3.7}$$

where $Atol$ and $Rtol$ are given the absolute and relative tolerances. If the control (3.7) is not satisfied, the current step is repeated with the halved stepsize. Otherwise, the current step is accepted and we use the standard step control strategy (see [14]) as the following

$$h_{n+1} = \min\Big\{\Delta, \Big(\frac{\rho \cdot tol}{\|\text{est}(x_n)\|}\Big)^{\frac{1}{p+1}}\Big\} h_n.$$

Also, in our numerical experiments, we have used $Atol = Rtol = tol$, $\rho = 0.95$ and $\Delta = 2$ for $k = 1, 2$ and $\Delta = 1.8$ for $k = 3$.

4. Numerical experiments

In this section, the constructed methods are verified by some numerical experiments in a variable stepsize environment.

We consider the following test problems:

Problem 1. Non–linear stiff ordinary differential equation [10]

$$\begin{cases} y_1' = -0.04y_1 + 10^4 y_2 y_3 - 0.96\exp(-x), \\ y_2' = 0.04y_1 - 10^4 y_2 y_3 - 10^7 y_2^2 - 0.04\exp(-x), \\ y_3' = 3 \times 10^7 y_2^2 + \exp(-x), \end{cases}$$

with $y(0) = [1, 0, 0]^T$ and $x \in [0, 10^5]$.

Problem 2. The Van der Pol's oscillator problem described in [15]

$$\begin{cases} y_1' = y_2, \\ \varepsilon y_2' = \big(1 - y_1^2\big) y_2 - y_1, \end{cases}$$

where $\varepsilon = 10^{-6}$, $x \in [0,\ 2/3]$ and $y(0) = [2,\ -2/3]^T$.

Using the mentioned strategies given in section 3, we have implemented HBDF of orders 2, 3 and 4 in variable stepsize environment for solving Problems 1 and 2. In Tables 2-7, we have reported *ns* as the number of steps, *nrs* as the number of rejected steps, *nfe* as the number of function evaluations, *nJe* as the number of Jacobian evaluations and *ge* as the global error at the end of the interval of integration for different given tolerances, *tol*.

The numerical results confirm the capability and efficiency of the proposed methods.

Table 2. Numerical results for Problem 1 solved by the method of order 2 with $h_0 = 10^{-5}$.

tol	*ns*	*nrs*	*nfe*	*ge*
10^{-2}	36	0	512	1.59×10^{-2}
10^{-4}	50	0	740	1.21×10^{-3}
10^{-6}	133	0	1701	8.11×10^{-5}
10^{-8}	529	0	6367	4.19×10^{-6}

Table 3. Numerical results for Problem 1 solved by the method of order 3 with $h_0 = 10^{-5}$.

tol	*ns*	*nrs*	*nfe*	*ge*
10^{-2}	54	0	1438	1.75×10^{-2}
10^{-4}	66	0	950	1.24×10^{-3}
10^{-6}	113	0	1257	5.54×10^{-5}
10^{-8}	299	0	3326	2.02×10^{-6}

Table 4. Numerical results for Problem 1 solved by the method of order 4 with $h_0 = 10^{-5}$.

tol	*ns*	*nrs*	*nfe*	*ge*
10^{-2}	172	0	5088	8.02×10^{-3}
10^{-4}	204	0	3067	7.35×10^{-4}
10^{-6}	188	0	2418	3.58×10^{-5}
10^{-8}	373	0	4809	1.35×10^{-6}

Table 5. Numerical results for Problem 2 solved by the method of order 2 with $h_0 = 10^{-5}$.

tol	*ns*	*nrs*	*nfe*	*ge*
10^{-2}	23	1	247	9.24×10^{-2}
10^{-4}	67	0	585	5.17×10^{-3}
10^{-6}	535	2	4803	4.55×10^{-4}
10^{-8}	1656	3	14922	4.40×10^{-6}

Table 6. Numerical results for Problem 2 solved by the method of order 3 with $h_0 = 10^{-5}$.

tol	*ns*	*nrs*	*nfe*	*ge*
10^{-2}	54	4	653	1.16×10^{-1}
10^{-4}	94	1	1014	5.07×10^{-3}
10^{-6}	271	8	2580	2.14×10^{-4}
10^{-8}	1109	9	10053	8.80×10^{-6}

Table 7. Numerical results for Problem 2 solved by the method of order 4 with $h_0 = 10^{-5}$.

tol	*ns*	*nrs*	*nfe*	*ge*
10^{-2}	150	0	1332	1.09×10^{-1}
10^{-4}	166	2	1497	3.12×10^{-3}
10^{-6}	286	9	2690	1.15×10^{-4}
10^{-8}	760	35	6367	1.54×10^{-6}

5. Conclusion

We considered a modified version of BDFs with two off–step points in the form of GLMs. This representation made it easier to improve the stability properties of the methods such that the constructed methods are A–stable up to order 4 and $A(\alpha)$-stable up to order 10 with larger angles. To apply the methods with the variable stepsize strategy, we converted the methods to the Nordsieck form. We showed that the constructed methods are practical by their implementation in variable stepsize environment on two stiff initial value problems.

References

[1] J.C. Butcher, *On the convergence of numerical solutions to ordinary differential equations*, Math. Comp., *20* (1966), 1–10.

[2] J.C. Butcher, *General linear methods for stiff differential equations*, BIT, *41* (2001), 240–264.

[3] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Wiley, New York, 2008.

[4] J.R. Cash, *The integration of stiff initial value problems in ODEs using modified extended backward differentiation formula*, Comput. Math. Appl., *9* (1983), 645–657.
[5] J.R. Cash, *Second derivative extended backeward differentation formula for the numerical integration of stiff systems*, SIAM J. Numer. Anal., *18* (1981), 21–36.
[6] J.R. Cash, *On the integration of stiff systems of ODEs using extended backward differentiation formula*, Numer. Math., *34* (1980), 235–246.
[7] G. Dahlquist, *A special stability problem for linear multistep methods*, BIT, *3* (1963), 27–43.
[8] M. Ebadi and M.Y. Gokhale, *Hybrid BDF methods for the numerical solutions of ordinary differential equations*, Numer. Algor., *55* (2010), 1–17.
[9] A.K. Ezzeddine and G. Hojjati, *Hybrid Extended Backward Differentiation Formulas for Stiff Systems*, Int. J. Nonlinear Science, *12* (2011), 196–204.
[10] J.E. Frank and P.J. Van Der Houwen, *Parallel iteration of the extended backward differentiation formulas*, IMA Journal of Numerical Analysis, *21* (2001), 367–385.
[11] C. Fredebeul, *A–BDF: A generalization of the backward differentiation formulae*, SIAM J. Numer. Anal., *12* (1998), 1917–1938.
[12] W.C. Gear, *Simultaneous Numerical Solution of Differential Algebraic Equation*, IEEE Transaction on Circuit Theory, *18* (1971), 89–95.
[13] W.C. Gear, *Hybrid methods for initial value problems in ordinary differential equations*, SIAM J. Numer. Anal., *2* (1965), 69–86.
[14] E.Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, Berlin, 2000.
[15] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential–Algebraic Problems*, Second Revised Edition. Springer Verlag, Berlin, Heidelberg, New York, 1996.
[16] G. Hojjati, M. Rahimi and S.M. Hosseini, *New second derivative multistep methods for stiff systems*, Appl. Math. Model., *30* (2006), 466–476.
[17] G. Hojjati, M. Rahimi and S.M. Hosseini, *A–EBDF: an adaptive method for numerical solution of stiff systems of ODEs*, Math. Comput. Simul., *66* (2004), 33–41.
[18] G. Hojjati and S. M. Hosseini, *Matrix–free MEBDF method for numerical solution of systems of ODEs*, Math. Comput. Model., *29* (1999), 67–77.
[19] S.J.Y. Huang, *Implementation of general linear methods for stiff ordinary differential equations*, PhD thesis, Department of Mathematics, Auckland University, 2005.
[20] Z. Jackiewicz, *General linear methods for ordinary differential equations*, Wiley, New Jersey, 2009.
[21] J.D. Lambert, *Computational methods in ordinary differential equations*, Wiley Eastern Limited, 1984.