

Solving Stiff Ordinary Differential Equations Using Extended Block Backward Differentiation Formulae

Siti Ainor Mohd Yatim, Zarina Bibi Ibrahim, Khairil Iskandar Othman and Mohamed Suleiman

Abstract A comprehensive research on the existing Block Backward Differentiation Formulae (BBDF) was done. Based on the suitability in solving stiff ordinary differential equations (ODEs), BBDF of order 3 up 5 is collected using simplified strategy in controlling the step size and order of the method. Thus, Extended Block Backward Differentiation Formulae (EBBDF) is derived with the intention of optimizing the performance in terms of precision and computation time. The accuracy of the method are investigated using linear and non linear stiff initial value problems and its performance is compared with MATLAB's suite of ODEs solvers namely ode15s and ode23s.

Keywords BDF methods · Block methods · Initial value problem · Numerical analysis · Ordinary differential equations · Stiff ODEs

S. A. M. Yatim (✉)
School of Distance Education, Universiti Sains Malaysia USM,
11800 Penang, Malaysia
e-mail: ainormy@yahoo.com

Z. B. Ibrahim · M. Suleiman
Department of Mathematics, Universiti Putra Malaysia UPM,
43400 Serdang, Selangor, Malaysia
e-mail: zarinabb@science.upm.edu.my

M. Suleiman
e-mail: mohameds@science.upm.edu.my

K. I. Othman
Department of Mathematics, Universiti Teknologi MARA,
40450 Shah Alam, Selangor, Malaysia
e-mail: khairil@tmsk.uitm.edu.my

1 Introduction

Applied problems arise from chemical, biological and physical phenomenon, particularly in many field of science and engineering [1–4], have inspired numbers of researches to develop effective and very accurate methods to solve stiff initial value problems (IVP) [5]. Many renowned method for solving stiff problems are based on backward differentiation formula (BDF) which also known as Gear’s method and was first introduced in [6]. The method was then developed to improve the results in terms of accuracy and computation time. The evidence is in the advancements of many other codes to meet the same objective of finding the most accurate approximation for IVPs. These codes include EPISODE, VODE, LSODE, MEBDF etc. [5–7].

Many in recent chapters have tried to describe and compare some of the best codes by taking into accounts their accuracy, rate of convergence, and computation time [8]. Consequently, some excellent codes which are both efficient and reliable for solving these particular classes of problems are made available. With the same objective, this chapter consider the numerical solution of the first order initial value problem,

$$y' = f(x, y) \quad (1)$$

with given initial values $y(a) = y_0$ in the given interval $x \in [a, b]$.

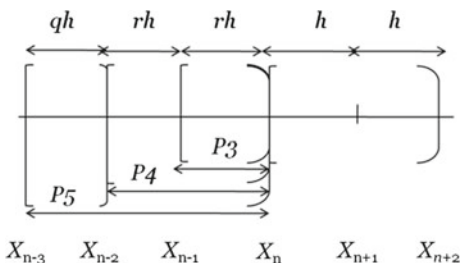
From the study on BBDF in [9], the competency of computing concurrent solution values at different points were revealed. The idea in [10, 11] was then extended by increasing the order of the method up to order 5. In this chapter, we extend and review the study of Extended Block Backward Differentiation Formulae (EBBDF) in [12]. Additionally, this chapter investigates the consistency and convergence of the method.

2 Derivation of Extended Block Backward Differentiation Formulae Method

2.1 Construction of EBBDF

Two values of y_{n+1} and y_{n+2} were computed simultaneously in block by using earlier blocks with each block containing a maximum of two points (Fig. 1). The orders of the method ($P3$, $P4$ and $P5$) are distinguished by the number of backvalues contained in total blocks. The ratio distance between current (x_n) and previous step (x_{n-1}) is represented as r and q in Fig. 1. In this chapter, the step size is given selection to decrease to half of the previous steps, or increase up to a factor of 1.9. For simplicity, q is assigned as 1, 2 and 10/19 for the case of constant, halving and increasing the step size respectively. The zero stability is achieved for each of these cases and explained in the next section.

Fig. 1 EBBDF method of order ($P3 - P5$)



We find approximating polynomials $P_k(x)$, by means of a k -degree polynomial interpolating the values of y at given points are $(x_{n-3}, y_{n-3}), (x_{n-2}, y_{n-2}), (x_{n-1}, y_{n-1}) \dots, (x_{n+2}, y_{n+2})$.

$$P_k = \sum_{j=0}^k y(x_{n+1-j}) \cdot L_{k,j}(x) \tag{2}$$

where

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

The interpolating polynomial of the function $y(x)$ using Lagrange polynomial in (2) gives the following corrector for the first point y_{n+1}^P , and second point y_{n+2}^P . The resulting Lagrange polynomial for each order was given as follows:

For EBBDF of order $P3$ ($P = 3$)

$$\begin{aligned} P(x) &= P(x_{n+1} + sh) \\ &= \frac{(r + 1 + s)(s + 1)(s)}{2r + 4} y_{n+2} + \frac{(r + 1 + s)(s + 1)(s - 1)}{-1 - r} y_{n+1} \\ &\quad + \frac{(r + 1 + s)(s - 1)(s)}{2r} y_n + \frac{(1 + s)(s - 1)(s)}{-r(-1 - r)(-r - 2)} y_{n-1} \end{aligned} \tag{3}$$

For EBBDF of order $P4$ ($P = 4$)

$$\begin{aligned} P(x) &= P(x_{n+1} + sh) \\ &= \frac{(2r + 1 + s)(r + 1 + s)(1 + s)(s)}{2(2r + 2)(r + 2)} y_{n+2} \\ &\quad + \frac{(2r + 1 + s)(r + 1 + s)(1 + s)(s - 1)}{-(2r + 1)(r + 1)} y_{n+1} \end{aligned}$$

$$\begin{aligned}
& + \frac{(2r+1+s)(r+1+s)(s)(s-1)}{4r^2} y_n \\
& + \frac{(2r+1+s)(1+s)(s)(s-1)}{-r^2(-r-1)(-r-2)} y_{n-1} \\
& + \frac{(r+1+s)(1+s)(s)(s-1)}{2r^2(-2r-1)(-2r-2)} y_{n-2}
\end{aligned} \tag{4}$$

For EBBDF of order $P5$ ($P = 5$)

$$\begin{aligned}
P(x) &= P(x_{n+1} + sh) \\
&= \frac{(q+2r+1+s)(2r+1+s)(r+1+s)(1+s)s}{2(q+2r+2)(2r+2)(r+2)} y_{n+2} \\
&+ \frac{(q+2r+1+s)(2r+1+s)(r+1+s)(1+s)(s-1)}{-(q+2r+1)(2r+1)(r+1)} y_{n+1} \\
&+ \frac{(q+2r+1+s)(2r+1+s)(r+1+s)s(s-1)}{4(q+2r)r^2} y_n \\
&+ \frac{(q+2r+1+s)(2r+1+s)(1+s)s(s-1)}{-r^2(q+r)(-r-1)(-r-2)} y_{n-1} \\
&+ \frac{(q+2r+1+s)(r+1+s)(1+s)s(s-1)}{2qr^2(-2r-1)(-2r-2)} y_{n-2} \\
&+ \frac{(2r+1+s)(r+1+s)(1+s)s(s-1)}{-q(-q-r)(-q-2r)(-q-2r-1)(-q-2r-2)} y_{n-3}
\end{aligned} \tag{5}$$

By substituting $s = 0$ and $s = 1$ gives the corrector for the first and second point respectively. Therefore by letting $r = 1, q = 1, r = 2, q = 2$ and $r = 1, q = 10/19$ we produced the following equations for the first and second point of EBBDF.

EBBDF of order $P3$ ($p = 3$)

When $r = 1, q = 1$

$$\begin{aligned}
y_{n+1} &= 2hf_{n+1} - \frac{2}{3}y_{n+2} + 2y_n - \frac{1}{3}y_{n-1} \\
y_{n+2} &= \frac{6}{11}hf_{n+2} + \frac{18}{11}y_{n+1} - \frac{9}{11}y_n + \frac{2}{11}y_{n-1}
\end{aligned}$$

When $r = 2, q = 2$

$$\begin{aligned}
y_{n+1} &= 3hf_{n+1} + \frac{9}{8}y_{n+2} + \frac{9}{4}y_n - \frac{1}{8}y_{n-1} \\
y_{n+2} &= \frac{4}{7}hf_{n+2} + \frac{32}{21}y_{n+1} - \frac{4}{7}y_n + \frac{1}{21}y_{n-1}
\end{aligned}$$

When $r = 1, q = 10/19$

$$y_{n+1} = \frac{29}{19}hf_{n+1} - \frac{841}{1824}y_{n+2} + \frac{841}{380}y_n - \frac{361}{480}y_{n-1}$$

$$y_{n+2} = \frac{48}{91}hf_{n+2} + \frac{4608}{2639}y_{n+1} - \frac{576}{455}y_n + \frac{6859}{13195}y_{n-1}$$

EBBDF of order $P4$ ($P = 4$)

When $r = 1, q = 1$

$$y_{n+1} = \frac{6}{5}hf_{n+1} - \frac{3}{10}y_{n+2} + \frac{9}{5}y_n - \frac{3}{5}y_{n-1} + \frac{1}{10}y_{n-2}$$

$$y_{n+2} = \frac{12}{25}hf_{n+2} + \frac{48}{25}y_{n+1} - \frac{36}{25}y_n + \frac{16}{25}y_{n-1} - \frac{3}{25}y_{n-2}$$

When $r = 2, q = 2$

$$y_{n+1} = \frac{15}{8}hf_{n+1} - \frac{75}{128}y_{n+2} + \frac{225}{128}y_n - \frac{25}{128}y_{n-1} + \frac{3}{128}y_{n-2}$$

$$y_{n+2} = \frac{12}{23}hf_{n+2} + \frac{192}{115}y_{n+1} - \frac{18}{23}y_n + \frac{3}{23}y_{n-1} - \frac{2}{115}y_{n-2}$$

When $r = 1, q = 10/19$

$$y_{n+1} = \frac{1131}{1292}hf_{n+1} - \frac{14703}{82688}y_{n+2} + \frac{1279161}{516800}y_n$$

$$- \frac{183027}{108800}y_{n-1} + \frac{10469}{27200}y_{n-2}$$

$$y_{n+2} = \frac{1392}{3095}hf_{n+2} + \frac{89088}{40235}y_{n+1} - \frac{242208}{77375}y_n$$

$$+ \frac{198911}{77375}y_{n-1} - \frac{658464}{1005875}y_{n-2}$$

EBBDF of order $P5$ ($P = 5$)

When $r = 1, q = 1$

$$y_{n+1} = \frac{12}{13}hf_{n+1} - \frac{12}{65}y_{n+2} + \frac{24}{13}y_n - \frac{12}{13}y_{n-1}$$

$$+ \frac{4}{13}y_{n-2} - \frac{3}{65}y_{n-3}$$

$$y_{n+2} = \frac{60}{137}hf_{n+2} + \frac{300}{137}y_{n+1} - \frac{300}{137}y_n + \frac{200}{137}y_{n-1}$$

$$- \frac{75}{137}y_{n-2} + \frac{12}{137}y_{n-3}$$

When $r = 2, q = 2$

$$\begin{aligned}
y_{n+1} &= \frac{105}{71}hf_{n+1} - \frac{3675}{9088}y_{n+2} + \frac{3675}{2272}y_n - \frac{1225}{4544}y_{n-1} \\
&\quad + \frac{147}{2272}y_{n-2} - \frac{75}{9088}y_{n-3} \\
y_{n+2} &= \frac{24}{49}hf_{n+2} + \frac{3072}{1715}y_{n+1} - \frac{48}{49}y_n + \frac{12}{49}y_{n-1} \\
&\quad - \frac{16}{245}y_{n-2} + \frac{3}{343}y_{n-3}
\end{aligned}$$

When $r = 1, q = 10/19$

$$\begin{aligned}
y_{n+1} &= \frac{402}{449}hf_{n+1} - \frac{13467}{77228}y_{n+2} + \frac{13467}{7184}y_n - \frac{13467}{13021}y_{n-1} \\
&\quad + \frac{4489}{8980}y_{n-2} - \frac{7428297}{44792240}y_{n-3} \\
y_{n+2} &= \frac{516}{1189}hf_{n+2} + \frac{177504}{79663}y_{n+1} - \frac{5547}{2378}y_n + \frac{59168}{34481}y_{n-1} \\
&\quad - \frac{5547}{5945}y_{n-2} + \frac{7428297}{23102270}y_{n-3}
\end{aligned}$$

As similar to analysis for order of Linear Multistep Method (LMM) given in [13], we use the following definition to determine the order of EBBDF method.

Definition 1 The LMM [13] and the associated difference operator L defined by

$$L[z(x); h] = \sum_{k=0}^j [\alpha_k z(x + kh) - h\beta_k z'(x + kh)] \quad (6)$$

are said to be of order p if $c_0 = c_1 = \dots = c_p = 0, C_{p+1} \neq 0$. The general form for the constant C_q is defined as

$$C_q = \sum_{k=0}^j \left[k^q \alpha_k - \frac{1}{(q-1)!} k^{q-1} \beta_k \right], \quad q = 2, 3, \dots, p+1 \quad (7)$$

Consequently, BBDF method can be represented in standard form by an equation $\sum_{j=0}^k A_j y_{n+j} = h \sum_{j=0}^k B_j f_{n+j}$ where A_j and B_j are r by r matrices with elements $a_{l,m}$ and $b_{l,m}$ for $l, m = 1, 2, \dots, r$. Since EBBDF for variable order (P) is a block method, we extend the Definition 1 in the form of

$$L[z(x); h] = \sum_{k=0}^j [A_k z(x + kh) - hB_k z'(x + kh)] \quad (8)$$

And the general form for the constant C_q is defined as

$$C_q = \sum_{k=0}^j \left[k^q A_k - \frac{1}{(q-1)!} k^{q-1} B_k \right] \quad q = 2, 3, \dots, p+1 \quad (9)$$

A_k is equal to the coefficients of y_k where $k = n = (p-2), \dots, n+1, n+2$ and $P = 3, 4, 5$.

Throughout this section, we illustrate the effect of Newton-type scheme which in general form of

$$y_{n+1,n+2}^{(i+1)} - y_{n+1,n+2}^{(i)} = - \left[(i-A) - hB \frac{\partial F}{\partial y} y_{n+1,n+2}^{(i)} \right]^{-1} \left[(I-A) y_{n+1,n+2}^i - hBF \left(y_{n+1,n+2}^{(i)} \right) - \xi \right] \quad (10)$$

The general form of EBBDF method is

$$\begin{cases} y_{n+1} = \alpha_1 h f_{n+1} + \theta_1 y_{n+2} + \psi_1 \\ y_{n+2} = \alpha_1 h f_{n+2} + \theta_1 y_{n+1} + \psi_2 \end{cases} \quad (11)$$

With ψ_1 and ψ_2 are the back values. By setting,

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, y_{n+1,n+2} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix}, B = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}, F_{n+1,n+2} = \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}, \text{ and}$$

$$\xi_{n+1,n+2} = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}$$

Equation (11) in matrix-vector form is equivalent to

$$(I-A)y_{n+1,n+2} = hBF_{n+1,n+2} + \xi_{n+1,n+2} \quad (12)$$

Equation (12) is simplified as

$$\hat{f}_{n+1,n+2} = (I-A)y_{n+1,n+2} - hBF_{n+1,n+2} - \xi_{n+1,n+2} = 0 \quad (13)$$

Newton iteration is performed to the system $\hat{f}_{n+1,n+2} = 0$, by taking the analogous form of (10) where $J_{n+1,n+2} = \left(\frac{\partial F}{\partial Y} \right) \left(Y_{n+1,n+2}^{(i)} \right)$, is the Jacobian matrix of F with respect to Y . Equation (10) is separated to three different matrices denoted as

$$E_{1,2}^{(i+1)} = y_{n+1,n+2}^{(i+1)} - y_{n+1,n+2}^{(i)} \quad (14)$$

$$\hat{A} = (I-A) - hB \frac{\partial F}{\partial Y} \left(y_{n+1,n+2}^{(i)} \right) \quad (15)$$

$$\hat{B} = (I - A)y_{n+1,n+2}^{(i)} - hBF \left(y_{n+1,n+2}^{(i)} \right) - \xi_{n+1,n+2} \quad (16)$$

Two-stage Newton iteration works to find the approximating solution to (1) with two simplified strategies based on evaluating the Jacobian ($J_{n+1,n+2}$) and LU factorization of \hat{A} [13].

2.2 Consistency and Convergence of EBBDF Methods

We first consider the general form of a block linear multistep method (LMM):

$$\sum_{j=0}^k A_j y_{n+j} = h \sum_{j=0}^k B_j F_{n+j} \quad (17)$$

where A_j and B_j are r by r matrices with elements α_{im}, β_{im} for $i.m = 0, 1, \dots, r$.

Equation (17) is applied for EBBDF method

$$\sum_{j=0}^k \alpha_{ij} y_{n+2-j} = h \sum_{j=0}^k \beta_j f_{n+2-j} \quad (18)$$

The expression (18) is expanded to give the following system of equation.

$$\begin{aligned} & \begin{bmatrix} \alpha_{1,0} & \alpha_{1,1} \\ \alpha_{2,0} & \alpha_{2,1} \end{bmatrix} \begin{bmatrix} y_{n+2} \\ y_{n+1} \end{bmatrix} + \dots + \begin{bmatrix} \alpha_{1,k-1} & \alpha_{1,k} \\ \alpha_{2,k-1} & \alpha_{2,k} \end{bmatrix} \begin{bmatrix} y_{n+3-k} \\ y_{n+2-k} \end{bmatrix} \\ & = h \left[\begin{bmatrix} \beta_{1,4} & \beta_{1,5} \\ \beta_{2,4} & \beta_{2,5} \end{bmatrix} \begin{bmatrix} f_{n+2} \\ f_{n+1} \end{bmatrix} + \dots + \begin{bmatrix} \beta_{1,k-1} & \beta_{1,k} \\ \beta_{2,k-1} & \beta_{2,k} \end{bmatrix} \begin{bmatrix} f_{n+3-k} \\ f_{n+2-k} \end{bmatrix} \right] \end{aligned} \quad (19)$$

With

$$A_0 = \begin{bmatrix} \alpha_{1,0} \\ \alpha_{2,0} \end{bmatrix}, \dots, A_k = \begin{bmatrix} \alpha_{1,k} \\ \alpha_{2,k} \end{bmatrix}, \text{ and } B_1 = \begin{bmatrix} \beta_{1,1} \\ \beta_{2,1} \end{bmatrix}, \dots, B_k = \begin{bmatrix} \beta_{1,k} \\ \beta_{2,k} \end{bmatrix}$$

Adopting the order procedure used in the single case for the block method, we associate with a linear difference operator $L[y(x_n), h]$, given as

$$L[y(x_n), h] = \sum_{j=0}^k [\alpha_j y(x_n + jh) - h\beta_j f(x_n + jh, y(x_n + jh))] \quad (20)$$

The use of appropriate Taylor expansions about a suitable x , reduces (20) to the form

$$L[y(x_n), h] = c_0 y(x_n) + c_1 h y'(x_n) + \dots + c_r h^r y^r(x_n) \quad (21)$$

where

$$c_0 = \sum_{j=0}^k A_j, c_1 = \sum_{j=0}^k j A_j - \sum_{j=0}^k B_j \text{ and,}$$

$$c_r = \frac{1}{r!} \sum_{j=0}^k j^r A_j - \frac{1}{(r-1)!} \sum_{j=0}^k j^{r-1} B_j$$

The EBBDF method is said to be of order p if $c_0 = c_1 = \dots = c_p = 0, c_{p+1} \neq 0$ and the local truncation error is

$$t_{n+k} = c_{p+1} h^{p+1} y^{p+1}(x_n) + O(h^{p+2}) \quad (22)$$

2.3 Stability Regions of the EBBDF Method

Definition 2 A method is said to be absolute stable in a region R for a given $h\lambda$ if for that $h\lambda$, all the roots r_s of the stability polynomial $\pi(r, h\lambda) = \rho(r) - h\lambda\sigma(r) = 0$, satisfy $|r_s| < 1, s = 1, 2, \dots, k$.

Definition 3 The LMM is said to be zero-stable if no root of the first characteristic polynomial $\rho(r)$ has modulus greater than one, and if every root with unit modulus is simple.

By applying test equation $y' = \lambda y$ to (1.1) we obtain,

$$\sum_{j=0}^k \alpha_{ij} y_{n+2-j} = h \sum_{j=0}^k \beta_j \lambda_{n+2-j} \quad (23)$$

The Eq. (23) is equivalent to

$$\sum_{j=0}^r A_j Y_j = 0 \quad (24)$$

where $A_j = [A_0, A_1, \dots, A_r], Y_j = [Y_0, Y_1, \dots, Y_r]$ and,

$$A_j = \begin{bmatrix} \alpha_{1,2j} & \alpha_{1,(2j+1)} - \beta_{1,(2j+1)} h\lambda \\ \alpha_{2,(2j)} - \beta_{2,(2j)} h\lambda & \alpha_{2,(2j+1)} \end{bmatrix}, Y_j = \begin{bmatrix} y_{n+2-2j} \\ y_{n+1-2j} \end{bmatrix}$$

The stability polynomial, $R(t, \hat{h})$ associated with the method of (18) is given by

$$\det \left(\sum_{j=0}^r A_j t^j \right) \quad (25)$$

while the absolute stability region of this method in the $h\lambda$ plane is determined by solving $\det \left(\sum_{j=0}^r A_j t^j \right) = 0$. The stability region was given by the set of points

determined by the boundary $t = e^{i\theta}$, $0 \leq \theta \leq 2\pi$. The stability region is obtained by finding the region for which $|t| < 1$.

2.4 Order and Stepsize Selection

The importance of choosing the step size is to achieve reduction in computation time and number of iterations. Meanwhile changing the order of the method is designed for finding the best approximation. Strategies proposed in [14] are applied in this study for choosing the step size and order. The strategy is to estimate the maximum step size for the following step. Methods of order $P - 1$, P , $P + 1$ are selected depending on the occurrence of every successful step. Consequently, the new step size h_{new} is obtained from which order produces the maximum step size.

The user initially will have to provide an error tolerance limit, TOL on any given step and obtain the local truncation error (LTE) for each iteration. The LTE is obtained from

$$LTE_k = y_{n+2}^{(P+1)} - y_{n+2}^{(P)}, \quad P = 3, 4, 5$$

where $y_{n+2}^{(P+1)}$ is the $(P + 1)$ th order method and $y_{n+2}^{(P)}$ is the k th order method. By finding the LTEs, the maximum step size is defined as

$$h_{P-1} = h_{old} \times \left(\frac{TOL}{LTE_{P-1}} \right)^{\frac{1}{P}}, \quad h_P = h_{old} \times \left(\frac{TOL}{LTE_P} \right)^{\frac{1}{P+1}}, \quad h_{P+1} = h_{old} \times \left(\frac{TOL}{LTE_{P+1}} \right)^{\frac{1}{P+2}}$$

where h_{old} is the stepsize from previous block and h_{max} is obtained from the maximum stepsize given in above equations.

The successful step is dependent on the condition $LTE < TOL$. If this condition fails, the values of y_{n+1}, y_{n+2} are rejected, and the current step is reiterated with step size selection ($q = 2$). On the contrary, the step size increment for each successful step is defined as $h_{new} = c \times h_{max}$ and if $h_{new} > 1.9 \times h_{old}$ then $h_{new} = 1.9 \times h_{old}$. Where c is the safety factor, p is the order of the method while and is the step size from previous and current block respectively. In this chapter, c is set to be 0.8 so as to make sure the rejected step is being reduced.

3 Numerical Results

We carry out numerical experiments to compare the performance of EBBDF method with stiff ODE solvers in **MATLAB** mentioned earlier. This chapter considers the comparison of four different factors namely number of steps taken, average error, maximum error and computation time. These test problems are performed under different conditions of error tolerances—(a) 10^{-2} , (b) 10^{-4} and, (c) 10^{-6}

Table 1 Numerical results for problem 1

	VSVO	ODE15s	ODE23s
TOL	1.000e-002		
Total steps	29	41	37
Maximum error	6.4925e-003	0.0084	0.0045
Average error	8.8396e-004	9.1188e-004	0.0011
Time	0.0104	0.1094	0.0781
TOL	1.000e-004		
Total steps	74	94	182
Maximum error	1.9531e-006	1.6634e-004	2.5500e-004
Average error	4.6275e-007	1.9159e-005	4.7081e-005
Time	0.0107	0.1250	0.1563
TOL	1.000e-006		
Total steps	279	163	1194
Maximum error	9.8112e-007	3.0953e-006	1.0911e-005
Average error	4.8070e-007	1.3149e-006	1.1596e-006
Time	0.0125	0.2813	0.6250

The test problems and solution are listed below

Problem 1

$$y' = -1000y + 3000 - 2000e^{(-x)}y(0) = 0 \quad 0 \leq x \leq 20$$

With solution: $y(x) = 3 - 0.998e^{(-1000x)} - 2.002e^{(-x)}$

Problem 2

Table 2 Numerical results for problem 2

	VSVO	ODE15s	ODE23s
TOL	1.000e-002		
Total steps	35	45	137
Maximum error	3.0045e-004	0.0146	0.0030
Average error	4.6584e-005	0.0069	0.0014
Time	0.0111	0.0156	0.2031
TOL	1.000e-004		
Total steps	84	99	1211
Maximum error	1.1002e-005	2.7591e-004	6.3075e-005
Average error	2.5775e-006	7.6548e-005	2.4915e-005
Time	0.0123	0.0469	0.3281
TOL	1.000e-006		
Total steps	380	186	8289
Maximum error	8.9627e-008	6.1936e-006	1.5667e-006
Average error	2.4244e-008	2.1681e-006	5.3155e-007
Time	0.0199	0.0781	2.3750

$$y_1' = -2y_1 + y_2 + 2 \sin(x) \quad y_1(0) = 2 \quad 0 \leq x \leq 10$$

$$y_2' = 998y_1 - 999y_2 + 999(\cos(x) - \sin(x)) \quad y_2(0) = 3$$

With solution:

$$y_1(x) = 2e^{(-x)} + \sin(x)$$

$$y_2(x) = 2e^{(-x)} + \cos(x)$$

From Tables 1 and 2, among the three methods tested, our method, EBBDF method requires the shortest execution time, smallest maximum error and average error for each given tolerance level. Moreover, EBBDF outperformed ode15s and ode23s in term of total steps when the tolerance level is less than 10^{-4} .

4 Conclusion and Future Work

Extended Block Backward Differentiation Formulae (EBBDF) was derived in this chapter. From the numerical experiments, the comparisons between EBBDF and solvers in **MATLAB**, show improvements in term of computation time, average error as well as maximum error. Therefore, we conclude that EBBDF can serve as an alternative solver for solving stiff ODEs. Subsequently, the method can be extended to solve higher order ODEs in future. One can also apply parallel computing to improve the computation time.

References

1. Sack-Davis R (1980) Fixed leading coefficient implementation of SD-formulas for stiff ODEs. *ACM Trans Math Softw* 6(4):540–562
2. Yatim SAM, Ibrahim ZB, Othman KI, Suleiman MB (2011) Quantitative comparison of numerical method for solving stiff ordinary differential equations. *Math Prob Eng* 2011, ID 193961
3. Mahmood AS, Casasus L, Al-Hayani W (2005) The decomposition method for stiff systems of ordinary differential equations. *Appl Math Comput* 167(2):964–975
4. Ibanez J, Hernandez V, Arias E, Ruiz PA (2009) Solving initial value problems for ordinary differential equations by two approaches: BDF and piecewise-linearized methods. *Comput Phys Commun* 180(5):712–723
5. Enright WH, Hull TE, Lindberg B (1975) Comparing numerical methods for stiff systems of O.D.Es. *BIT* 15(1):10–48
6. Byrne GD, Hindmarsh AC, Jackson KR, Brown HG (1977) A comparison of two ode codes: gear and episode. *Comput Chem Eng* 1(2):133–147
7. Cash JR, Considine S (1992) An MEBDF code for stiff initial value problems. *ACM Trans Math Softw* 18(2):142–155
8. Abelman S, Patidar KC (2008) Comparison of some recent numerical methods for initial-value problems for stiff ordinary differential equations. *Comput Math Appl* 55(4):733–744

9. Ibrahim ZB, Suleiman MB, Othman KI (2008) Fixed coefficients block backward differentiation formulas for the numerical solution of stiff ordinary differential equations. *Eur J Sci Res* 21(3):508–520
10. Ibrahim ZB, Othman KI, Suleiman MB (2007) Variable step size block backward differentiation formula for solving stiff odes. In: *Proceedings of the world congress on engineering 2007, WCE 2007*, London, UK, pp 785–789, 2–4 July 2007
11. Yatim SAM, Ibrahim ZB, Othman KI, Suleiman MB (2010) Fifth order variable step block backward differentiation formulae for solving stiff ODEs. *Proc World Acad Sci Eng Tech* 62:998–1000
12. Yatim SAM, Ibrahim ZB, Othman KI, Suleiman MB (2012) Numerical solution of extended block backward differentiation formulae for solving stiff ODEs, *Lecture notes in engineering and computer science: proceedings of the world congress on engineering 2012, WCE 2012*, London, UK, pp 109–113, 4–6 July 2012
13. Lambert JD (1973) *Computational methods ordinary differential equation*. Wiley, New York
14. Hall G, Watt JM (1976) *Modern numerical methods for ordinary differential equations*. Clarendon Press, Oxford