Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 989381, 11 pages
http://dx.doi.org/10.1155/2013/989381

Research Article

A Numerical Algorithm for Solving Stiff Ordinary Differential Equations

S. A. M. Yatim,[1] Z. B. Ibrahim,[2] K. I. Othman,[3] and M. B. Suleiman[2]

[1] *School of Distance Education, Universiti Sains Malaysia, 11800 Penang, Malaysia*
[2] *Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Selangor, 43400 Serdang, Malaysia*
[3] *Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Selangor, 40450 Shah Alam, Malaysia*

Correspondence should be addressed to Z. B. Ibrahim; zarinabb@math.upm.edu.my

Received 6 August 2012; Revised 24 October 2012; Accepted 7 November 2012

Academic Editor: J. Rodellar

An advanced method using block backward differentiation formula (BBDF) is introduced with efficient strategy in choosing the step size and order of the method. Variable step and variable order block backward differentiation formula (VSVO-BBDF) approach is applied throughout the numerical computation. The stability regions of the VSVO-BBDF method are investigated and presented in distinct graphs. The improved performances in terms of accuracy and computation time are presented in the numerical results with different sets of test problems. Comparisons are made between the proposed method and MATLAB's suite of ordinary differential equations (ODEs) solvers, namely, ode15s and ode23s.

1. Introduction

Many studies on solving the equations of stiff ordinary differential equations (ODEs) have been done by researchers or mathematicians specifically. With the numbers of numerical methods that currently exist in the literature, extensive research has been done to unveil the comparison between their rate of convergence, number of computations, accuracy, and capability to solve certain type of test problems [1]. The well-known numerical methods that are used widely are from the class of BDFs or commonly understood as Gear's Method [2]. However, many other methods that have evolved to this date are for solving stiff ODEs which arise in many fields of the applied sciences [3–6]. The problems considered in this paper are for the numerical solution of the initial value problem,

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

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

The existing numerical methods have often been compared to one another to find the best approximation and the best method. MATLAB ODE suite is one of the most preferred solvers to be used for comparison purposes [7, 8]. Stiff ODE solvers that are available in MATLAB ODE suite are ode15s and ode23s which are based on the numerical differentiation formulas [8, 9] and the modified Rosenbrock formula of order 2, respectively [8].

An improved method has been identified in [10], which was expanded from the method incorporated with BDF proposed by Gear. Since then, the study on producing block approximations $y_{n+1}, y_{n+2}, y_{n+3}, \ldots, y_{n+k}$ also known as Block Backward Differentiation Formulae (BBDF) [11] has attracted much attention. BBDF method by using variable step size in [12] demonstrates the competency of computing concurrent solution values at different points. Consequently, study in [13, 14] is an extension of a previous study in a way that the accuracy is improved by increasing the order of the method up to order 5. Thus, these studies lead to enhancing the existing method to become Variable Order Variable Step Block Backward Differentiation Formula of order 3 until 5 (VSVO-BBDF).

In Section 2, we introduce the general formulation of VSVO-BBDF of order 3 to 5. Order conditions are listed in Section 3 while in Section 4, we consider the implementation of VSVO-BBDF. The analysis of stability regions is illustrated

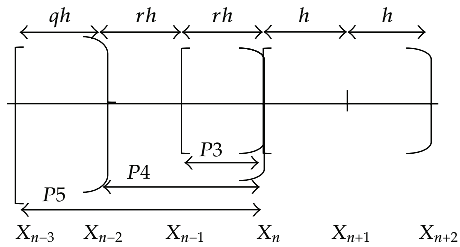


FIGURE 1: VSVO-BBDF method of order ($P3$–$P5$).

in Section 5. This is followed by the strategy in choosing the step size and order of VSVO-BBDF in Section 6. Numerical results are presented in Section 7. The appendix describes the algorithm applied for VSVO-BBDF method.

2. VSVO-BBDF Method Formulation

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 (Figure 1). The orders of the method ($P3$, $P4$, and $P5$) are distinguished by the number of back values contained in total blocks. The ratio distance between current (x_n) and previous step (x_{n-1}) is represented as r and q in Figure 1. In this paper, 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.

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

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

where

$$L_{k,j}(x) = \prod_{\substack{i=0 \\ i \neq j}}^{k} \frac{\left(x - x_{n+1-i}\right)}{\left(x_{n+1-j} - x_{n+1-i}\right)} \quad \text{for each } j = 0, 1, \ldots, k. \tag{3}$$

Predictors for the first point y_{n+1}^{p} and second point y_{n+2}^{p} were computed using back values as the interpolating points. The resulting Lagrange polynomial for each order was given as follows.

For VSVO-BBDF of order $P3$ ($P = 3$)

$$P(x) = P\left(x_{n+1} + sh\right) = \frac{(r+1+s)}{r} y_n + \frac{(1+s)}{-r} y_{n-1}. \tag{4}$$

For VSVO-BBDF of order $P4$ ($P = 4$)

$$\begin{aligned} P(x) &= P\left(x_{n+1} + sh\right) \\ &= \frac{(2r+1+s)(r+1+s)}{2r^2} y_n + \frac{(2r+1+s)(1+s)}{-r^2} y_{n-1} \\ &\quad + \frac{(r+1+s)(1+s)}{2r^2} y_{n-2}. \end{aligned} \tag{5}$$

For VSVO-BBDF of order $P5$ ($P = 5$)

$$\begin{aligned} P(x) &= P\left(x_{n+1} + sh\right) \\ &= \frac{(q+2r+1+s)(2r+1+s)(r+1+s)}{(2(q+2r))r^2} y_n \\ &\quad + \frac{(q+2r+1+s)(2r+1+s)(1+s)}{-(q+r)r^2} y_{n-1} \\ &\quad + \frac{(q+2r+1+s)(r+1+s)(1+s)}{(2qr^2)} y_{n-2} \\ &\quad + \frac{(2r+1+s)(r+1+s)(1+s)}{-q(-q-r)(-q-2r)} y_{n-3}. \end{aligned} \tag{6}$$

Substituting $s = 0$ and $s = 1$ gives the predictor for the first and second points, respectively. Therefore by letting $r = 1$, $q = 1$, $r = 2$, $q = 2$ and $r = 1$, $q = 10/19$. This produced the following coefficients (Tables 1, 2, and 3) for the first and second points of predictor formulae for VSVO-BBDF method.

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 VSVO-BBDF of order $P3$ ($P = 3$)

$$\begin{aligned} P(x) &= P\left(x_{n+1} + sh\right) \\ &= \frac{(r+1+s)(s+1)(s)}{2r+4} y_{n+2} \\ &\quad + \frac{(r+1+s)(s+1)(s-1)}{-1-r} y_{n+1} \\ &\quad + \frac{(r+1+s)(s-1)(s)}{2r} y_n \\ &\quad + \frac{(1+s)(s-1)(s)}{-r(-1-r)(-r-2)} y_{n-1}. \end{aligned} \tag{7}$$

For VSVO-BBDF of order $P4$ ($P = 4$)

$$\begin{aligned} P(x) &= P\left(x_{n+1} + sh\right) \\ &= \frac{(2r+1+s)(r+1+s)(1+s)(s)}{2(2r+2)(r+2)} y_{n+2} \\ &\quad + \frac{(2r+1+s)(r+1+s)(1+s)(s-1)}{-(2r+1)(r+1)} y_{n+1} \\ &\quad + \frac{(2r+1+s)(r+1+s)(s)(s-1)}{4r^2} y_n \\ &\quad + \frac{(2r+1+s)(1+s)(s)(s-1)}{-r^2(-r-1)(-r-2)} y_{n-1} \\ &\quad + \frac{(r+1+s)(1+s)(s)(s-1)}{2r^2(-2r-1)(-2r-2)} y_{n-2}. \end{aligned} \tag{8}$$

Table 1: Coefficients for the first and second point of predictor formulae for VSVO-BBDF when $r = 1$, and $r = 1$, $q = 1$.

Order		y_n	y_{n-1}	y_{n-2}	y_{n-3}
$P = 3$	hf_{n+1}	2	−1		
	hf_{n+2}	3	−2		
$P = 4$	hf_{n+1}	3	−3	1	
	hf_{n+2}	6	−8	3	
$P = 5$	hf_{n+1}	4	−6	4	−1
	hf_{n+2}	10	−20	15	−4

Table 2: Coefficients for the first and second point of predictor formulae for VSVO-BBDF when $r = 2$, and $r = 2$, $q = 2$.

Order		y_n	y_{n-1}	y_{n-2}	y_{n-3}
$P = 3$	hf_{n+1}	$\frac{3}{2}$	$-\frac{1}{2}$		
	hf_{n+2}	2	−1		
$P = 4$	hf_{n+1}	$\frac{15}{8}$	$-\frac{5}{4}$	$\frac{3}{8}$	
	hf_{n+2}	3	−3	1	
$P = 5$	hf_{n+1}	$\frac{35}{16}$	$-\frac{35}{16}$	$\frac{21}{16}$	$-\frac{5}{16}$
	hf_{n+2}	−1	4	−6	4

For VSVO-BBDF of order $P5$ $(P = 5)$

$$\begin{aligned}
P(x) &= P(x_{n+1} + sh) \\
&= \Big(\big((q + 2r + 1 + s)(2r + 1 + s)(r + 1 + s)(1 + s)s\big) \\
&\quad \times \big(2(q + 2r + 2)(2r + 2)(r + 2)\big)^{-1}\Big) y_{n+2} \\
&\quad + \Big(\big((q + 2r + 1 + s)(2r + 1 + s)(r + 1 + s) \\
&\quad\quad \times (1 + s)(s - 1)\big) \\
&\quad \times \big(-(q + 2r + 1)(2r + 1)(r + 1)\big)^{-1}\Big) y_{n+1} \\
&\quad + \frac{(q + 2r + 1 + s)(2r + 1 + s)(r + 1 + s)s(s - 1)}{4(q + 2r)r^2} y_n \\
&\quad + \frac{(q + 2r + 1 + s)(2r + 1 + s)(1 + s)s(s - 1)}{-r^2(q + r)(-r - 1)(-r - 2)} y_{n-1} \\
&\quad + \frac{(q + 2r + 1 + s)(r + 1 + s)(1 + s)s(s - 1)}{2qr^2(-2r - 1)(-2r - 2)} y_{n-2} \\
&\quad + \Big(\big((2r + 1 + s)(r + 1 + s)(1 + s)s(s - 1)\big) \\
&\quad \times \big(-q(-q - r)(-q - 2r) \\
&\quad\quad \times (-q - 2r - 1)(-q - 2r - 2)\big)^{-1}\Big) y_{n-3}.
\end{aligned} \tag{9}$$

Linear Multistep Method (LMM) given in [15] is given in the definition below.

Definition 1. The linear k-step method can be represented in standard form by an equation $\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}$, where $y_{n+j} \approx y(x_{n+j})$ and $f_{n+j} \equiv f(x_{n+j}, y_{n+j})$, coefficients α_j, β_j are suitably chosen constants subject to conditions $\alpha_k = 1$, $|\alpha_0| + |\beta_0| \neq 0$, and k is defined as the order of the particular method employed. This method is said to be explicit if $\beta_k = 0$ and implicit otherwise.

Substituting $s = 0$ and $s = 1$ gives the corrector for the first and second points, respectively. Therefore by letting $r = 1$, $q = 1$, $r = 2$, $q = 2$ and $r = 1$, $q = 10/19$. This produced the following coefficients (Tables 4, 5, and 6), as in Definition 1 for the first and second points of VSVO-BBDF method.

3. Order Conditions for General VSVO-BBDF

As similar to analysis for order of Linear Multistep Method (LMM) given in [15], we use the following to determine the order of VSVO-BBDF method.

Definition 2. The LMM [15] and the associated difference operator L defined by

$$L[z(x); h] = \sum_{k=0}^{j} \left[\alpha_k z(x + kh) - h\beta_k z'(x + kh)\right] \tag{10}$$

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

$$C_q = \sum_{k=0}^{j} \left[k^q \alpha_k - \frac{1}{(q - 1)!} k^{q-1} \beta_k\right], \quad q = 2, 3, \ldots p + 1. \tag{11}$$

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

Table 3: Coefficients for the first and second point of predictor formulae for VSVO-BBDF when $r = 10/19$, and $r = 1$, $q = 10/19$.

Order		y_n	y_{n-1}	y_{n-2}	y_{n-3}
$P = 3$	hf_{n+1}	$\frac{29}{10}$	$-\frac{19}{10}$		
	hf_{n+2}	$\frac{24}{5}$	$-\frac{19}{5}$		
$P = 4$	hf_{n+1}	$\frac{1131}{200}$	$-\frac{741}{100}$	$-\frac{551}{200}$	
	hf_{n+2}	$\frac{348}{25}$	$\frac{551}{25}$	$\frac{228}{25}$	
$P = 5$	hf_{n+1}	$\frac{67}{16}$	$-\frac{201}{29}$	$\frac{67}{10}$	$-\frac{67}{32}$
	hf_{n+2}	$\frac{43}{4}$	$-\frac{688}{29}$	$\frac{129}{5}$	$-\frac{6859}{580}$

Table 4: Coefficients for the first and second point of VSVO-BBDF when $r = 1$, and $r = 1$, $q = 1$.

Order		$\alpha_{k,0}$	$\alpha_{k,1}$	$\alpha_{k,2}$	$\alpha_{k,3}$	$\alpha_{k,4}$	$\alpha_{k,5}$
$P = 3$	hf_{n+1}	$\frac{1}{6}$	-1	$\frac{1}{2}$	$\frac{1}{3}$		
	hf_{n+2}	$-\frac{1}{3}$	$\frac{3}{2}$	-3	$\frac{11}{6}$		
$P = 4$	hf_{n+1}	$-\frac{1}{12}$	$\frac{1}{2}$	$-\frac{3}{2}$	$\frac{5}{6}$	$\frac{1}{4}$	
	hf_{n+2}	$\frac{1}{4}$	$-\frac{4}{3}$	3	-4	$\frac{25}{12}$	
$P = 5$	hf_{n+1}	$\frac{1}{12}$	$-\frac{1}{2}$	$\frac{7}{6}$	$-\frac{1}{3}$	$-\frac{5}{4}$	$\frac{5}{6}$
	hf_{n+2}	$-\frac{5}{6}$	$\frac{61}{12}$	-13	$\frac{107}{6}$	$-\frac{77}{6}$	$\frac{15}{4}$

$l, m = 1, 2, \ldots, r$. Since VSVO-BBDF for variable order (P) is a block method, we extend the Definition 2 in the form of

$$L\left[z(x); h\right] = \sum_{k=0}^{j}\left[A_k z(x+kh) - hB_k z'(x+kh)\right]. \quad (12)$$

And the general form for the constant C_q is defined as

$$C_q = \sum_{k=0}^{j}\left[k^q A_k - \frac{1}{(q-1)!}k^{q-1}B_k\right], \quad q = 2, 3, \ldots p+1. \quad (13)$$

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

4. Implementation of VSVO-BBDF Method

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

$$y^{(i+1)}_{n+1,n+2} - y^{(i)}_{n+1,n+2} = -\left[(I-A) - hB\frac{\partial F}{\partial y}\left(y^{(i)}_{n+1,n+2}\right)\right]^{-1} \times\left[(I-A)\, y^{(i)}_{n+1,n+2} - hBF\left(y^{(i)}_{n+1,n+2}\right) - \xi\right]. \quad (14)$$

The general form of VSVO-BBDF method is

$$\begin{aligned} y_{n+1} &= \alpha_1 hf_{n+1} + \theta_1 y_{n+2} + \psi_1, \\ y_{n+2} &= \alpha_1 hf_{n+2} + \theta_1 y_{n+1} + \psi_2, \end{aligned} \quad (15)$$

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

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad y_{n+1,n+2} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix}, \quad B = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix},$$
$$F_{n+1,n+2} = \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}, \quad \xi_{n+1,n+2} = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}. \quad (16)$$

Equation (15) in matrix-vector form is equivalent to

$$(I-A)\, y_{n+1,n+2} = hBF_{n+1,n+2} + \xi_{n+1,n+2}. \quad (17)$$

Equation (17) is simplified as

$$\widehat{f}_{n+1,n+2} = (I-A)\, y_{n+1,n+2} - hBF_{n+1,n+2} - \xi_{n+1,n+2} = 0. \quad (18)$$

Newton iteration is performed to the system $\widehat{f}_{n+1,n+2} = 0$; by taking the analogous form of (14) where $J_{n+1,n+2} = (\partial F/\partial Y)(Y^{(i)}_{n+1,n+2})$ is the Jacobian matrix of F with respect

Table 5: Coefficients for the first and second point of VSVO-BBDF when $r = 2$, and $r = 2$, $q = 2$.

Order		$\widehat{\alpha}_{k,0}$	$\widehat{\alpha}_{k,1}$	$\widehat{\alpha}_{k,2}$	$\widehat{\alpha}_{k,3}$	$\widehat{\alpha}_{k,4}$	$\widehat{\alpha}_{k,5}$
$P = 3$	hf_{n+1}	$\frac{1}{24}$	$-\frac{3}{4}$	$\frac{1}{3}$	$\frac{3}{8}$		
	hf_{n+2}	$-\frac{1}{12}$	1	$-\frac{8}{3}$	$\frac{7}{4}$		
$P = 4$	hf_{n+1}	$-\frac{1}{80}$	$\frac{5}{48}$	$-\frac{15}{16}$	$\frac{8}{15}$	$\frac{5}{16}$	
	hf_{n+2}	$\frac{1}{30}$	$-\frac{1}{4}$	$\frac{3}{2}$	$-\frac{16}{5}$	$\frac{23}{12}$	
$P = 5$	hf_{n+1}	$\frac{1}{168}$	$-\frac{1}{24}$	$\frac{1}{8}$	$\frac{17}{24}$	$-\frac{12}{7}$	$\frac{11}{12}$
	hf_{n+2}	$-\frac{23}{336}$	$\frac{1}{2}$	$-\frac{43}{24}$	$\frac{37}{6}$	$-\frac{160}{21}$	$\frac{45}{16}$

Table 6: Coefficients for the first and second point of VSVO-BBDF when $r = 10/19$, and $r = 1$, $q = 10/19$.

Order		$\alpha_{k,0}$	$\alpha_{k,1}$	$\alpha_{k,2}$	$\alpha_{k,3}$	$\alpha_{k,4}$	$\alpha_{k,5}$
$P = 3$	hf_{n+1}	$\frac{6859}{13920}$	$-\frac{29}{20}$	$\frac{19}{29}$	$\frac{29}{96}$		
	hf_{n+2}	$-\frac{6859}{6960}$	$\frac{12}{5}$	$-\frac{96}{29}$	$\frac{91}{48}$		
$P = 4$	hf_{n+1}	$-\frac{6854}{15600}$	$\frac{89167}{46400}$	$-\frac{1131}{400}$	$\frac{1292}{1131}$	$\frac{13}{64}$	
	hf_{n+2}	$\frac{13718}{9425}$	$-\frac{6859}{1200}$	$\frac{174}{25}$	$-\frac{64}{13}$	$\frac{3095}{1392}$	
$P = 5$	hf_{n+1}	$\frac{2476099}{8020704}$	$-\frac{7}{8}$	$\frac{124}{87}$	$-\frac{47}{96}$	$-\frac{80}{67}$	$\frac{851}{1032}$
	hf_{n+2}	$-\frac{12380495}{4010352}$	$\frac{53}{6}$	$-\frac{452}{29}$	$\frac{931}{48}$	$-\frac{2692}{201}$	$\frac{165}{43}$

to Y. Equation (14) is separated to three different matrices denoted as

$$E_{1,2}^{(i+1)} = y_{n+1,n+2}^{(i+1)} - y_{n+1,n+2}^{(i)},$$
$$\widehat{A} = (I - A) - hB\frac{\partial F}{\partial Y}\left(y_{n+1,n+2}^{(i)}\right), \tag{19}$$
$$\widehat{B} = (I - A)\, y_{n+1,n+2}^{(i)} - hBF\left(y_{n+1,n+2}^{(i)}\right) - \xi_{n+1,n+2}.$$

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 $\widehat{A}$ [16]. Two-stage Newton iteration is carried out as follows.

Step 1. Compute the values for $e_{n+1,n+2}^{(i+1)} = \widehat{A}^{-1}\widehat{B}$, where

$$\widehat{A} = \begin{bmatrix} 1 - \alpha_1 h\dfrac{\partial F_{n+1}}{\partial y_{n+1}} & \theta_1 \\ \theta_2 & 1 - \alpha_2 h\dfrac{\partial F_{n+2}}{\partial y_{n+2}} \end{bmatrix},$$
$$\widehat{B} = \begin{bmatrix} y_{n+1}^{(i)} + \alpha_1 hF_{n+1}^{(i)} + \xi_1 & \theta_1 y_{n+2}^{(i)} \\ \theta_2 y_{n+1}^{(i)} & y_{n+2}^{(i)} + \alpha_2 hF_{n+2}^{(i)} + \xi_2 \end{bmatrix}. \tag{20}$$

Step 2. Calculate the corrected value for $y_{n+1,n+2}^{(i+1)}$ with the value $e_{n+1,n+2}^{(i+1)}$ from (1).

Step 3. Solve $e_{n+1,n+2}^{(i+1)} = \widehat{A}^{-1}\widehat{B}$ for the second iteration.

Step 4. Let $y_{n+1,n+2}^{(i+1)}$ be equal to the second iteration of $e_{n+1,n+2}^{(i+1)} = \widehat{A}^{-1}\widehat{B}$.

The error is defined as

$$\text{error} = \left| y_{\text{approximate}} - y_{\text{actual}} \right|, \tag{21}$$

and the maximum error is defined as

$$\text{MAXE} = \left(\max_{1<i<\text{TS}} \left(\max_{1<i<n} (\text{error}) \right) \right), \tag{22}$$

where TS is the total steps and n is the number of equations. Consequently, the rest of the method is carried out as the appendix.

5. Stability Conditions for VSVO-BBDF Method

To begin this section, we provide some definitions for the stability conditions of VSVO-BBDF method as in (15).

Table 7: Roots for different step sizes and orders.

Roots for different step size	$P = 3$	$P = 4$	$P = 5$
$r = 1$, and $r = 1$, $q = 1$	$t = -0.4347826087e - 1$, and $t = 1$	$t = -0.24414201370$, $t = 0.02079175991$, and $t = 1$	$t = -0.9671746307e - 1$, $t = 0.01195305716$, and $t = 1$
$r = 2$, and $r = 2$, $q = 2$	$t = -0.01315789474$, and $t = 1$	$t = -0.052708171410$, $t = 0.003257621961$, and $t = 1$	$t = -0.06178960900$, $t = -0.001737052270$, and $t = 1$
$r = \frac{10}{19}$, and $r = 1$, $q = \frac{10}{19}$	$t = -0.1098846524$, and $t = -1$	$t = -0.93455113330$, $t = 0.08581158625$, and $t = 1$	$t = -0.4375589404$, $t = -0.007529578675$, and $t = 1$

Definition 3. A method is said to be zero stable if the roots of the polynomial $\rho(z) = \pi(z, 0)$ satisfy the condition $z_1 = 1 \leq |z_2|$, $z_2 \neq 1$.

Definition 4. A method is said to be absolutely 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 $|r_s| < 1$, $s = 1, 2, \ldots, k$.

Definition 5. A method is said to be stiffly stable if $\mathfrak{R}_1$ and $\mathfrak{R}_2$ are contained in the absolute stability region, and it is accurate for all $h \in \mathfrak{R}_2$ when applied to the scalar test equation $y' = \lambda y$; λ is a complex constant with $R\lambda < 0$, where $\mathfrak{R}_1 = \{h\lambda \mid Rh\lambda < -t\}$, $\mathfrak{R}_2 = \{h\lambda \mid -t \leq Rh\lambda \leq u, -v \leq \coprod h\lambda < v\}$, and t, u, and v are positive constants.

The stability polynomial, $R(t, \widehat{h})$ associated with the method of (15), is given by $\det(At^2 - Bt - C)$, while the absolute stability region of this method in the $h\lambda$ plane is determined by solving $\det(At^2 - Bt - C = 0)$. Consequently, to determine zero stable, we substitute $\widehat{h} = h\lambda = 0$ to $R(t, \widehat{h})$ for each order of VSVO-BBDF method.

Hence, the roots for the three different step size and order (P) selections obtained by using Maple are listed in Table 7.

The stability region was given by the set of points determined by the boundary $t = e^{i\theta}$, $0 \leq \theta \leq 2\pi$. We obtained the stability region by finding the region for which $|t| < 1$. Since all of the roots in Table 7 have modulus less than or equal to 1, the method (15) is said to be zero stable. Figure 2 shows the stability for orders $P3$, $P4$, and $P5$ of VSVO-BBDF method, respectively. The stability regions lie outside the closed region for each case.

Based on Figure 2, VSVO-BBDF method possesses the region absolute stability, which contains almost the whole of the half-plane $\mathrm{Re}(h\lambda) < 0$.

6. Choosing Order and Step Size

To increase the efficiency in BBDF algorithm, VSVO-BBDF algorithm is designed to have the capacity to vary not only the step size, but also the order of the method employed. 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. The essential components of VSVO-BBDF algorithm are the local truncation error (LTE), strategies for deciding when to change step size and order, and techniques for changing the step size and order. Strategies proposed in [17] 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, and $P + 1$ 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 LTE for each iteration. The LTE is obtained from

$$\mathrm{LTE}_k = y_{n+2}^{(P+1)} - y_{n+2}^{(P)}, \quad P = 3, 4, 5, \tag{23}$$

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

$$\begin{aligned} h_{P-1} &= h_{\text{old}} \times \left(\frac{\mathrm{TOL}}{\mathrm{LTE}_{P-1}}\right)^{1/P}, \\ h_P &= h_{\text{old}} \times \left(\frac{\mathrm{TOL}}{\mathrm{LTE}_P}\right)^{1/(P+1)}, \\ h_{P+1} &= h_{\text{old}} \times \left(\frac{\mathrm{TOL}}{\mathrm{LTE}_{P+1}}\right)^{1/(P+2)}, \end{aligned} \tag{24}$$

where h_{old} is the step size from previous block and $h_{\max}$ is obtained from the maximum step size given in the above equations.

There are 3 cases of the possibilities in choosing the step size.

Case 1. From order 3 to order 4 and from order 4 to order 3, the step size is allowed to increase, decrease, or be maintained, that is, $(r = 1)$, $(r = 2)$ or $(r = 10/19)$.

Case 2. From order 3 to order 5 and from order 4 to order 5, the step size is allowed to decrease, or be maintained, that is, $(r = 1)$ $(q = 1)$, $(r = 2)$ $(q = 2)$, or $(r = 1)$ $(q = 10/19)$.

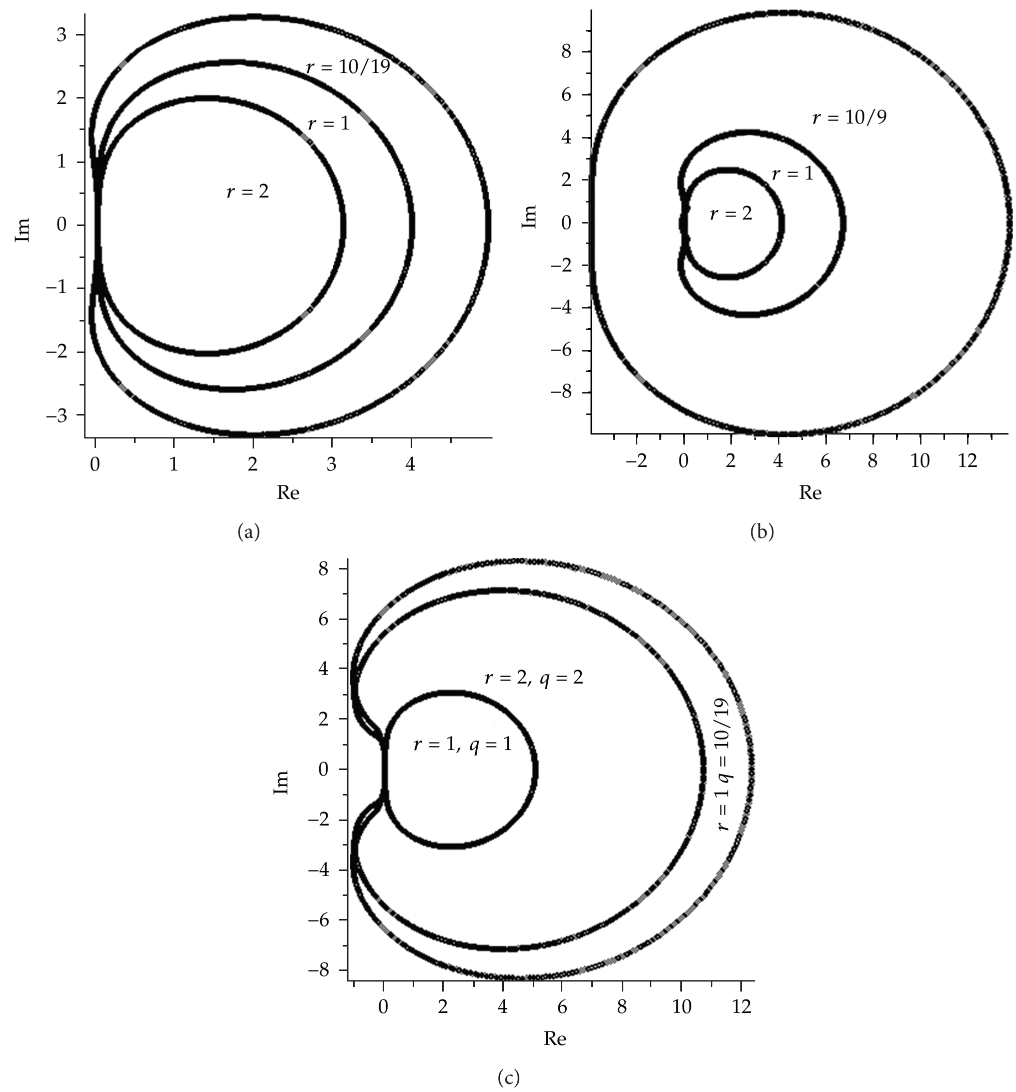


Figure 2: Stability regions for order (a) $P3$, (b) $P4$, and (c) $P5$.

Case 3. From order 5 to order 3 and from order 5 to order 4, the step size is allowed to decrease, or be maintained, that is, $(r = 1)$ $(r = 2)$ or $(r = 10/19)$.

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

$$h_{\text{new}} = c \times h_{\max} \text{ and if}$$

$$h_{\text{new}} > 1.9 \times h_{\text{old}} \text{ then } h_{\text{new}} = 1.9 \times h_{\text{old}},$$

where c is the safety factor, p is the order of the method while h_{old} and h_{new} is the step size from previous and current block, respectively. In this paper, c is set to be 0.8 so as to make sure the rejected step is being reduced.

7. Numerical Results

We carry out numerical experiments arising from problems in physics to compare the performance of VSVO-BBDF method with stiff ODE solvers in *MATLAB* version 7 in Windows XP or later. The syntax for using ode15s and ode23s is similar for every test problem. For example,

$$[X, Y] = \text{ode15s}\left(\text{“odefun”}, \text{int}, \text{init}, \text{options}\right), \qquad (25)$$

where odefun is a function stored that evaluates the right side of the differential equations, int is a vector of the integration interval, $[x_o, x_{\text{end}}]$, init is a column vector of initial conditions, $[y_0]$, and options is an adjustable format of optional parameters to change the default integration properties.

We created the above option structure to vary the relative tolerance (RelTol) and absolute tolerance (AbsTol) according to specific error tolerance. These test problems are performed under different conditions of error tolerances—(a) 10^{-2}, (b) 10^{-4}, and (c) 10^{-6}.

Table 8: Numerical results for Problem 1.

TOL	MTD	TS	AVEE	MAXE	TIME
10^{-2}	VSVO-BBDF	21	$2.9370e-005$	$2.8298e-004$	0.0103
	ode15s	28	$1.3000e-003$	$8.4000e-003$	0.0313
	ode23s	19	$1.0000e-003$	$4.5000e-003$	0.1406
10^{-4}	VSVO-BBDF	48	$1.0716e-006$	$3.2212e-006$	0.0105
	ode15s	60	$3.0358e-005$	$1.6621e-004$	0.0156
	ode32s	42	$1.1285e-004$	$2.5683e-004$	0.0313
10^{-6}	VSVO-BBDF	164	$1.6733e-008$	$3.1232e-008$	0.0115
	ode15s	100	$7.2564e-007$	$2.7506e-006$	0.0313
	ode23s	143	$7.3558e-006$	$1.2514e-005$	0.0469

Table 9: Numerical results for Problem 2.

TOL	MTD	STs	AVEE	MAXE	TIME
10^{-2}	VSVO-BBDF	22	$7.1459e-005$	$2.5736e-004$	0.0106
	ode15s	29	$9.0287e-004$	$5.2000e-003$	0.0781
	ode23s	25	$3.3626e-004$	$1.1000e-003$	0.0781
10^{-4}	VSVO-BBDF	54	$7.4173e-006$	$3.7659e-004$	0.0111
	ode15s	55	$1.7139e-005$	$8.5506e-005$	0.1250
	ode32s	118	$1.3783e-005$	$6.9774e-005$	0.2031
10^{-6}	VSVO-BBDF	194	$6.3429e-009$	$3.2882e-008$	0.0143
	ode15s	197	$2.1320e-007$	$1.0790e-006$	0.2344
	ode23s	773	$3.3163e-007$	$2.8081e-006$	0.4531

The test problems and solutions are listed below.

Problem 1. Consider,

$$y' = -100(y - x) + 1, \quad y(0) = 1, \quad 0 \le x \le 10, \tag{26}$$

with solution

$$y(x) = e^{-100x} + x. \tag{27}$$

Problem 2. Consider,

$$\begin{aligned} y_1' &= -1002 y_1 + 1000 {y_2}^2, \quad y_1(0) = 1, \quad 0 \le x \le 10, \\ y_2' &= y_1 - y_2(1 + y_2), \qquad y_2(0) = 1, \end{aligned} \tag{28}$$

with solution

$$\begin{aligned} y_1 &= e^{-2x}, \\ y_2 &= e^{-x}. \end{aligned} \tag{29}$$

See Kaps [18].

Problem 3. Consider,

$$\begin{aligned} y_1' &= -2y_1 + y_2 + 2\sin(x), \quad y_1(0) = 2, \quad 0 \le x \le 10, \\ y_2' &= 998 y_1 - 999 y_2 + 999(\cos(x) - \sin(x)), \qquad y_2(0) = 3, \end{aligned} \tag{30}$$

with solution

$$\begin{aligned} y_1(x) &= 2e^{-x} + \sin(x), \\ y_2(x) &= 2e^{-x} + \cos(x). \end{aligned} \tag{31}$$

See Lambert [15].

This paper considers the comparison of four different factors, namely, number of steps taken, average error, maximum error, and computation time. From Table 8, among the three methods tested, our method, VSVO-BBDF method, requires the shortest execution time, smallest maximum error, and average error for each given tolerance level. From Figure 3, we can see more clearly that VSVO-BBDF gives the lowest maximum error for every tolerance level.

Again by comparing the four factors mentioned earlier, we can see VSVO-BBDF in Table 9 gives the minimum maximum error for every tolerance level except for TOL 10e–4. However, our method prevails in terms of average error for each given tolerance level. VSVO-BBDF once again requires the shortest execution time for each given tolerance level. Figure 4 illustrates the efficiency of VSVO-BBDF as compared to ode15s and ode23s.

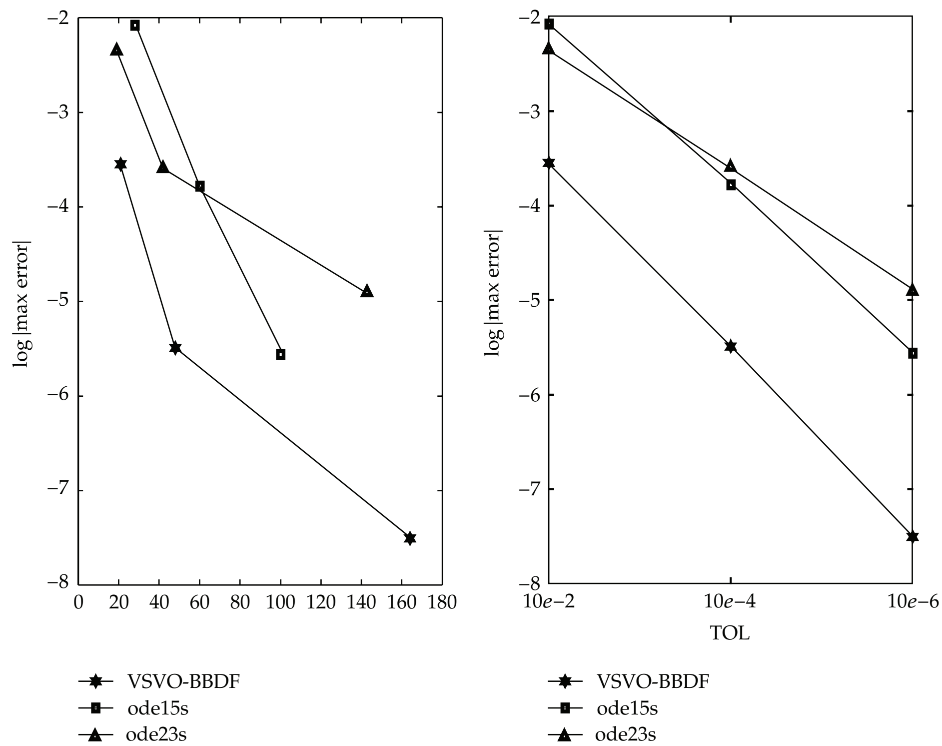
From Table 10, VSVO-BBDF method gives the shortest execution time for every given tolerance level. While in terms of average error and maximum error, VSVO-BBDF method once again gives the best result as compared to ode15s and ode23s. Figure 5 clarifies the efficiency of the proposed method based on its total steps and tolerance level.

8. Conclusion

The objective is met when VSVO-BBDF method outperformed ode15s and ode23s in terms of average error as well as maximum error. In most of the cases, VSVO-BBDF has successfully managed to reduce the number of total steps taken. As for the computation time wise, it gave lesser values

Table 10: Numerical results for Problem 3.

TOL	MTD	TSs	AVEE	MAXE	TIME
	VSVO-BBDF	35	4.6584e − 005	3.0045e − 004	0.0111
10^{-2}	ode15s	45	6.9000e − 003	1.4620e − 002	0.0156
	ode23s	137	4.0745e − 004	3.0000e − 002	0.2031
	VSVO-BBDF	84	2.5775e − 006	1.1002e − 005	0.3281
10^{-4}	ode15s	93	7.6548e − 005	2.7591e − 004	0.0469
	ode32s	1211	2.4915e − 005	6.3075e − 005	0.0156
	VSVO-BBDF	380	2.4244e − 008	8.9627e − 008	0.0199
10^{-6}	ode15s	186	2.1681e − 006	6.1936e − 006	0.0781
	ode23s	2829	5.3155e − 007	1.5667e − 006	2.3281



(a) (b)

Figure 3: Efficiency curves for Problem 1.

for all cases. Therefore, we can conclude that VSVO-BBDF can serve as an alternative solver for solving stiff ordinary differential equations which arise in engineering and applied sciences.

Appendix

The following notation is used in the algorithm of VSVO-BBDF:

h: step size,

ε: tolerance,

TS: total steps,

Order: order of the method,

JCBN: jacobian,

E: increment.

The algorithm for VSVO-BBDF code is given as follows.

Step 1. Let order = 3.

Step 2. Calculate initial step size (h) and ε.

Step 3. Calculate initial array $y_1, \ldots y_i$, i = order − 1.

Step 4. Calculate the predictor values for y_{i+1}, y_{i+2}, i = order − 1.

Step 5. Calculate F_i for $i = 1, \ldots,$ order + 1.

Step 6. Calculate the corrector values for y_{i+1}, y_{i+2}, i = order − 1.

Step 7. Calculate F_i for $i = 1, \ldots,$ order + 1.

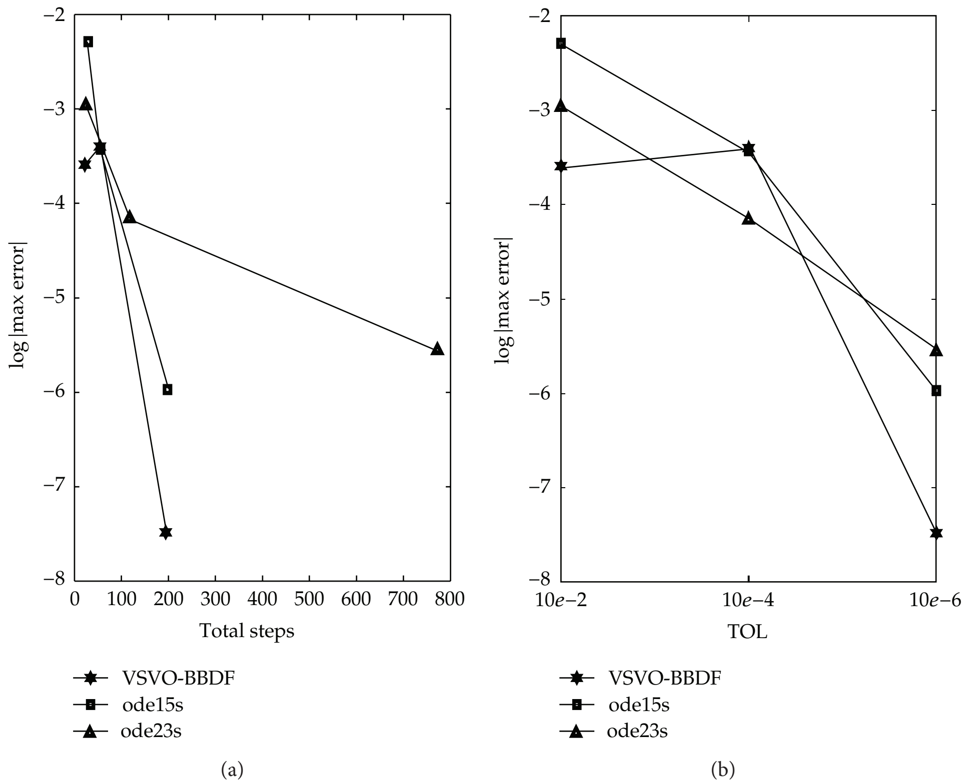


Figure 4: Efficiency curves for Problem 2.

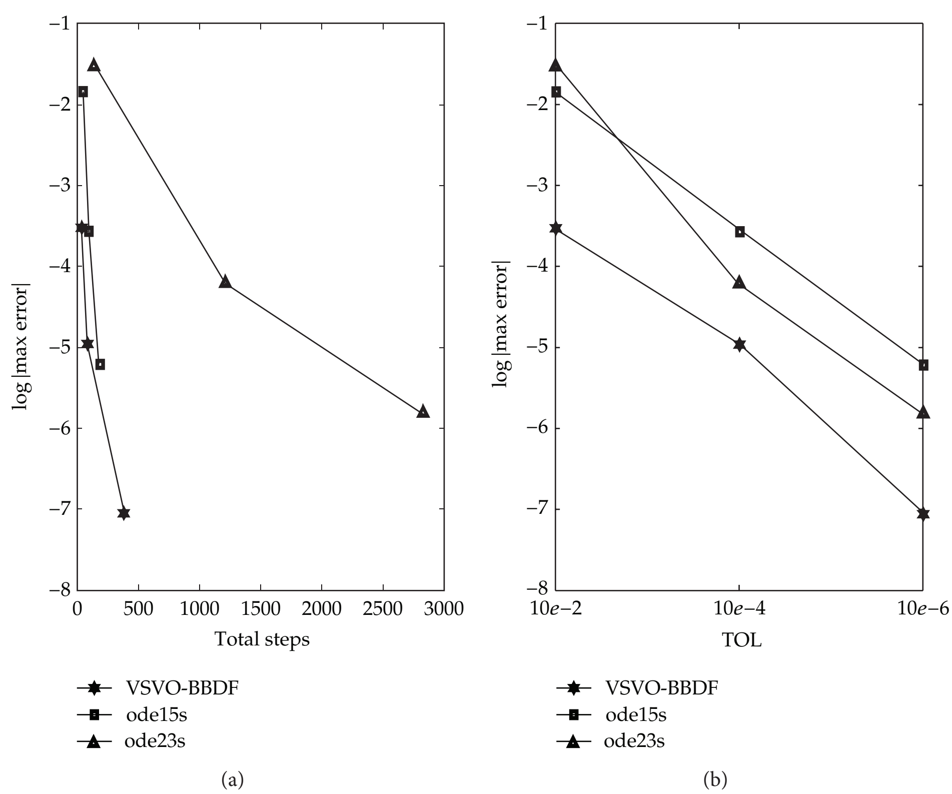


Figure 5: Efficiency curves for Problem 3.

Implementation of VSVO-BBDF.

Step 8. Calculate Jacobian (JACBN).

Step 9. Perform LU-Decomposition to calculate the increment E.

Consider,

$$E_{1,2}^{(i+1)} = y_{n+1,n+2}^{(i+1)} - y_{n+1,n+2}^{(i+1)}, \quad n = \text{order} - 1. \tag{A.1}$$

Test for convergence.

Step 10. If $E > \varepsilon$, convergence is false. Else go to Step 11.

Step 11. Calculate LTE_i, $i = \text{order}$.

Change order.

Step 12. Let order equal 4 and order equal 5.

Step 13. Repeat from Step 2 until Step 11 for each order correspondingly.

Step size and order control.

Step 14. Calculate $h_{\max}$ for choosing the new step size.

Consider,

$$h_k = h_{\text{old}} \times \left(\frac{\text{TOL}}{\text{LTE}_{2,k}}\right)^{1/(k+1)}. \tag{A.2}$$

The final step size after a successful step ($\text{LTE} < \varepsilon$) is given by

$$h_{\text{new}} = c \times h_{\text{old}} \times \left(\frac{\text{TOL}}{\text{LTE}}\right)^{1/p}. \tag{A.3}$$

The final step size after a failure step ($\text{LTE} > \varepsilon$) is given by

$$h_{\text{new}} \geq \frac{1}{2} \times h_{\text{old}}. \tag{A.4}$$

Step 15. If $h_{\text{new}} \geq 1.9 \times h_{\text{old}}$ then $h_{\text{new}} = 1.9 \times h_{\text{old}}$.

Else $h_{\text{new}} = h_{\text{old}}$.

Step 16. Update values:

$$y_i = y_{i+2}, \quad x_i = x_{i+2} \quad \text{for } i = 1, \ldots, \text{order} - 1. \tag{A.5}$$

Step 17. Repeat Step 1 to Step 16 until $h = \text{end point}$.

The Abbreviations Used in Figures 3–5 and Tables 8–10

TSs: The total number of steps taken
TOL: The initial value for the local error estimate
MAXE: The maximum error
AVEE: The average error
MTD: The method used
TIME: The total execution time (seconds).

References

[1] W. H. Enright, T. E. Hull, and B. Lindberg, "Comparing numerical methods for stiff systems of O.D.E:s," *BIT*, vol. 15, no. 1, pp. 10–48, 1975.

[2] G. D. Byrne, A. C. Hindmarsh, K. R. Jackson, and H. G. Brown, "A comparison of two ode codes: gear and episode," *Computers and Chemical Engineering*, vol. 1, no. 2, pp. 125–131, 1977.

[3] R. Sacks-Davis, "Fixed leading coefficient implementation of sd-formulas for stiff ODES," *ACM Transactions on Mathematical Software*, vol. 6, no. 4, pp. 540–562, 1980.

[4] A. S. Mahmood, L. Casasús, and W. Al-Hayani, "The decomposition method for stiff systems of ordinary differential equations," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 964–975, 2005.

[5] J. Ibáñez, V. Hernández, E. Arias, and P. A. Ruiz, "Solving initial value problems for ordinary differential equations by two approaches: BDF and piecewise-linearized methods," *Computer Physics Communications*, vol. 180, no. 5, pp. 712–723, 2009.

[6] H. Aminikhah and J. Biazar, "A new analytical method for system of ODEs," *Numerical Methods for Partial Differential Equations*, vol. 26, no. 5, pp. 1115–1124, 2010.

[7] S. Abelman and K. C. Patidar, "Comparison of some recent numerical methods for initial-value problems for stiff ordinary differential equations," *Computers and Mathematics with Applications*, vol. 55, no. 4, pp. 733–744, 2008.

[8] L. F. Shampine and M. W. Reichelt, "The MATLAB ode suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1–22, 1997.

[9] L. F. Shampine, M. W. Reichelt, and J. A. Kierzenka, "Solving index-I DAEs in MATLAB and simulink," *SIAM Review*, vol. 41, no. 3, pp. 538–552, 1999.

[10] T. Kato and K. Ikeuchi, "Variable order and variable step-size integration method for transient analysis programs," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 206–213, 1991.

[11] Z. B. Ibrahim, M. Suleiman, and K. I. Othman, "Fixed coefficients block backward differentiation formulas for the numerical solution of stiff ordinary differential equations," *European Journal of Scientific Research*, vol. 21, no. 3, pp. 508–520, 2008.

[12] Z. B. Ibrahim, K. I. Othman, and M. B. Suleiman, "Variable step size block backward differentiation formula for solving stiff odes," in *Proceedings of the World Congress on Engineering*, vol. 2, pp. 785–789, London, UK, 2007.

[13] S. A. M. Yatim, Z. B. Ibrahim, K. I. Othman, and F. Ismail, "Fifth order variable step block backward differentiation formulae for solving stiff ODEs," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 62, pp. 998–1000, 2010.

[14] S. A. M. Yatim, Z. B. Ibrahim, K. I. Othman, and M. B. Suleiman, "Quantitative comparison of numerical method for solving stiff ordinary differential equations," *Mathematical Problems in Engineering*, vol. 2011, Article ID 193691, 12 pages, 2011.

[15] J. D. Lambert, *Computational Methods in Ordinary Differential Equation*, John Wiley & Sons, New York, NY, USA, 1991.

[16] K. R. Jackson, "The numerical solution of large systems of stiff IVPs for ODEs," *Applied Numerical Mathematics*, vol. 20, no. 1-2, pp. 5–20, 1996.

[17] G. Hall and J. M. Watt, *Modern Numerical Methods for Ordinary Differential Equations*, Clarendon Press, Oxford, UK, 1976.

[18] P. Kaps and G. Wanner, "A study of Rosenbrock-type methods of high order," *Numerische Mathematik*, vol. 38, no. 2, pp. 279–298, 1981.