

High Order Block Method for Third Order ODEs

A. I. Asnor¹, S. A. M. Yatim¹, Z. B. Ibrahim² and N. Zainuddin³

¹School of Distance Education, Universiti Sains Malaysia, USM Penang, 11800, Malaysia

²Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Serdang, Selangor Darul Ehsan, 43400, Malaysia

³Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Perak Darul Ridzuan, 32610, Malaysia

*Corresponding Author: S. A. M. Yatim. Email: ainor@usm.my

Received: 16 October 2020; Accepted: 28 November 2020

Abstract: Many initial value problems are difficult to be solved using ordinary, explicit step-by-step methods because most of these problems are considered stiff. Certain implicit methods, however, are capable of solving stiff ordinary differential equations (ODEs) usually found in most applied problems. This study aims to develop a new numerical method, namely the high order variable step variable order block backward differentiation formula (VSVO-HOBBDF) for the main purpose of approximating the solutions of third order ODEs. The computational work of the VSVO-HOBBDF method was carried out using the strategy of varying the step size and order in a single code. The order of the proposed method was then discussed in detail. The advancement of this strategy is intended to enhance the efficiency of the proposed method to approximate solutions effectively. In order to confirm the efficiency of the VSVO-HOBBDF method over the two ODE solvers in MATLAB, particularly ode15s and ode23s, a numerical experiment was conducted on a set of stiff problems. The numerical results prove that for this particular set of problem, the use of the proposed method is more efficient than the comparable methods. VSVO-HOBBDF method is thus recommended as a reliable alternative solver for the third order ODEs.

Keywords: Block method; stiff ODEs; third order; variable step variable order

1 Introduction

Differential equations are among the most important mathematical tools used in producing models in fields such as the physical sciences, engineering, and biological sciences. In general, the ordinary differential equation (ODE) is an equation that consists of a function and one or more derivatives. A function that satisfies the differential equation is a solution of a differential equation when the function and its derivatives are substituted in the equation. The real-life problems found in science and engineering can be converted into mathematical models in the form of lower or



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

higher order ODEs. The problem of higher order ODEs to be considered in this paper is the third-order stiff ODEs,

$$y''' = f(x, y, y', y''), \quad x \in [a, z] \quad (1)$$

with its initial conditions $y(a) = y_0$, $y'(a) = y'_0$, $y''(a) = y''_0$, where a and z are finite that represent the starting point and the endpoint respectively.

Some difficult problems may need a solver that can approximate their solutions since these problems may not have actual solutions. In fact, the actual solutions of complex problems are impossible to find. These problems can instead be solved using numerical methods that are tailored to specific cases. There are two important cases that need to be assured in ODEs, particularly stiff or non-stiff cases. Only certain numerical methods can provide suitable solvers for stiff or non-stiff problems. The problems are stiff if they satisfy the following conditions [1],

i. $\text{Re}(\lambda_i) < 0$, $i = 1, 2, \dots, n$ and

ii. $\max |\text{Re}(\lambda_i)| \gg \min |\text{Re}(\lambda_i)|$ where λ_i are the eigenvalues of the Jacobian matrix, $\frac{\partial f}{\partial y}$ and

the ratio $\frac{\max_i |\text{Re}(\lambda_i)|}{\min_i |\text{Re}(\lambda_i)|}$ is called the stiffness ratio of stiffness index.

Unlike non-stiff problems, stiff problems require an implicit method in finding the approximate solutions because the explicit method cannot handle this kind of problems effectively since a very small value of step size is needed. Currently, block backward differentiation formula (BBDF) method is perceived as another solver for stiff ODEs. This method is known to perform better than the non-block backward differentiation formula (BDF) method and MATLAB's ODE solver [2–4]. BBDF method can produce better approximate solutions and provide numerous solutions simultaneously. By providing numerous solutions simultaneously, the method can significantly reduce computational effort and cost.

In addition to producing numerous solutions simultaneously, BBDF method also requires less computational effort as it is capable of solving higher order problems directly. BBDF method can directly solve higher order problems because it does not require a reduction process to reduce the higher order problems to their first order system before the computational work can start. This will automatically reduce the computational cost, especially in terms of time and the number of total steps taken. The effectiveness of BBDF method in solving higher order problems have been proven in a number of research [5–9]. Albeit implementing different approaches to BBDF method such as constant step size, variable step size, variable order with constant step size in a single code and also variable order with variable step size in a single code for the direct solutions of second order ODEs, these researches have all favoured the method for its effectiveness.

The evolution of various numerical methods in the literature is intended to improve the existing methods by offering better solutions. However, there is a dearth of research in methods that solve higher-order stiff ODEs, especially third-order stiff ODEs. Most past research focused on the special and non-stiff third order ODEs as discussed in [10–13]. We, therefore, aim to develop an efficient direct method to provide numerical solutions for third-order ODEs.

Inspired by the approach applied in a couple of research [6,7], we intend to adopt the variable step and variable order approach in BBDF method for solving the third order stiff ODEs directly. Choosing a suitable step size is imperative in reducing the computation time and the number

of iterations, while changing the order helps in finding the best approximation that will produce better or more accurate results.

2 Development of VSVO-HOBBDF Method

This section will explain the development of the high order variable step variable order block backward differentiation formula (VSVO-HOBBDF). VSVO-HOBBDF method consists of order two, three, and four variable step higher order BBDF (VS-HOBBDF) method of different values of the step size in a single code. The illustration of the variable step BBDF method is shown in Fig. 1.

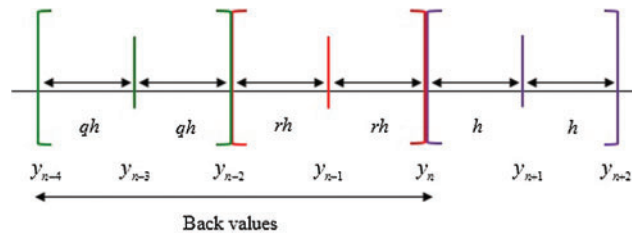


Figure 1: Illustration of variable step BBDF method

h , r , and q in Fig. 1 are defined as step size, step size ratio for the first previous block, and step size ratio for the second previous block respectively. y_{n-4} , y_{n-3} , y_{n-2} , y_{n-1} and y_n are all the back values while y_{n+1} and y_{n+2} are the two new points that we are going to approximate. For each order of the VS-HOBBDF method, the back values will be different. The k back values for each order of the VS-HOBBDF method are specified as follows:

- Order two VS-HOBBDF (2VS-HOBBDF)

$k = 3$ back values: y_{n-2}, y_{n-1}, y_n

- Order three VS-HOBBDF (3VS-HOBBDF)

$k = 4$ back values: $y_{n-3}, y_{n-2}, y_{n-1}, y_n$

- Order four VS-HOBBDF (4VS-HOBBDF)

$k = 5$ back values: $y_{n-4}, y_{n-3}, y_{n-2}, y_{n-1}, y_n$

The VS-HOBBDF formulas for order 2, 3, and 4 are derived from Lagrange polynomial. To derive each order of VS-HOBBDF, all the back values and the two new current points will be the interpolation points. We begin the development with 2VS-HOBBDF method.

Development of the First and Second Points of Order Two, Three and Four VS-HOBBDF Method

We begin with the development of 2VS-HOBBDF method as follows:

- Step 1:

Taking $k = 3$ gives the Lagrange polynomial for 2VS-HOBBDF given by

$$P_k(x) = \sum_{j=0}^{k+1} Y_j \cdot \prod_{\substack{i=0 \\ i \neq j}}^{k+1} \frac{(x - x_{n+2-i})}{(x_{n+2-j} - x_{n+2-i})} \tag{2}$$

where $x = s \cdot h + x_{n+1}$ and $Y = y(x_{n+2-j})$.

- Step 2:

Replacing $f(x, y, y', y'')$ in (1) by polynomial (2). Subsequently, referring to Fig. 1, the step size of the computed block is $2h$ whereas the step sizes for the previous blocks are $2rh$ and $2qh$. Then, substituting them in the polynomials (2) afterward give the polynomials in terms of $s, h,$ and r .

- Step 3:

Differentiating the updated polynomials once, twice and thrice with respect to s , then followed by replacing s with 0 at $x = x_{n+1}$. The polynomials are now in terms of r only.

- Step 4:

Consequently, replacing r term in first, second and third differentiation with 1, 2 and 10/19 give

- $r = 1$ (Constant step size)

$$\begin{aligned} hy'_{n+1} &= -\frac{1}{12}y_{n-2} + \frac{1}{2}y_{n-1} - \frac{3}{2}y_n + \frac{5}{6}y_{n+1} + \frac{1}{4}y_{n+2} \\ h^2y''_{n+1} &= -\frac{1}{12}y_{n-2} + \frac{1}{3}y_{n-1} + \frac{1}{2}y_n - \frac{5}{3}y_{n+1} + \frac{11}{12}y_{n+2} \\ h^3y'''_{n+1} &= \frac{1}{2}y_{n-2} - 3y_{n-1} + 6y_n - 5y_{n+1} + \frac{3}{2}y_{n+2} \end{aligned} \quad (3)$$

- $r = 2$ (Halving the step size)

$$\begin{aligned} hy'_{n+1} &= -\frac{1}{80}y_{n-2} + \frac{5}{48}y_{n-1} - \frac{15}{16}y_n + \frac{8}{15}y_{n+1} + \frac{5}{16}y_{n+2} \\ h^2y''_{n+1} &= -\frac{1}{120}y_{n-2} + \frac{1}{24}y_{n-1} + \frac{7}{8}y_n - \frac{28}{15}y_{n+1} + \frac{23}{24}y_{n+2} \\ h^3y'''_{n+1} &= \frac{3}{40}y_{n-2} - \frac{5}{8}y_{n-1} + \frac{21}{8}y_n - \frac{16}{5}y_{n+1} + \frac{9}{8}y_{n+2} \end{aligned} \quad (4)$$

- $r = \frac{10}{19}$ (Increment of the step size by a factor of 1.9)

$$\begin{aligned} hy'_{n+1} &= -\frac{6859}{15600}y_{n-2} + \frac{89167}{46400}y_{n-1} - \frac{1131}{400}y_n + \frac{1292}{1131}y_{n+1} + \frac{13}{64}y_{n+2} \\ h^2y''_{n+1} &= -\frac{130321}{226200}y_{n-2} + \frac{130321}{69600}y_{n-1} - \frac{161}{200}y_n - \frac{1540}{1131}y_{n+1} + \frac{2423}{2784}y_{n+2} \\ h^3y'''_{n+1} &= \frac{6859}{2600}y_{n-2} - \frac{267501}{23200}y_{n-1} + \frac{2793}{200}y_n - \frac{2584}{377}y_{n+1} + \frac{57}{32}y_{n+2} \end{aligned} \quad (5)$$

- Step 5:

Repeating Step 3 while replacing s with 1 at $x = x_{n+2}$ and continuing the process with Step 4 to obtain the following equations:

- $r = 1$ (Constant step size)

$$\begin{aligned}
 hy'_{n+2} &= \frac{1}{4}y_{n-2} - \frac{4}{3}y_{n-1} + 3y_n - 4y_{n+1} + \frac{25}{12}y_{n+2} \\
 h^2y''_{n+2} &= \frac{11}{12}y_{n-2} - \frac{14}{3}y_{n-1} + \frac{19}{2}y_n - \frac{26}{3}y_{n+1} + \frac{35}{12}y_{n+2} \\
 h^3y'''_{n+2} &= \frac{3}{2}y_{n-2} - 7y_{n-1} + 12y_n - 9y_{n+1} + \frac{5}{2}y_{n+2}
 \end{aligned} \tag{6}$$

- $r = 2$ (Halving the step size)

$$\begin{aligned}
 hy'_{n+2} &= \frac{1}{30}y_{n-2} - \frac{1}{4}y_{n-1} + \frac{3}{2}y_n - \frac{16}{5}y_{n+1} + \frac{23}{12}y_{n+2} \\
 h^2y''_{n+2} &= \frac{7}{60}y_{n-2} - \frac{5}{6}y_{n-1} + \frac{17}{4}y_n - \frac{88}{15}y_{n+1} + \frac{7}{3}y_{n+2} \\
 h^3y'''_{n+2} &= \frac{7}{40}y_{n-2} - \frac{9}{8}y_{n-1} + \frac{33}{8}y_n - \frac{24}{5}y_{n+1} + \frac{13}{8}y_{n+2}
 \end{aligned} \tag{7}$$

- $r = \frac{10}{19}$ (Increment of the step size by a factor of 1.9)

$$\begin{aligned}
 hy'_{n+2} &= \frac{13718}{9425}y_{n-2} - \frac{6859}{1200}y_{n-1} + \frac{174}{25}y_n - \frac{64}{13}y_{n+1} + \frac{3095}{1392}y_{n+2} \\
 h^2y''_{n+2} &= \frac{48013}{8700}y_{n-2} - \frac{363527}{17400}y_{n-1} + \frac{2399}{100}y_n - \frac{1048}{87}y_{n+1} + \frac{2387}{696}y_{n+2} \\
 h^3y'''_{n+2} &= \frac{144039}{15080}y_{n-2} - \frac{157757}{4640}y_{n-1} + \frac{285}{8}y_n - \frac{5472}{377}y_{n+1} + \frac{3097}{928}y_{n+2}
 \end{aligned} \tag{8}$$

- Step 6:

Repeating Steps 1–5 for deriving the formulae of 3VS-HOBBDF ($k = 4$) and 4VS-HOBBDF ($k = 5$) methods.

The formulae of VSVO-HOBBDF can be generalised in matrix form as follows:

$$\begin{bmatrix} y_{n+1} \\ y'_{n+1} \\ y''_{n+1} \\ y_{n+2} \\ y'_{n+2} \\ y''_{n+2} \end{bmatrix} = \begin{bmatrix} \theta_{k,1}y_{n+2} + h^3\alpha_{k,1}f_{n+1} + B_{k,1} \\ \frac{1}{h}(\beta_{k,1}y_{n+1} + \hat{\beta}_{k,1}y_{n+2} + B'_{k,1}) \\ \frac{1}{h^2}(\gamma_{k,1}y_{n+1} + \hat{\gamma}_{k,1}y_{n+2} + B''_{k,1}) \\ \theta_{k,2}y_{n+1} + h^3\alpha_{k,2}f_{n+2} + B_{k,2} \\ \frac{1}{h}(\beta_{k,2}y_{n+1} + \hat{\beta}_{k,2}y_{n+2} + B'_{k,2}) \\ \frac{1}{h^2}(\gamma_{k,2}y_{n+1} + \hat{\gamma}_{k,2}y_{n+2} + B''_{k,2}) \end{bmatrix} \quad (9)$$

where $B_{k,1}$, $B'_{k,1}$, $B''_{k,1}$, $B_{k,2}$, $B'_{k,2}$, and $B''_{k,2}$, $k = 3, 4, 5$ are the back values for the first and second point respectively.

3 Implementation of VSVO-HOBBDF Method

This section presents the formulae in the previous section as implemented in two stages of Newton's iteration. Therefore, the general form of linear equation $\hat{E} = \hat{M}^{-1}\hat{N}$ is given by

$$\hat{E} = \begin{bmatrix} e_{n+1} \\ e_{n+2} \end{bmatrix}^{i+1}, \quad \hat{M} = \begin{bmatrix} 1 - P - Q - R & -\theta_{k,1} - Y - Z \\ -\theta_{k,2} - W - X & 1 - T - V - C \end{bmatrix},$$

$$\hat{N} = \begin{bmatrix} -1 & \theta_{k,1} \\ \theta_{k,2} & -1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} + h^3 \begin{bmatrix} \alpha_{k,1} \\ \alpha_{k,2} \end{bmatrix} \begin{bmatrix} f(y_{n+1}, y'_{n+1}, y''_{n+1}) \\ f(y_{n+2}, y'_{n+2}, y''_{n+2}) \end{bmatrix} + \begin{bmatrix} B_{k,1} \\ B_{k,2} \end{bmatrix},$$

where,

$$P = h^3\alpha_{k,1} \left(\frac{\partial f_{n+1}}{\partial y_{n+1}} \right), \quad Q = h^2\alpha_{k,1}\beta_{k,1} \left(\frac{\partial f_{n+1}}{\partial y'_{n+1}} \right), \quad R = h\alpha_{k,1}\gamma_{k,1} \left(\frac{\partial f_{n+1}}{\partial y''_{n+1}} \right),$$

$$Y = h^2\alpha_{k,1}\hat{\beta}_{k,1} \left(\frac{\partial f_{n+1}}{\partial y'_{n+1}} \right), \quad Z = h\alpha_{k,1}\hat{\gamma}_{k,1} \left(\frac{\partial f_{n+1}}{\partial y''_{n+1}} \right),$$

$$T = h^3\alpha_{k,2} \left(\frac{\partial f_{n+2}}{\partial y_{n+2}} \right), \quad V = h^2\alpha_{k,2}\hat{\beta}_{k,2} \left(\frac{\partial f_{n+2}}{\partial y'_{n+2}} \right), \quad C = h\alpha_{k,2}\hat{\gamma}_{k,2} \left(\frac{\partial f_{n+2}}{\partial y''_{n+2}} \right),$$

$$W = h^2\alpha_{k,2}\beta_{k,2} \left(\frac{\partial f_{n+2}}{\partial y'_{n+2}} \right), \quad X = h\alpha_{k,2}\gamma_{k,2} \left(\frac{\partial f_{n+2}}{\partial y''_{n+2}} \right).$$

4 Order of VSVO-HOBBDF Method

The linear difference operator, $L[y(x); h]$ associated with the VSVO-HOBBDF method is given by

$$L[y(x); h] = \sum_{j=0}^k \left[\alpha_j y(x+jh) - h\beta_j y'(x+jh) - h^2\gamma_j y''(x+jh) - h^3\lambda_j y'''(x+jh) \right] \tag{10}$$

where $y(x)$ is an arbitrary function, continuously differentiable on $[a, b]$.

Expanding $y(x+jh)$, $y'(x+jh)$, $y''(x+jh)$ and $y'''(x+jh)$ in (10), as Taylor Series about x . Then, collecting the terms obtain

$$L[y(x); h] = C_0 y(x) + C_1 h y'(x) + C_2 h^2 y''(x) + C_3 h^3 y'''(x) + \dots + C_q h^q y^{(q)}(x) + \dots \tag{11}$$

where coefficients of C_q are constants and assign as follows:

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

$$C_1 = \sum_{j=0}^k (j\alpha_j - \beta_j),$$

$$C_2 = \sum_{j=0}^k \left(\frac{j^2\alpha_j}{2!} - j\beta_j - \gamma_j \right),$$

$$C_3 = \sum_{j=0}^k \left(\frac{j^3\alpha_j}{3!} - \frac{j^2\beta_j}{2!} - j\gamma_j - \lambda_j \right),$$

⋮

$$C_q = \sum_{j=0}^k \left(\frac{j^q\alpha_j}{q!} - \frac{j^{q-1}\beta_j}{(q-1)!} - \frac{j^{q-2}\gamma_j}{(q-2)!} - \frac{j^{q-3}\lambda_j}{(q-3)!} \right), \quad q = 2, 3, 4, \dots$$

Definition 3

The linear difference operator and the associate LMM in (10) are said to be of order p if $C_0 = C_1 = \dots = C_{p+2} = 0$ and $C_{p+3} \neq 0$. Note that C_{p+3} is the error constant.

Applying the following steps to determine the order of each VS-HOBBDF methods by letting the simpler form of the formulae be equivalent to

$$\sum_{j=0}^s \alpha_j y_{n+j} = h \sum_{j=0}^s \beta_j y'_{n+j} + h^2 \sum_{j=0}^s \gamma_j y''_{n+j} + h^3 \sum_{j=0}^s \lambda_j f_{n+j} \tag{12}$$

where α_j , β_j , γ_j and λ_j are not all zero.

We begin by determining the order of 2VS-HOBBDF method. Rearranging the formulae of 2VS-HOBBDF method in the previous section gives

$$\begin{bmatrix} \frac{1}{4} & \frac{5}{6} & -\frac{3}{2} & \frac{1}{2} & -\frac{1}{12} & 0 \\ \frac{11}{12} & -\frac{5}{3} & \frac{1}{2} & \frac{1}{3} & -\frac{1}{12} & 0 \\ -\frac{3}{10} & 1 & -\frac{6}{5} & \frac{3}{5} & -\frac{1}{10} & 0 \\ \frac{25}{12} & -4 & 3 & -\frac{4}{3} & \frac{1}{4} & 0 \\ \frac{35}{12} & -\frac{26}{3} & \frac{19}{2} & -\frac{14}{3} & \frac{11}{12} & 0 \\ 1 & -\frac{18}{5} & \frac{24}{5} & -\frac{14}{5} & \frac{3}{5} & 0 \end{bmatrix} \begin{bmatrix} y_{n+2} \\ y_{n+1} \\ y_n \\ y_{n-1} \\ y_{n-2} \\ y_{n-3} \end{bmatrix} = h \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'_{n+2} \\ y'_{n+1} \\ y'_n \\ y'_{n-1} \\ y'_{n-2} \\ y'_{n-3} \end{bmatrix} + h^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \times \begin{bmatrix} y''_{n+2} \\ y''_{n+1} \\ y''_n \\ y''_{n-1} \\ y''_{n-2} \\ y''_{n-3} \end{bmatrix} + h^3 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{5} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y'''_{n+2} \\ y'''_{n+1} \\ y'''_n \\ y'''_{n-1} \\ y'''_{n-2} \\ y'''_{n-3} \end{bmatrix}$$

Subsequently, α_j , β_j , γ_j and λ_j represent each column of the matrices as follows:

$$\alpha_1 = \begin{bmatrix} -\frac{1}{12} \\ -\frac{1}{12} \\ -\frac{1}{10} \\ \frac{1}{4} \\ \frac{11}{12} \\ \frac{3}{5} \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \\ \frac{3}{5} \\ -\frac{4}{3} \\ -\frac{14}{3} \\ -\frac{14}{5} \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} -\frac{3}{2} \\ \frac{1}{2} \\ -\frac{6}{5} \\ 3 \\ \frac{19}{2} \\ \frac{24}{5} \end{bmatrix}, \quad \alpha_4 = \begin{bmatrix} \frac{5}{6} \\ -\frac{5}{3} \\ 1 \\ -4 \\ -\frac{26}{3} \\ -\frac{18}{5} \end{bmatrix}, \quad \alpha_5 = \begin{bmatrix} \frac{1}{4} \\ \frac{11}{12} \\ -\frac{3}{10} \\ \frac{25}{12} \\ \frac{35}{12} \\ 1 \end{bmatrix},$$

$$\beta_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \beta_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \gamma_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \gamma_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \lambda_4 = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{5} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \lambda_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{2}{5} \end{bmatrix},$$

$$\alpha_0 = \beta_0 = \beta_1 = \beta_2 = \beta_3 = \gamma_0 = \gamma_1 = \gamma_2 = \gamma_3 = \lambda_0 = \lambda_1 = \lambda_2 = \lambda_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Obtain the values for C_q as follows:

$$C_0 = C_1 = C_2 = C_3 = C_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C_5 = \begin{bmatrix} \frac{1}{20} \\ \frac{1}{12} \\ \frac{1}{20} \\ -\frac{1}{5} \\ \frac{5}{6} \\ -\frac{7}{10} \end{bmatrix}.$$

Since $C_5 \neq 0$, from the Definition 3 gives

$$C_5 = C_{p+3},$$

$$p + 3 = 5,$$

$$p = 2.$$

Thereby, the 2VS-HOBBDF is of order two with error constant $\left[\frac{1}{20}, \frac{1}{12}, \frac{1}{20}, -\frac{1}{5}, -\frac{5}{6}, -\frac{7}{10} \right]^T$.

Repeating the similar steps to determine the order of 3VS-HOBBDF and 4VS-HOBBDF methods. The values of C_q for 3VS-HOBBDF and 4VS-HOBBDF methods are obtained as follows:

• 3VS-HOBBDF

$$C_0 = C_1 = C_2 = C_3 = C_4 = C_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C_6 = \begin{bmatrix} \frac{1}{30} \\ \frac{13}{180} \\ \frac{1}{50} \\ -\frac{1}{6} \\ -\frac{137}{180} \\ \frac{15}{34} \end{bmatrix}.$$

$C_6 \neq 0$, therefore, the 3VS-HOBBDF is of order three with error constant $\left[\frac{1}{30}, \frac{13}{180}, \frac{1}{50}, -\frac{1}{6}, -\frac{137}{180}, -\frac{15}{34} \right]^T$.

• 4VS-HOBBDF

$$C_0 = C_1 = C_2 = C_3 = C_4 = C_5 = C_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C_7 = \begin{bmatrix} \frac{1}{42} \\ \frac{11}{180} \\ \frac{1}{120} \\ -\frac{1}{7} \\ -\frac{7}{10} \\ \frac{232}{735} \end{bmatrix}.$$

Since $C_7 \neq 0$, the 4VS-HOBBDF is of order four with error constant $\left[\frac{1}{42}, \frac{11}{180}, \frac{1}{120}, -\frac{1}{7}, -\frac{7}{10}, \frac{232}{735} \right]^T$.

5 Technique of Varying the Step Size and Order

The variable step size and variable order approaches are implemented in a single code in the VSVO-HOBBDF method. The order is varied in order to achieve the best solutions with good accuracy while the step size is varied in order to optimize the computational effort.

The technique for varying the step size and order is as follows:

- Step 1: Start the integration with the 2VS-HOBBDF method.
- Step 2: Check the local truncation error (LTE)
 - if $LTE < \text{tolerance (TOL)}$
 - Accept the solutions. Go to Step 4.
 - else
 - Reject the solutions. Go to Step 3.
- Step 3: The step size is halving ($r = 2$).
 - Repeat the computation for the current block. Go to Step 2.
- Step 4: Check the suitable value for the new step size.
 - The new step size is either maintained ($r = 1$) or increased to a factor of 1.9 ($r = 10/19$).
- Step 5: Proceed with the next block integration step.
 - The order is maintained for two consecutive blocks and is considered to increase the order after two successful steps.
- Step 6: Repeat the Steps 2–5 until the end of the interval.

Note that if two failed steps occurred, the order is decreased.

6 Numerical Experiments

The aim of conducting the numerical experiments was to analyse the performance of the VSVO-HOBBDF method. Three tested problems from the third order ODE were considered for the purpose of testing the proposed method. The performance of the method was tested at different value of tolerances. Note that ode15s is a variable step variable order method based on the numerical differentiation formulas (NDFs) of orders 1 to 5 or backward differentiation formulas (BDFs) and ode23s is based on Modified Rosenbrock formula of order 2.

Problem 1 (Boundary layer flow)

$$2y''' + yy'' = 0, \quad x \in [0, 1], \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 1.$$

Exact solutions at $x = 1$ from MATLAB:

ode15s: 0.495900477127

ode23s: 0.495892222080

Problem 2 (Thin film flow)

$$y''' = y^{-2}, \quad x \in [0, 1], \quad y(0) = 1, \quad y'(0) = 1, \quad y''(0) = 1.$$

Exact solutions $x = 1$ from MATLAB:

ode15s: 2.608276239679

ode23s: 2.608268411808

Problem 3 (Linear system)

$$y_1''' = \frac{1}{68} (817y_1 + 1393y_2 + 448y_3),$$

$$y_2''' = -\frac{1}{68} (1141y_1 + 2837y_2 + 896y_3),$$

$$y_3''' = \frac{1}{136} (3059y_1 + 4319y_2 + 1592y_3),$$

$$x \in [0, 2], \quad y_1(0) = 2, \quad y_1'(0) = -12, \quad y_1''(0) = 20,$$

$$y_2(0) = -2, \quad y_2'(0) = 28, \quad y_2''(0) = -52, \quad y_3(0) = -12, \quad y_3'(0) = -33, \quad y_3''(0) = 5.$$

$$\text{Exact solutions: } y_1(x) = e^x - 2e^{2x} + 3e^{-3x},$$

$$y_2(x) = 3e^x + 2e^{2x} - 7e^{-3x},$$

$$y_3(x) = -11e^x - 5e^{2x} + 4e^{-3x},$$

7 Numerical Results and Discussion

The numerical results were analysed at different value of tolerances, 10^{-2} , 10^{-3} , 10^{-4} , and 10^{-5} . The results obtained from the numerical experiments for all the tested problems are presented in [Tabs. 1–3](#), followed by the analysis of each method's performance.

On another note, the exact solutions at $x = 1$ for Problems 1 and 2 are determined by using $\text{TOL} = 1e - 14$ from `ode15s` and `ode23s` in MATLAB since these problems have no exact solutions. Thus, each problem will have two exact solutions.

The error for each integration step is evaluated by letting Er as the error. The average error and maximum error are defined as follows:

$$(Er_i)_n = \left(\left| \frac{(y_i)_n - (y(x_i))_n}{A + B(y(x_i))_n} \right| \right)$$

where n is the number of equations, y_i is the approximate solutions and $y(x_i)$ are the exact solutions. Mixed error test, $A = 1$, $B = 1$ is applied for Problem 3 and $A = 1$, $B = 0$, which correspond to the absolute error tests are applied for Problems 1 and 2. For Problems 1 and 2, the Er are calculated at the endpoint, i.e., $x = 1$.

Meanwhile, the average error (AR) and maximum error (MR) are defined as

$$AR = \frac{\sum_{i=1}^{NTS} \sum_{i=1}^n (Er_i)_n}{n(NTS)},$$

$$MR = \max_{1 < i < NTS} \left(\max_{1 < i < n} (Er_i)_n \right)$$

where NTS is the number of total steps.

Table 1: Numerical results for Problem 1

| TOL | Method | Solution | Er |
|-----------|-------------|----------------|--|
| 10^{-2} | ode15s | 0.495551606970 | $3.488701574990e-4$ |
| | ode23s | 0.495811512520 | $8.070956022499e-5$ |
| | VSVO-HOBBDF | 0.495899660257 | $8.168701410294e-7$ (Exact from ode15s) $7.438176927999e-6$ (Exact from ode23s) |
| 10^{-3} | ode15s | 0.495703246865 | $1.972302623010e-4$ |
| | ode23s | 0.495809725565 | $8.249651550202e-5$ |
| | VSVO-HOBBDF | 0.495899604467 | $8.726601410380e-7$ (Exact from ode15s) $7.382386927990e-6$ (Exact from ode23s) |
| 10^{-4} | ode15s | 0.495752998053 | $1.474790740520e-4$ |
| | ode23s | 0.495810478550 | $8.174352955598e-5$ |
| | VSVO-HOBBDF | 0.495899835629 | $6.414981410519e-7$ (Exact from ode15s) $7.613548927976e-6$ (Exact from ode23s) |
| 10^{-5} | ode15s | 0.495899816531 | $6.605957580397e-7$ |
| | ode23s | 0.495864503015 | $2.771906537302e-5$ |
| | VSVO-HOBBDF | 0.495900059174 | $4.179531410409e-7$ (Exact from ode15s) $7.837093927987e-6$ (Exact from ode23s) |

Table 2: Numerical results for Problem 2

| TOL | Method | Solution | Er |
|-----------|-------------|----------------|--|
| 10^{-2} | ode15s | 2.616045399016 | $7.769159337219e-3$ |
| | ode23s | 2.607847695913 | $4.207158949261e-4$ |
| | VSVO-HOBBDF | 2.609396055175 | $1.119815495873e-3$ (Exact from ode15s) $1.127643367183e-3$ (Exact from ode23s) |
| 10^{-3} | ode15s | 2.609472832095 | $1.196592415831e-3$ |
| | ode23s | 2.607901352978 | $3.670588298119e-4$ |
| | VSVO-HOBBDF | 2.608403168545 | $1.269288658734e-4$ (Exact from ode15s) $1.347567371832e-4$ (Exact from ode23s) |
| 10^{-4} | ode15s | 2.608393806585 | $1.175669058133e-4$ |
| | ode23s | 2.608140422730 | $1.279890779240e-4$ |
| | VSVO-HOBBDF | 2.608287358662 | $1.111898287309e-5$ (Exact from ode15s) $1.894685418291e-5$ (Exact from ode23s) |
| 10^{-5} | ode15s | 2.608289758829 | $1.351914940928e-5$ |
| | ode23s | 2.608244192536 | $2.421927217311e-5$ |
| | VSVO-HOBBDF | 2.608275932195 | $3.074841266049e-7$ (Exact from ode15s) $7.520387183213e-6$ (Exact from ode23s) |

The numerical results for all the considered problems are presented in [Tabs. 1–3](#). The efficiency of the VSVO-HOBBDF method was analysed at four tolerance values and compared with two MATLAB's solvers, ode15s and ode23s. As clearly shown, most of the approximated results produced by VSVO-HOBBDF method for Problems 1 and 2 are more accurate, except for

Problem 2 at tolerance 10^{-2} at which ode23s gave better accuracy. The errors produced between the VSVO-HOBBDF method and the exact solutions from ode15s are smaller compared to the errors produced between the VSVO-HOBBDF method and the exact solutions from ode23s. This indicates the VSVO-HOBBDF method is converged to the exact solutions of the ode15s for both problems.

Table 3: Numerical results for Problem 3

| TOL | Method | MR | AR | NTS | Time |
|-----------|-------------|----------------|----------------|-----|----------|
| 10^{-2} | ode15s | 1.7788779e+001 | 7.6708193e-001 | 19 | 0.019257 |
| | ode23s | 2.0368867e+001 | 1.2915294+000 | 14 | 0.032387 |
| | VSVO-HOBBDF | 1.4812670e-004 | 3.2579805e-005 | 23 | 0.002084 |
| 10^{-3} | ode15s | 3.8004342e+000 | 1.4508834e-001 | 30 | 0.030312 |
| | ode23s | 6.1997178e+000 | 3.2981645e-001 | 29 | 0.131393 |
| | VSVO-HOBBDF | 2.0976738e-005 | 5.0364134e-006 | 34 | 0.002056 |
| 10^{-4} | ode15s | 2.1777362e-001 | 6.9996357e-003 | 41 | 0.034609 |
| | ode23s | 1.3446495e+000 | 5.4354494e-002 | 65 | 0.164096 |
| | VSVO-HOBBDF | 3.1693461e-006 | 9.7643404e-007 | 50 | 0.003303 |
| 10^{-5} | ode15s | 1.0817203e-003 | 3.9550488e-002 | 63 | 0.047459 |
| | ode23s | 2.9226260e-001 | 1.1217509e-002 | 138 | 0.299325 |
| | VSVO-HOBBDF | 6.6178473e-007 | 2.2209951e-007 | 74 | 0.004518 |

Meanwhile, the approximate solutions of Problem 3 at all tolerance values improved in accuracy when the VSVO-HOBBDF method was applied. From our observation, errors produced by the VSVO-HOBBDF method were smaller as compared to the errors produced by both MATLAB's codes at all tolerance values. The VSVO-BBDF method also outperformed the ode23s at tolerance values of 10^{-4} and 10^{-5} with less number of total steps. As we applied the VSVO-BBDF method to the tested problems, the time taken was also reduced at each tolerance as compared to the MATLAB's solver.

The results obtained thus proved that the VSVO-BBDF method is effective in solving the considered problems by providing better approximated solutions and reducing the computational cost.

8 Conclusions

This paper delineates an approach by varying the step size and order to the VSVO-HOBBDF method in a single code. The numerical experiments were applied to a number of third-order ODE problems. The performance of VSVO-HOBBDF method was evaluated based on the results. By adapting the variable step size and variable order in the same code, we have proven the efficiency of VSVO-HOBBDF method. The results also suggest that the accuracy is improved and the computational cost is also reduced. Compared to both MATLAB's codes, VSVO-HOBBDF method performs better and is a reliable direct solver for third order ODEs. Therefore, VSVO-HOBBDF method is recommended as an alternative solver for third-order ODEs.

Funding Statement: This research was funded by Fundamental Research Grant Scheme Universiti Sains Malaysia, Grant No. 203/PJJAUH/6711688 received by S. A. M. Yatim. Url at <http://www.research.usm.my/default.asp?tag=3&f=1&k=1>.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. D. Lambert, Stiffness: Linear stability theory in: *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. New York, USA: John Wiley & Sons, pp. 216–217, 1991.
- [2] Z. B. Ibrahim, K. I. Othman and M. Suleiman, “Implicit r-point block backward differentiation formula for solving first-order stiff ODEs,” *Applied Mathematics & Computation*, vol. 186, no. 1, pp. 558–565, 2007.
- [3] N. A. A. M. Nasir, Z. B. Ibrahim and M. B. Suleiman, “Fifth order two-point block backward differentiation formulas for solving ordinary differential equations,” *Applied Mathematical Sciences*, vol. 5, no. 71, pp. 3505–3518, 2011.
- [4] N. A. A. M. Nasir, Z. B. Ibrahim, K. I. Othman and M. Suleiman, “Numerical solution of first order stiff ordinary differential equations using fifth order block backward differentiation formulas,” *Sains Malaysiana*, vol. 41, no. 4, pp. 489–492, 2012.
- [5] Z. B. Ibrahim, K. I. Othman and M. Suleiman, “2-point block predictor-corrector of backward differentiation formulas for solving second order ordinary differential equations directly,” *Chiang Mai Journal of Science*, vol. 39, no. 3, pp. 502–510, 2012.
- [6] S. A. M. Yatim, Z. B. Ibrahim, K. I. Othman and M. B. Suleiman, “A numerical algorithm for solving stiff ordinary differential equations,” *Mathematical Problems in Engineering*, vol. 2013, 2013.
- [7] S. A. M. Yatim, “Variable step variable order block backward differentiation formulae for solving stiff ordinary differential equations,” Ph.D. dissertation, Universiti Putra Malaysia, Malaysia, 2013.
- [8] N. Zainuddin, Z. B. Ibrahim, K. I. Othman and M. B. Suleiman, “Direct fifth order block backward differentiation formulas for solving second order ordinary differential equations,” *Chiang Mai Journal of Science*, vol. 43, no. 5, pp. 1171–1181, 2016.
- [9] Z. B. Ibrahim, N. Zainuddin, K. I. Othman, M. Suleiman and I. S. M. Zawawi, “Variable order block method for solving second order ordinary differential equations,” *Sains Malaysiana*, vol. 48, no. 8, pp. 1761–1769, 2019.
- [10] K. A. Hussain, F. Ismail, N. Senu and F. Rabiei, “Fourth-order improved Runge-Kutta method for directly solving special third-order ordinary differential equations,” *Iranian Journal of Science & Technology, Transactions A: Science*, vol. 41, no. 2, pp. 429–437, 2017.
- [11] Y. D. Jikantoro, F. Ismail, N. Senu and Z. B. Ibrahim, “Hybrid methods for direct integration of special third order ordinary differential equations,” *Applied Mathematics & Computation*, vol. 320, pp. 452–463, 2018.
- [12] O. Adeyeye and Z. Omar, “Solving third order ordinary differential equations using one-step block method with four equidistant generalized hybrid points,” *IAENG International Journal of Applied Mathematics*, vol. 49, no. 2, pp. 253–261, 2019.
- [13] K. C. Lee, N. Senu, A. Ahmadian, S. N. I. Ibrahim and D. Baleanu, “Numerical study of third-order ordinary differential equations using a new class of two derivative Runge-Kutta type methods,” *Alexandria Engineering Journal*, vol. 59, no. 4, pp. 2449–2467, 2020.