Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 565137, 10 pages
http://dx.doi.org/10.1155/2014/565137

Research Article

Solving Nonstiff Higher Order Odes Using Variable Order Step Size Backward Difference Directly

Ahmad Fadly Nurullah Rasedee,[1] Mohamed bin Suleiman,[1] and Zarina Bibi Ibrahim[1,2]

[1] *Institute for Mathematical Research, UPM, Selangor Darul Ehsan, 43400 Serdang, Malaysia*
[2] *Department of Mathematics, Faculty of Science, UPM, Selangor Darul Ehsan, 43400 Serdang, Malaysia*

Correspondence should be addressed to Ahmad Fadly Nurullah Rasedee; ahmadfadlynurullah@yahoo.com

Received 11 March 2014; Revised 3 July 2014; Accepted 22 July 2014; Published 19 August 2014

Academic Editor: Alessandro Palmeri

The current numerical techniques for solving a system of higher order ordinary differential equations (ODEs) directly calculate the integration coefficients at every step. Here, we propose a method to solve higher order ODEs directly by calculating the integration coefficients only once at the beginning of the integration and if required once more at the end. The formulae will be derived in terms of backward difference in a constant step size formulation. The method developed will be validated by solving some higher order ODEs directly using variable order step size. To simplify the evaluations of the integration coefficients, we find the relationship between various orders. The results presented confirmed our hypothesis.

1. Introduction

In this paper, we will focus only on nonstiff ODEs of the form

$$y^{(d)} = f\left(x, \tilde{Y}\right) \tag{1}$$

given

$$\begin{aligned} \tilde{Y} &= \left(y, y', y'', \ldots, y^{(d-1)}\right), \\ \tilde{\eta} &= \left(\eta, \eta', \eta'', \ldots, \eta^{(d-1)}\right), \end{aligned} \tag{2}$$

where $\tilde{Y}(a) = \tilde{\eta}$ in the interval $a \le x \le b$, d is the order of the ODE. We impose the condition that $\tilde{Y}$ is continuous. This implies that solution for $\tilde{Y}$ exists.

Many science and engineering problems are in the form of higher order initial value problems (IVPs) ODEs. Authors such as Gear [1], Suleiman [2, 3], Hall and Suleiman [4], and Omar [5] suggested a new approach for solving higher order ODEs directly. An algorithm was designed by Suleiman [2] to solve stiff and nonstiff higher order ODEs directly without reducing the order of the problems to first order. He called it the direct integration method. The drawbacks to methods described by Omar [5], Suleiman [3], Gear [1], and Lambert [6] are the tedious calculations of the divided differences and recurrence relations in computing the integration coefficients.

Here, a new and efficient algorithm for solving IVPs of higher order ODEs directly using variable order step size method in its backward difference formulation (MSBD method) is developed. In contrast to the integration coefficients used in Suleiman's [3] direct integration (DI) method, the code MSBD calculates the integration coefficients once at the start and once when calculating the last point. The simpler nature of the backward difference compared to the divided difference gives an added advantage to the MSBD over the DI method when formulating and coding the method. This produces more elegant error formulae. The DI method in Suleiman [2] has been proven to converge. We note that the DI method is formulated based on divided difference while the proposed MSBD technique is based on backward difference. Therefore, in a similar way, we can prove the convergence of the MSBD. A brief explanation of the DI code is given below.

Let $P_{k,n}(x)$ be the integrating polynomial with degree $k-1$; interpolating k points is denoted by

$$P_{k,n}(x) = f_n + (x - x_n) f_{[n,n-1]} + \cdots + (x - x_n) \cdots (x - x_{n-k+2}) f_{[n,n-1,\ldots,n-k+1]}, \tag{3}$$

and obtained through divided difference.

We define $g_{i,t}$, $t > 0$ to be the t-fold integral and the integration coefficients obtained are denoted as follows:

$$g_{i,t} = (x_{n+1} - x_{n-i+1}) g_{i-1,t} - t g_{i-1,t+1}. \tag{4}$$

For the case $i = 0$,

$$g_{0,t} = \frac{h_{n+1}^t}{t!}, \tag{5}$$

where $h_r = x_r - x_{r-1}$, $r = 1, 2, 3, \ldots$.

Let e be given explicitly by

$$e = f(x_{n+1}, \underline{P}_{n+1}) - P_{n+1}^{(d)}, \qquad P_{n+1}^{(d-t)}, \quad t = 1, 2, \ldots, d, \tag{6}$$

which are the predicted derivatives.

Using the $g_{i,t}$ coefficients, the predictor and corrector are given by

$$P_{n+1}^{(r)} = \sum_{i=0}^{d-1-r} \frac{h^i}{i!} y_n^{(r+i)} + \sum_{i=0}^{k-1} g_{i,d-r} f_{[n,n-1,\ldots,n-i]}$$
$$y_{n+1}^{(d-t)} = P_{n+1}^{(d-t)} + \frac{g_{k,t}}{g_{k,0}} e, \quad t = 1, 2, \ldots, d. \tag{7}$$

2. Derivation of dth Order Explicit and Implicit Backward Differences Method

In this section, the integration coefficients of the backward difference formulae are derived in order to obtain a relationship between the explicit and implicit formulae. This will make the computation easier.

Integrating (1) once yields

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y, y', y'')\, dx. \tag{8}$$

Let $P_n(x)$ be the interpolating polynomial which interpolates the k values $(x_n, f_n), (x_{n-1}, f_{n-1}), \ldots, (x_{n-k+1}, f_{n-k+1})$; then

$$P_n(x) = \sum_{i=0}^{k-1} (-1)^i \binom{-s}{i} \nabla^i f_n. \tag{9}$$

Next, approximating f in (8) with $P_n(x)$ and letting

$$x = x_n + sh \tag{10}$$

give us

$$y(x_{n+1}) = y(x_n) + \int_0^1 \sum_{i=0}^{k-1} (-1)^i \binom{-s}{i} \nabla^i f_n\, ds, \tag{11}$$

where

$$y_{1,i} = (-1)^i \int_0^1 \binom{-s}{i} ds. \tag{12}$$

Let the generating function $G_1(t)$ for the coefficients $\gamma_{1,i}$ be defined as follows:

$$G_1(t) = \sum_{i=0}^{\infty} \gamma_{1,i} t^i. \tag{13}$$

Substituting (12) in $G_1(t)$ gives

$$G_1(t) = \int_0^1 e^{-s \log(1-t)} ds \tag{14}$$

which leads to

$$G_1(t) = -\left[\frac{(1-t)^{-1}}{\log(1-t)} - \frac{1}{\log(1-t)} \right]. \tag{15}$$

Hence, the coefficients of the backward difference formulation, $\gamma_{1,k}$, are given by

$$\sum_{i=0}^{k} \left(\frac{\gamma_{1,i}}{k-i+1} \right) = 1.$$
$$\gamma_{1,k} = 1 - \sum_{i=0}^{k-1} \left(\frac{\gamma_{1,i}}{k-i+1} \right), \quad k = 1, 2, \ldots, \ \gamma_{1,0} = 1. \tag{16}$$

To derive the dth order generating function, we integrate (1) d number of times and mathematical induction gives

$$G_{(d)}(t) = \frac{1}{(d-1)!} \left[\frac{1^{(d-1)}}{\log(1-t)} - \frac{(d-1)! G_{(d-1)}(t)}{\log(1-t)} \right]. \tag{17}$$

This gives the generalized explicit coefficients which are denoted by

$$\gamma_{(d),0} = \gamma_{(d-1),1}$$
$$\gamma_{(d),k} = \gamma_{(d-1),k+1} - \sum_{i=0}^{k-1} \frac{\gamma_{(d),i}}{k-i+1} \quad k = 0, 1, 2 \ldots. \tag{18}$$

Similarly, deriving the generalized implicit formulae gives the following relationship for the generating functions:

$$G_{(d)}^*(t) = \frac{1}{(d-1)!} \left[\frac{(1-t)}{\log(1-t)} - \frac{(d-1)! G_{(d-1)}^*(t)}{\log(1-t)} \right], \tag{19}$$

where the coefficients are given by

$$\gamma_{(d),0}^* = \sum_{i=0}^{1} \gamma_{(d-1),i}^*$$
$$\gamma_{(d),k}^* = \sum_{i=0}^{k+1} \gamma_{(d-1),i}^* - \sum_{i=0}^{k-1} \gamma_{(d),i}^* I_{1,k+1-i}, \quad k = 1, 2, 3 \ldots, \tag{20}$$

and $I_{1,k+1-i}$ is the Lagrange coefficient.

3. The Relationship between the Explicit and Implicit Integration Coefficients

Calculating the integration coefficients directly is time consuming if large numbers of integrations are involved. By obtaining a recursive relationship between the coefficients, we are able to obtain the implicit integration coefficient more efficiently. The relationship between the explicit and implicit coefficients is discussed below.

For first order coefficients,

$$G_1^*(t) = -\left[\frac{1}{\log(1-t)} - \frac{1-t}{\log(1-t)}\right]. \tag{21}$$

It can be written as

$$G_1^*(t) = -(1-t)\left[\frac{1}{(1-t)\log(1-t)} - \frac{1}{\log(1-t)}\right]. \tag{22}$$

By substituting

$$G_1(t) = \frac{1}{(1-t)\log(1-t)} - \frac{1}{\log(1-t)} \tag{23}$$

into (22), we have

$$G_1^*(t) = (1-t)\,G_1(t) \tag{24}$$

$$\left(\sum_{i=0}^{\infty}\gamma_{1,i}^* t^i\right) = (1-t)\left(\sum_{i=0}^{\infty}\gamma_{1,i} t^i\right). \tag{25}$$

Solving the equation above gives the recursive relationship

$$\sum_{i=0}^{k}\gamma_{1,i}^* = \gamma_{1,k}. \tag{26}$$

For second order coefficient,

$$G_2^*(t) = -\frac{1}{1!}\left[\frac{1}{\log(1-t)} - \frac{1!G_1^*(t)}{\log(1-t)}\right]. \tag{27}$$

It can be written as

$$G_2^*(t) = \frac{(1-t)}{1!}\left[\frac{1}{\log(1-t)} - \frac{1!G_1^*(t)}{(1-t)\log(1-t)}\right]. \tag{28}$$

Substituting (24) into the equation above gives

$$G_2^*(t) = \frac{(1-t)}{1!}\left[\frac{1}{\log(1-t)} - \frac{1!(1-t)G_1(t)}{(1-t)\log(1-t)}\right] \tag{29}$$

or

$$G_2^*(t) = \frac{(1-t)}{1!}\left[\frac{1}{\log(1-t)} - \frac{1!G_1(t)}{\log(1-t)}\right] \tag{30}$$

which can be simplified as

$$G_2^*(t) = (1-t)\,G_2(t),$$
$$\left(\sum_{i=0}^{\infty}\gamma_{2,i}^* t^i\right) = (1-t)\left(\sum_{i=0}^{\infty}\gamma_{2,i} t^i\right). \tag{31}$$

Solving the equation above gives the recursive relationship

$$\sum_{i=0}^{k}\gamma_{2,i}^* = \gamma_{2,k}. \tag{32}$$

For dth order coefficient, using mathematical induction, we have

$$G_{(d)}^*(t) = (1-t)\,G_{(d)}(t)$$
$$\left(\sum_{i=0}^{\infty}\gamma_{(d),i}^* t^i\right) = (1-t)\left(\sum_{i=0}^{\infty}\gamma_{(d),i} t^i\right). \tag{33}$$

In similar manner to the case of the second order coefficients, we obtain the general dth order coefficients as follows:

$$\sum_{i=0}^{k}\gamma_{(d),i}^* = \gamma_{(d),k}. \tag{34}$$

4. Order and Step Size Selection

An important decision to be made is whether to accept the result of an integration step. Although the efficiency of an algorithm is affected by the strategy for selecting the order and step size, its reliability is affected by the acceptance criteria used.

The literature on variable order step size codes show numerous strategies of order and step size selection. Varying the order in a multistep method is very easy. The order depends directly on the back values stored. After the completion of any step, the order can be increased by one if none of the back values used in the previous step are discarded. The order can also be decreased to any value desired by discarding the appropriate number of back values. Through experience, it is suggested that the order strategies which are not biased to selecting a lower order when implemented in an Adams code are efficient for nonstiff problems. After considering certain criteria, we adopted the order strategies similar to those used by Shampine and Gordon [7].

Let h be the calculated step size and h_{new} the final step size. See Suleiman [3]. In practice, we multiply h by a safety factor R such that $h_{\text{new}} = Rh$, in order to give a more conservative estimate for h_{new} and hence reduce the numbers of steps rejected. For our code, we take the safety factor as 0.8. Shampine and Gordon [7] discuss a practical approach to the result of the convergence and stability when using a variable step size. Their experience suggests that their ratio of successive step sizes will need to be restricted in order to ensure stability. For this reason and reasons of economy discussed in the section of generating the algorithm, it would appear that the use of a constant step size is desirable. Opposing this view, however, is the fact that the use of the maximum possible step size minimizes the number of steps to be taken and hence the number of derivative evaluations.

5. Error Estimation

Adopting the same strategies from Hall and Watt [8] for higher order ODEs, the estimations for the local errors on each step of the integration are shown below.

The predictor takes the form of

$$
\begin{aligned}
{}^{\mathrm{pr}}y_{n+1}^{(d)} &= \sum_{i=0}^{k-1}\gamma_{d,i}\nabla^i f_n \\
{}^{\mathrm{pr}}y_{n+1}^{(d-1)} &= y_n^{(d-1)} + h\sum_{i=0}^{k-1}\gamma_{d,i}\nabla^i f_n \\
&\vdots \\
{}^{\mathrm{pr}}y_{n+1} &= \sum_{i=0}^{d-1}\frac{h^i}{i!}y_n^i + h^d\sum_{i=0}^{k-1}\gamma_{d,i}\nabla^i f_n,
\end{aligned}
\tag{35}
$$

where the constant $\gamma_{d,i}$ is independent of k. In $P_kEC_{k+1}E$ algorithm, the corrector can be written as follows:

$$
\begin{aligned}
y_{n+1}^{(d)} &= \sum_{i=0}^{k-1}\gamma_{d,i}^*\nabla_{\mathrm{pr}}^i f_{n+1} \\
y_{n+1}^{(d-1)} &= y_n^{(d-1)} + h\sum_{i=0}^{k-1}\gamma_{d,i}^*\nabla_{\mathrm{pr}}^i f_{n+1} \\
&\vdots \\
y_{n+1} &= \sum_{i=0}^{d-1}\frac{h^i}{i!}y_n^i + h^d\sum_{i=0}^{k-1}\gamma_{d,i}^*\nabla_{\mathrm{pr}}^i f_{n+1},
\end{aligned}
\tag{36}
$$

where ∇_{pr}^i denotes the ith backward difference using $f(x_{n+1}, \widetilde{Y}_{n+1}^{\mathrm{pr}})$ for f_{n+1}. It can be shown that

$$
\gamma_{1,0}^* = \gamma_{1,0} = 1, \qquad \gamma_{d,0}^* = \gamma_{d,0}, \qquad \sum_{i=0}^{k}\gamma_{d,i}^* = \gamma_{d,k}, \tag{37}
$$

where (36), can be simplified for computation to

$$
\begin{aligned}
y_{n+1}^{(d)} &= {}^{\mathrm{pr}}y_{n+1}^{(d)} + \gamma_{0,k}\nabla_{\mathrm{pr}}^k f_{n+1} \\
y_{n+1}^{(d-1)} &= {}^{\mathrm{pr}}y_{n+1}^{(d-1)} + h\gamma_{1,k}\nabla_{\mathrm{pr}}^k f_{n+1} \\
&\vdots \\
y_{n+1} &= {}^{\mathrm{pr}}y_{n+1} + h^d\gamma_{d,k}\nabla_{\mathrm{pr}}^k f_{n+1}.
\end{aligned}
\tag{38}
$$

The Milne error estimate mentions that the local truncation error (LTE) is simply the difference between the two possibilities of k and $k-1$ giving

$$
\begin{aligned}
\widetilde{E}_k &= \gamma_{0,k}^*\nabla_{\mathrm{pr}}^k f_{n+1} \\
\widetilde{E}_k^{(1)} &= h\gamma_{1,k}^*\nabla_{\mathrm{pr}}^k f_{n+1} \\
&\vdots \\
\widetilde{E}_k^{(d)} &= h^d\gamma_{d,k}^*\nabla_{\mathrm{pr}}^k f_{n+1}.
\end{aligned}
\tag{39}
$$

The selection of an appropriate p, for $\widetilde{E}_k^{(d-p)}$, to control the order and step size is as mentioned in Suleiman [3].

When reducing the order, the estimated error would be

$$
\begin{aligned}
&E_{k-1}^{(d-p)} \\
&\quad = h^{(d-p)}\gamma_{(d-p),k-1}^* \\
&\qquad \times \left[\nabla_{\mathrm{pr}}^{k-1} f_{n+1} + f\left(x_{n+1}, \widetilde{Y}_{n+1}^{\mathrm{pr}} - h^{(d-p)}\gamma_{(d-p),k-1}\nabla^{k-1} f_n\right)\right. \\
&\qquad\quad \left. - f\left(x_{n+1}, \widetilde{Y}_{n+1}^{\mathrm{pr}}\right)\right].
\end{aligned}
\tag{40}
$$

Since a different predictor, of order one less, would have been used, from the mean value theorem, we are able to write

$$
\begin{aligned}
E_{k-1}^{(d-p)} &= h^{(d-p)}\gamma_{(d-p),k-1}^* \\
&\quad \times \left[\nabla_{pr}^{k-1} f_{n+1} - h^{(d-p)}\gamma_{(d-p),k-1}\nabla^{k-1} f_n \cdot \frac{\partial f}{\partial y}\right].
\end{aligned}
\tag{41}
$$

In practice, to avoid the extra derivative evaluation required for (40), the decision to reduce the order is based on the estimate

$$
\widetilde{E}_{k-1}^{(d-p)} = h^{(d-p)}\gamma_{(d-p),k-1}^*\nabla_{\mathrm{pr}}^{k-1} f_{n+1}, \tag{42}
$$

which is immediately available.

To consider the increase in order, the estimated error is

$$
\begin{aligned}
&E_{k+1}^{(d-p)} \\
&= h^{(d-p)}\gamma_{(d-p),k+1}^*\nabla_{\mathrm{pr}}^{k+1} f\left(x_{n+1}, \widetilde{Y}_{n+1}^{\mathrm{pr}} + h^{(d-p)}\gamma_{(d-p),k}\nabla^k f_n\right),
\end{aligned}
\tag{43}
$$

where the predicted difference is based on a predictor with an extra term included. In estimating $E_{k+1}^{(d-p)}$, we obtain

$$
\begin{aligned}
E_{k+1}^{(d-p)} &= h^{(d-p)}\gamma_{(d-p),k+1}^* \\
&\quad \times \left[\nabla_{\mathrm{pr}}^{k+1} f_{n+1} + f\left(x_{n+1}, \widetilde{Y}_{n+1}^{\mathrm{pr}} + h^{(d-p)}\gamma_{(d-p),k}\nabla^k f_n\right)\right. \\
&\quad\quad \left. - f\left(x_{n+1}, \widetilde{Y}_{n+1}^{\mathrm{pr}} + h^{(d-p)}\gamma_{(d-p),k}\nabla^k f_{n+1}\right)\right]
\end{aligned}
\tag{44}
$$

leading to

$$
\begin{aligned}
E_{k+1}^{(d-p)} &= h^{(d-p)}\gamma_{(d-p),k+1}^* \\
&\quad \times \left[\nabla_{\mathrm{pr}}^{k+1} f_{n+1} - h^{(d-p)}\gamma_{(d-p),k}\nabla^{k+1} f_{n+1} \cdot \frac{\partial f}{\partial y}\right],
\end{aligned}
\tag{45}
$$

which establishes the asymptotic validity of using

$$
\widetilde{E}_{k+1}^{(d-p)} = h^{(d-p)}\gamma_{(d-p),k+1}^*\nabla^{k+1} f_{n+1}. \tag{46}
$$

Table 1: $r = 0.8$ explicit integration coefficients.

k	0	1	2	3	4	5	6
$\gamma_{1,k}$	$\frac{4}{5}$	$\frac{8}{25}$	$\frac{92}{375}$	$\frac{392}{1875}$	$\frac{26234}{140625}$	$\frac{13344}{78125}$	$\frac{11736472}{73828125}$
$\gamma_{2,k}$	$\frac{8}{25}$	$\frac{32}{375}$	$\frac{112}{1875}$	$\frac{6784}{140625}$	$\frac{9712}{234375}$	$\frac{906544}{24609375}$	$\frac{12355736}{369140625}$
$\gamma_{3,k}$	$\frac{32}{375}$	$\frac{32}{1875}$	$\frac{176}{15625}$	$\frac{6176}{703125}$	$\frac{181064}{24609375}$	$\frac{112736}{17578125}$	$\frac{31750448}{5537109375}$
$\gamma_{4,k}$	$\frac{32}{1875}$	$\frac{128}{46875}$	$\frac{1216}{703125}$	$\frac{32384}{24609375}$	$\frac{399776}{369140625}$	$\frac{5155328}{5537109375}$	$\frac{113924096}{138427734375}$
$\gamma_{5,k}$	$\frac{128}{46875}$	$\frac{256}{703125}$	$\frac{5504}{24609375}$	$\frac{61696}{369140625}$	$\frac{751808}{5537109375}$	$\frac{2284928}{19775390625}$	$\frac{2313234752}{22840576171875}$

Table 2: $r = 0.8$ implicit integration coefficients.

k	0	1	2	3	4	5	6
$\gamma^*_{1,k}$	$\frac{4}{5}$	$\frac{-8}{25}$	$\frac{-28}{375}$	$\frac{24}{625}$	$\frac{-3466}{140625}$	$\frac{-12416}{703125}$	$\frac{-997928}{73828125}$
$\gamma^*_{2,k}$	$\frac{8}{25}$	$\frac{-64}{375}$	$\frac{-64}{1875}$	$\frac{-2336}{140625}$	$\frac{-1448}{140625}$	$\frac{-176944}{24609375}$	$\frac{-1990648}{369140625}$
$\gamma^*_{3,k}$	$\frac{32}{375}$	$\frac{-32}{625}$	$\frac{-144}{15625}$	$\frac{-608}{140625}$	$\frac{-1432}{546875}$	$\frac{-221152}{123046875}$	$\frac{-7362256}{5537109375}$
$\gamma^*_{4,k}$	$\frac{32}{1875}$	$\frac{-512}{46875}$	$\frac{-256}{140625}$	$\frac{-4096}{4921875}$	$\frac{-36608}{73828125}$	$\frac{-265984}{791015625}$	$\frac{-34103936}{138427734375}$
$\gamma^*_{5,k}$	$\frac{128}{46875}$	$\frac{-256}{140625}$	$\frac{-1408}{4921875}$	$\frac{-9472}{73828125}$	$\frac{-83648}{1107421875}$	$\frac{-7028096}{138427734375}$	$\frac{-168729536}{4568115234375}$

6. Changing the Step Size

We implement the step size changing technique as mentioned in Lambert [6]. Implementing the Adams-Bashforth methods as predictors and Adams-Moulton methods as correctors or which is commonly known as ABM methods in PECE mode in backwards difference form, we then adopt the algorithm for doubling or halving the step size from Krogh [9] which was derived for the ABM method.

7. Last Step for Solving Higher Order ODEs

As we mentioned previously, the method we proposed calculates the integrating coefficients only once. In the case of the last step, generally, the final step size will not always fit into our current step size strategy of $h, 2h$, and $(1/2)h$. Here, we need to calculate the coefficients once more for the last step when the step size is in the form of rh, $r > 0$. The following are the generating functions for the explicit coefficients:

$$G_{(d)}(t) = \frac{1}{(d-1)!}\left[\frac{(r)^{(d-1)}}{\log(1-t)} - \frac{(d-1)!G_{(d-1)}(t)}{\log(1-t)}\right] \quad (47)$$

and implicit coefficients:

$$G^*_{(d)} = \frac{1}{(d-1)!}\left[\frac{(1-t)^r(r)^{(d-1)}}{\log(1-t)} - \frac{(d-1)!G^*_{(d-1)}(t)}{\log(1-t)}\right]. \quad (48)$$

The following yields the relationship between the explicit and implicit coefficients:

$$G^*_{(d)}(t) = (1-t)^r G_{(d)}(t) \quad (49)$$

$$\gamma^*_{(d),k} = \gamma_{(d),k} + \sum_{i=0}^{k-1}\left(\frac{\gamma_{(d),k-(1+i)}(-1)^{i+1}}{(i+1)!}\prod_{j=0}^{i}(-j+r)\right). \quad (50)$$

Tables 1 and 2 contain values for the coefficients when $r = 8/10$.

8. The MSBD Algorithm

The previous write-up was devoted to the calculation of the integration coefficients for the explicit formulae which can form the basis of the predicted values and also the implicit coefficients which are obtained from the explicit formulae.

For the sake of clarity, we present the algorithm for the MSBD.

Step 1. The integration coefficients are calculated from the algorithm in (16), (18), and (34).

Step 2. Use the k back values to obtain the predictor in (30).

Step 3. Calculate the corrected values in terms of the predicted values given in (35).

Table 3: Comparison between the 1PBDVSO, DI, B1, and D1 methods for solving Problem 1.

TOL	MTD	STEPS	FS	MAXE	AVER	TIME (total)
10^{-2}	D1	113	5	8.80309 (−2)	1.19400 (−1)	2078
	B1	89	3	2.07488 (−1)	6.87734 (−2)	1628
	DI	76	1	8.78700 (−2)	2.56159 (−2)	969
	1PBDVSO	71	0	1.17774 (−1)	1.70700 (−2)	969
10^{-3}	D1	146	7	2.68984 (−2)	5.25240 (−2)	2560
	B1	111	0	3.05138 (−2)	6.28779 (−3)	2071
	DI	85	2	2.12907 (−2)	9.00022 (−3)	1135
	1PBDVSO	79	1	1.81903 (−2)	6.05764 (−3)	1063
10^{-4}	D1	151	2	4.30130 (−3)	5.45887 (−3)	3024
	B1	170	1	5.37539 (−4)	1.06165 (−4)	3169
	DI	94	1	4.25614 (−3)	1.53472 (−3)	1284
	1PBDVSO	149	0	1.97342 (−5)	5.05592 (−6)	1951
10^{-5}	D1	218	1	1.84329 (−5)	1.33438 (−5)	3860
	B1	200	7	4.40121 (−4)	1.37204 (−4)	3652
	DI	161	0	6.68588 (−4)	2.70629 (−4)	2094
	1PBDVSO	160	0	1.83864 (−5)	6.47215 (−6)	2080
10^{-6}	D1	275	3	1.80685 (−6)	1.81711 (−6)	5227
	B1	121	0	1.38184 (−5)	2.65555 (−6)	4077
	DI	179	1	3.15166 (−4)	1.29162 (−4)	2294
	1PBDVSO	176	0	5.88468 (−6)	2.26518 (−6)	2272
10^{-7}	D1	219	9	4.47927 (−6)	7.17250 (−6)	5413
	B1	330	0	1.06409 (−7)	6.01321 (−8)	6257
	DI	193	1	1.44784 (−4)	5.87351 (−5)	2502
	1PBDVSO	197	0	2.58941 (−7)	9.00261 (−8)	2513
10^{-8}	D1	348	2	3.49871 (−8)	2.75009 (−8)	6310
	B1	287	2	4.59227 (−7)	1.19307 (−7)	5367
	DI	205	0	6.69118 (−5)	2.76364 (−5)	8665
	1PBDVSO	209	0	6.96499 (−8)	2.35911 (−8)	2664
10^{-9}	D1	260	7	2.68246 (−8)	1.79939 (−8)	8633
	B1	424	0	4.48526 (−10)	1.44218 (−10)	8271
	DI	226	0	3.11980 (−5)	1.25997 (−5)	2944
	1PBDVSO	225	0	1.87994 (−9)	4.25589 (−10)	2851
10^{-10}	D1	526	9	3.45300 (−9)	2.61398 (−9)	9866
	B1	475	10	7.88269 (−9)	1.39352 (−9)	8822
	DI	370	0	1.44822 (−7)	5.81042 (−8)	4639
	1PBDVSO	248	0	4.13390 (−9)	8.63144 (−10)	3109

Step 4. The errors $\widetilde{E}_{k-1}^{(d-p)}$, $E_k^{(d-p)}$, and $\widetilde{E}_{k+1}^{(d-p)}$ are obtained in (39), (42), and (46).

Step 5. Determine whether E_k satisfies the local accuracy requirements which we take to be $\theta_{n+1}^{\mathrm{pr}}|E_k^{(d-p)}| < \mathrm{TOL}$, where $\theta_{n+1}^{\mathrm{pr}} = 1/(A + B + P_n)$ with $A = 1$, $B = 0$ give the absolute error test; $A = 0$, $B = 1$ which gives a relative error test, while $A = B = 1$ give a mixed error test.

Step 6. The order is lowered by one if, for $k > 2$, $\max(|\widetilde{E}_{k-1}^{(d-p)}|, |\widetilde{E}_{k-2}^{(d-p)}|) \le |E_k^{(d-p)}|$ and $k = 2$, $|\widetilde{E}_{k-1}^{(d-p)}| \le 0.5\,|E_k^{(d-p)}|$. If we have $\widetilde{E}_{k+1}^{(d-p)}$ available, we lower the order if $k > 1$ and $|\widetilde{E}_{k-1}^{(d-p)}| \le \min(|E_k^{(d-p)}|, |\widetilde{E}_{k+1}^{(d-p)}|)$. The order is raised by one only after $k+1$ successful step at constant step size such that, for $k > 1$, $|\widetilde{E}_{k+1}^{(d-p)}| < |E_k^{(d-p)}| < \max(|\widetilde{E}_{k-1}^{(d-p)}|, |\widetilde{E}_{k-2}^{(d-p)}|)$ and, for $k = 1$, $|\widetilde{E}_{k+1}^{(d-p)}| < 0.5\,|E_k^{(d-p)}|$. We restrict k in the range $1 \le k \le 12$.

Step 7. Step size selection is also determined based on the local accuracy requirements. The step size algorithm for doubling $\nabla_{(D)}^{i} I(x) := \nabla_{(D)}^{i-1} I(x) - \nabla_{(D)}^{i-1} I(x - 2h)$ and halving $\nabla_{(H)}^{i} I(x) := \nabla_{(H)}^{i-1} I(x) - \nabla_{(H)}^{i-1} I(x - h/2)$, $i = 2, 3, \ldots, k - 1$ is derived by Krogh [9].

Step 8. If $h < |x\mathrm{end} - x|$, where h is the current step size and $x$end denotes the end of the interval, repeat Steps 2–7. If not, determine r where $r = |x\mathrm{end} - x|/h$. We then calculate the new integration coefficients as given in (47), (48), and (50). Using the new coefficients, repeat Steps 2–7 and exit the program.

9. Test Problems and Numerical Results

9.1. Numerical Results. Tables 3, 4, 5, and 6 show the numerical results for Problems 1–4, when solved with direct integration, backwards difference, and also reducing to first order

Table 4: Comparison between the 1PBDVSO, DI, B1, and D1 methods for solving Problem 2.

TOL	MTD	STEPS	FS	MAXE	AVER	TIME (total)
10^{-2}	D1	135	0	4.83764 (−2)	3.21705 (−2)	6621
	B1	134	0	2.63945 (−2)	2.14665 (−2)	4358
	DI	96	0	4.99681 (−3)	2.47698 (−3)	1961
	1PBDVSO	173	0	1.01436 (−4)	8.58874 (−5)	3218
10^{-3}	D1	175	1	8.81849 (−3)	5.48179 (−3)	8294
	B1	121	0	3.90401 (−2)	1.65274 (−2)	3896
	DI	157	0	6.91170 (−5)	4.87625 (−5)	2890
	1PBDVSO	162	0	2.40484 (−3)	1.71894 (−3)	3005
10^{-4}	D1	233	2	2.85719 (−3)	1.39202 (−3)	11684
	B1	173	0	1.44401 (−3)	9.91992 (−4)	5520
	DI	141	0	1.79264 (−4)	1.19952 (−4)	2622
	1PBDVSO	142	0	2.60404 (−4)	1.89056 (−4)	2635
10^{-5}	D1	226	1	3.05429 (−4)	1.22401 (−4)	11095
	B1	263	2	1.97450 (−4)	5.02997 (−5)	8492
	DI	238	0	5.14212 (−6)	3.89969 (−6)	4137
	1PBDVSO	237	0	1.03862 (−5)	8.18350 (−6)	4268
10^{-6}	D1	337	1	1.12897 (−5)	5.53620 (−6)	16060
	B1	262	0	6.52399 (−5)	2.34931 (−5)	8429
	DI	214	0	1.18826 (−5)	9.01852 (−6)	3766
	1PBDVSO	218	0	1.56415 (−6)	1.09680 (−6)	3941
10^{-7}	D1	421	2	4.00671 (−7)	4.00671 (−7)	19735
	B1	337	0	1.70966 (−6)	1.05119 (−6)	10899
	DI	368	0	1.79221 (−7)	1.35202 (−7)	6213
	1PBDVSO	370	0	2.75078 (−7)	1.91969 (−7)	6584
10^{-8}	D1	610	4	3.05239 (−7)	2.22380 (−7)	29060
	B1	523	3	4.90321 (−8)	3.52515 (−8)	16840
	DI	335	1	3.11508 (−8)	2.58590 (−8)	5641
	1PBDVSO	336	0	3.64286 (−8)	1.21667 (−8)	5996
10^{-9}	D1	534	2	4.96195 (−8)	1.71145 (−8)	25234
	B1	555	5	9.16995 (−8)	5.35093 (−8)	17610
	DI	578	1	1.20539 (−8)	9.25523 (−9)	9347
	1PBDVSO	575	0	3.80099 (−9)	2.95718 (−9)	10114
10^{-10}	D1	800	2	2.12904 (−10)	1.21736 (−10)	38235
	B1	671	4	8.63683 (−10)	3.89440 (−10)	21617
	DI	516	0	8.37294 (−10)	6.51148 (−10)	8541
	1PBDVSO	517	0	3.30834 (−9)	2.52921 (−9)	9011

systems. Problems 1 and 2 are well-behaved problems whereas Problems 3 and 4 are without exact solutions. For the first two problems, we evaluate the maximum and average values of the error in the computed solution y. The definition of the error is as defined above. The following notation will indicate

TIME: the execution time taken in microseconds,

FS: the number of failed steps,

STEPS: total steps,

MTD: the name of the method,

SUCSTEP: the number of successful step,

MAXE: the maximum error,

AVER: the average error,

TOL: the tolerance used,

DI: direct integration,

D1: direct integration with the reduction to first order system,

1PBDVSO: backward difference,

B1: backward difference with the reduction to first order system.

The errors calculated are defined as

$$(e_i)_t = \left| \frac{(y_i)_t - (y(x_i))_t}{A + B(y(x_i))_t} \right| \tag{51}$$

with $(y)_t$ as the tth component of $\widetilde{y}$. $A = 1,\ B = 0$ correspond to the absolute error test, $A = 1,\ B = 1$ correspond to the mixed error test, and $A = 0, B = 1$ correspond to the relative

Table 5: Comparison between the 1PBDVSO, DI, B1, and D1 methods for solving Problem 3.

TOL	MTD	$y(1)$
10^{-2}	1PBDVSO	1.9788520539e + 000
	DI	1.9163232164e + 000
	B1	1.8780198255e + 000
	D1	–1.#IND000000e + 000
10^{-4}	1PBDVSO	1.8706114418e + 000
	DI	1.8694497531e + 000
	B1	1.8694030590e + 000
	D1	–1.#IND000000e + 000
10^{-6}	1PBDVSO	1.8694735377e + 000
	DI	1.8694497531e + 000
	B1	1.8694368915e + 000
	D1	–1.#IND000000e + 000
10^{-8}	1PBDVSO	1.8694389431e + 000
	DI	1.8694388843e + 000
	B1	1.8694402726e + 000
	D1	–1.#IND000000e + 000
10^{-10}	1PBDVSO	1.8694388662e + 000
	DI	1.8694388834e + 000
	B1	Fail to compute
	D1	–1.#IND000000e + 000

error test. The mixed error test is used for all test problems. The maximum and average errors are given by

$$\begin{aligned} \text{MAXE} &= \max_{1<i<\text{SUCSTEP}} \left(\max_{1<i<N} (e_i)_t \right), \\ \text{AVER} &= \frac{\sum_{i=1}^{\text{SUCSTEP}} \sum_{t=1}^{N} (e_i)_t}{(N)\,(\text{SUCSTEP})}, \end{aligned} \tag{52}$$

where N is the number of equations in the systems.

The result of the backwards difference method (1PBDVSO) is measured against D1, B1, and DI method for solving the same problems. The variable step size order codes are implemented for solving the problems. We begin to compute the solution using a higher tolerance (TOL $* 10^{-2}$).

Problem 1. Consider

$$\begin{gathered} y_1'' = -\frac{y_1}{r^3}, \qquad y_1(0) = 1, \qquad y_1'(0) = 0 \\ y_2'' = -\frac{y_2}{r^3}, \qquad y_2(0) = 0, \qquad y_2'(0) = 1 \\ r = \left(y_1^2 + y_2^2 \right)^{1/2} \\ 0 \le x \le 16\pi. \end{gathered} \tag{53}$$

Solution is as follows:

$$y_1(x) = \cos x, \qquad y_2(x) = \sin x. \tag{54}$$

First order system is as follows:

$$\begin{gathered} y_1' = y_3, \qquad y_2' = y_4, \qquad y_3' = -\frac{y_1}{r^3}, \qquad y_4' = -\frac{y_2}{r^3} \\ y_1(0) = 1, \qquad y_2(0) = 0, \qquad y_3(0) = 0, \qquad y_4(0) = 1 \end{gathered} \tag{55}$$

Solution is as follows:

$$\begin{aligned} y_1(x) &= \cos x, \qquad y_2(x) = \sin x, \\ y_3(x) &= -\sin x, \qquad y_4(x) = \cos x. \end{aligned} \tag{56}$$

For source, see Shampine and Gordon [7].

Problem 2. Consider

$$\begin{gathered} y^{(8)} = y, \qquad y'(0) = y''(0) = \cdots = y^{(7)}(0) = 1, \\ 0 \le x \le 100. \end{gathered} \tag{57}$$

Solution is as follows:

$$y = e^x. \tag{58}$$

First order system is as follows:

$$\begin{gathered} y_1' = y_2, \qquad y_2' = y_3, \qquad y_3' = y_4, \qquad y_4' = y_5, \\ y_5' = y_6, \qquad y_6' = y_7, \qquad y_7' = y_8, \qquad y_8' = y_1 \\ y_1(0) = y_2(0) = \cdots = y_8(0) = 1. \end{gathered} \tag{59}$$

Solution is as follows:

$$y_1(x) = y_2(x) = y_3(x) = \cdots = y_8(x) = e^x. \tag{60}$$

For source, see Suleiman [2].

Problem 3. Van der Pol's equation is as follows:

$$\begin{gathered} y'' = 5\left(1 - y^2\right) y' - y \\ y(0) = 2, \qquad y'(0) = 0, \\ 0 \le x \le 1. \end{gathered} \tag{61}$$

First order system is as follows:

$$\begin{gathered} y_1' = y_2, \qquad y_2' = 5\left(1 - y_1^2\right) y_2 - y_1, \\ y_1(0) = 2, \qquad y_2(0) = 0. \end{gathered} \tag{62}$$

For source, see Suleiman [2].

Table 6: Comparison between the 1PBDVSO, DI, B1, and D1 methods for solving Problem 4.

TOL	MTD	$y(1)$
10^{-4}	1PBDVSO	$-1.5171246226e-009$
	DI	$-2.5152953310e-008$
	B1	$9.9810188187e-009$
	D1	Fail to compute
10^{-6}	1PBDVSO	$1.0035458019e-008$
	DI	$1.0609214996e-008$
	B1	$9.9998786959e-009$
	D1	Fail to compute
10^{-8}	1PBDVSO	$1.0004172282e-008$
	DI	$1.0004031492e-008$
	B1	$1.0000000079e-008$
	D1	Fail to compute
10^{-10}	1PBDVSO	$9.9999571459e-009$
	DI	$9.9999855296e-009$
	B1	$9.9999999555e-009$
	D1	Fail to compute

Problem 4. Control theory is as follows:

$$\begin{aligned} y^{(iv)} = {} & \left(y^2 - \sin y - 100^4\right) y \\ & + \left(y' y'' \frac{1}{\left(y^2+1\right)} - 4 \times 100^3\right) y' \\ & + \left(1 - 6 \times 100^2\right) y'' \\ & + \left(10 e^{y'''^2} - 4 \times 100\right) y''' + 1 \\ y(0) = {} & y'(0) = y''(0) = y'''(0) = 0, \\ & 0 \le x \le 1. \end{aligned} \tag{63}$$

First order system is as follows:

$$\begin{aligned} & y_1' = y_2, \qquad y_2' = y_3, \qquad y_3' = y_4, \\ y_4' = {} & \left(y_1^2 - \sin y - 100^4\right) y_1 \\ & + \left(y_2 y_3 \frac{1}{\left(y_1^2+1\right)} - 4 \times 100^3\right) y_2 + \left(1 - 6 \times 100^2\right) y_3 \\ & + \left(10 e^{y_4^2} - 4 \times 100\right) y_4 + 1 \\ & y_1(0) = y_2(0) = y_3(0) = y_4(0) = 0. \end{aligned} \tag{64}$$

For source, see Enright et al. [10].

10. Comments on Numerical Results and Conclusion

Results for the first problem in Table 3 whose solution has the trigonometric form show that the 1PBDVSO is quicker in terms of execution times compared to the other three methods. This is not surprising because the divided difference has an element of division whereas backward difference is without one. And since the solution involves trigonometric functions, the divided difference can be small and can magnify round-off errors. This can clearly be seen as we decrease the tolerance. The solution to Problem 2 has the solution in the positive exponential form and, therefore, the effect in terms of accumulated errors between divided difference and backward difference is small. This is reflected in the graph of accuracy versus execution times in Figure 2, where 1PBDVSO and DI methods almost overlapped each other.

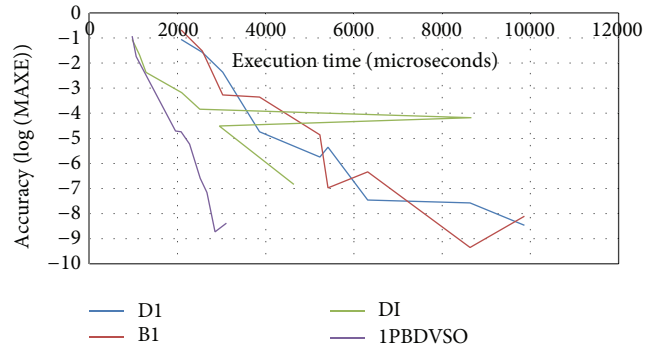

Figure 1: Comparison of efficiency between the 1PBDVSO, DI, B1, and D1 methods for Problem 1.

In Problem 3, the reduction to first order system using divided difference cannot solve the Van der Pol equation while the backward difference could. However, the DI direct method and 1PBDVSO are successful in solving this equation. This conclusion is judged by the closeness of the solution at $x = 1$. This also indicates the instability of the problem when reduced to first order and solved using divided difference. For Problem 4, which has no given solution, similar behavior pattern occurs in solving the problem. This indicates the preferability of the high order problems being solved directly rather than reduction to first order. Overall, from the tables, we see that the accuracy of 1PBDVSO is better compared to the other methods. In order to see this more clearly, we present the graphs of execution times against accuracy. To determine the more accurate relationship, we take any particular values of the abscissae (the execution times) and the ordinates (accuracy) which are lowest in values representing more accurate points. For graph in Figure 1, the 1PBDVSO is the most accurate method since, for all the abscissae chosen from the graphs, they are below the values of the corresponding ordinates.

The numerical results show that the 1PBDVSO is better than DI in terms of accuracy, competitive in terms of execution times, and more stable when tested with more demanding problems. We are inclined to choose 1PBDVSO because, for some problems, they are more accurate. It can be seen from the figures above, where the undermost curve is more efficient. Using this simple performance index, the point which is lowest at a particular time is more accurate. These results show that the variable order step size method

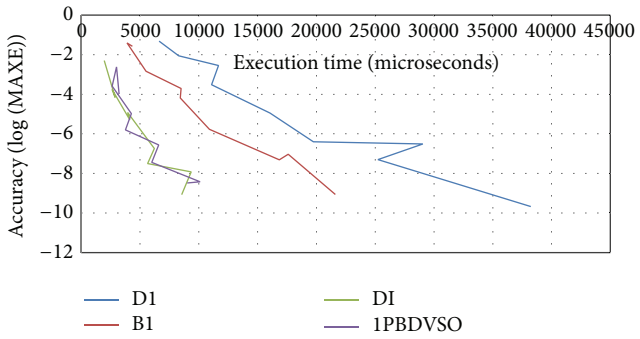

Figure 2: Comparison of efficiency between the 1PBDVSO, DI, B1, and D1 methods for Problem 2.

gives acceptable solutions and is suitable for solving higher order ODEs directly.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The researchers would like to thank the Ministry of Higher Education (MOHE) for its My Brain 15 (My PhD) scholarship.

References

[1] C. W. Gear, "The numerical integration of ordinary differential equations," *Mathematics of Computation*, vol. 21, pp. 146–156, 1967.

[2] M. B. Suleiman, *Generalised multistep Adams and Backward differentiation methods for the solution of stiff and non-stiff ordinary differential equations [Ph.D. thesis]*, University of Manchester, Manchester, UK, 1979.

[3] M. B. Suleiman, "Solving nonstiff higher order ODEs directly by the direct integration method," *Applied Mathematics and Computation*, vol. 33, no. 3, pp. 197–219, 1989.

[4] G. Hall and M. B. Suleiman, "Stability of Adams-type formulae for second-order ordinary differential equations," *IMA: Journal of Numerical Analysis*, vol. 1, no. 4, pp. 427–428, 1981.

[5] Z. B. Omar, *Parallel block methods for solving higher order ordinary differential equations directly [Ph.D. thesis]*, Universiti Putra Malaysia, 1999.

[6] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, New York, NY, USA, 1973.

[7] L. F. Shampine and M. K. Gordon, *Computed Solutions of Ordinary Differential Equations*, W. H. Freeman, San Francisco, Calif, USA, 1975.

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

[9] F. T. Krogh, "Algorithms for changing the step size," *SIAM Journal on Numerical Analysis*, vol. 10, no. 5, pp. 949–965, 1973.

[10] W. H. Enright, T. E. Hull, and B. Lindberg, "Comparing numerical methods for systems of ordinary differential equations," Tech. Rep. 69, Department of Computer Science, University of Toronto, 1974.