MATHEMATICAL MODELING AND COMPUTING, Vol. 9, No. 1, pp. 101–110 (2022)

Variable order step size method for solving orbital problems with periodic solutions

Rasedee A. F. N.[1], Jamaludin N. A.[2], Najib N.[1], Abdul Sathar M. H.[3], Wong T. J.[4], Koo L. F.[4]

[1] *Faculty of Economics and Muamalat, Universiti Sains Islam Malaysia,*
71800 Nilai, Negeri Sembilan, Malaysia
[2] *Centre for Defence Foundation Studies, Universiti Pertahanan Nasional Malaysia,*
Kem Sungai Besi, 57000, Kuala lumpur, Malaysia
[3] *The Centre of Foundation Studies for Agricultural Science, Universiti Putra Malaysia,*
43400 UPM Serdang, Selangor, Malaysia
[4] *Department of Science and Technology, Faculty of Humanities, Management and Science,*
Universiti Putra Malaysia, Bintulu Sarawak Campus, 97008 Bintulu, Sarawak, Malaysia

(Received 7 July 2021; Revised 24 November 2021; Accepted 24 November 2021)

Existing variable order step size numerical techniques for solving a system of higher-order ordinary differential equations (ODEs) requires direct calculating the integration coefficients at each step change. In this study, a variable order step size is presented for direct solving higher-order orbital equations. The proposed algorithm calculates the integration coefficients only once at the beginning and, if necessary, once at the end. The accuracy of the numerical approximation is validated with well-known orbital differential equations. To reduce computational costs, we obtain the relationship for the predictor-corrector algorithm between integration coefficients of various orders. The efficiency of the proposed method is substantiated by the graphical representation of accuracy at the total evaluation steps.

Keywords: *applied mathematics, backward difference, ODEs, multistep, variable order step size.*

2010 MSC: 34A12, 34A34, 65L06, 65L05 **DOI:** 10.23939/mmc2022.01.101

1. Introdcution

Numerous significant scientific problems can be formulated in the form of a higher-order initial value problem (IVP) of Ordinary Differential Equations (ODEs). Consider the higher-order ODE of the form

$$y^{(d)} = f(x, \widetilde{Y}). \tag{1}$$

The initial value condition is denoted by $\widetilde{Y}(\alpha) = \widetilde{\eta}$, where

$$\widetilde{Y}(t) = (y, y', \dots, y^{(d-1)}), \quad \widetilde{\varepsilon} = (\varepsilon, \varepsilon', \dots, \varepsilon^{(d-1)}), \quad \alpha \leqslant x \leqslant \beta. \tag{2}$$

For this study, we focus on non-linear ODEs in the form of orbital problems with periodic solutions. A novel technique for approximating higher order ODEs was suggested by some studies, such as [1–4]. Suleiman [3] designed an algorithm for direct solving stiff and nonstiff higher-order ODEs directly. This eliminates the need for the tedious calculations when reducing the problems to first order ODEs. This technique was referred to as the Direct Integration (DI) method, which is a multistep method based on a divided difference formulation. Although the DI algorithm was shown to be viable, it had one major drawback. The tedious calculations of the divided difference integration coefficients at

This work was supported by Universiti Pertahanan Nasional Malaysia (UPNM) under the Short Term Grant Scheme, project number UPNM/2020/GPJP/SG/5.

every step change were time-consuming. This study develops an efficient multistep method equipped with a variable order step size (VOS) algorithm in predictor-corrector form for direct solving higher-order initial value ODEs, which are formulated in backward difference form (1PVOSBD method). The 1PVOSBD code calculates the integration coefficients only at the start, contrary to the integration coefficients required by the direct integration (DI) method. In addition, the backward difference produces error formulae that are more elegant in comparison to the divided difference.

2. The explicit and implicit integration coefficients

Given the higher-order ODE (1), the derivation of the predictor–corrector backward difference method begins with obtaining the explicit integration coefficient by integrating (1) once, yielding

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f\big(y, y', \dots, y^{(d-1)}\big)\,dt. \tag{3}$$

To solve the integral in (3), first consider $P_n(t)$ as the Newton–Gregory backward difference interpolation polynomial. $P_n(t)$ interpolates $f(y, y', \dots, y^{(d-1)})$ at k back values (t_n, f_n), (t_{n-1}, f_{n-1}), …, (t_{n-k+1}, f_{n-k+1}) defined as follows

$$P_n(x) = \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n, \quad s = \frac{t - t_n}{h}.$$

This allows (3) to be approximated using $P_n(t)$. Next, by replacing dt by $h\,ds$ changes the limit of integration yielding

$$y^{(d-1)}(t_{n+1}) = y^{(d-1)}(t_n) + \int_0^1 \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n\, h\, ds.$$

The equation above can be denoted by

$$y^{(d-1)}(t_{n+1}) = y^{(d-1)}(t_n) + h \sum_{i=0}^{k-1} \delta_{1,i} \nabla^i f_n, \quad \delta_{1,i} = (-1)^i \int_0^1 \begin{pmatrix} -s \\ i \end{pmatrix} ds.$$

Let $\Gamma_1(t)$ denote the generating function for the sets of coefficients $\delta_{1,i}$ defined as follows:

$$\Gamma_1(t) = \sum_{i=0}^{\infty} \delta_{1,i} t^i.$$

Substituting $\delta_{1,i}$ in $\Gamma_1(t)$ yields:

$$\Gamma_1(t) = \int_0^1 e^{-s \log(1-t)}\, dt.$$

Solving the integral above integrating the equation above, the generating function can be conveyed as

$$\Gamma_1(t) = -\left[\frac{(1-t)^{-1}}{\log(1-t)} - \frac{1}{\log(1-t)} \right].$$

This gives the backward difference coefficients formulation, $\delta_{1,k}$ such as

$$\delta_{1,k} = 1 - \sum_{i=0}^{k-1} \left(\frac{\delta_{1,i}}{k-i+1} \right), \quad k = 1, 2, \dots, \quad \delta_{1,0} = 1. \tag{4}$$

Next, we proceed with the d-th order generating function. This is established by integrating (1) d folds. Followed by some mathematical induction, we are able to obtain the general solution of $y(t_{n+1})$ in the following form of

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \ldots + \frac{h^{(d-1)}}{(d-1)!}y^{(d-1)}(t_n) + \frac{h^d}{d!}\sum_{i=0}^{k-1}\delta_{d,i}\nabla^i f_n.$$

Similarly to the first-order generating function (Γ_1), the d-th generating function, $G_d(t)$ can be denoted as

$$\Gamma_{(d)}(t) = \frac{1}{(d-1)!}\left[\frac{1-(d-1)!\Gamma_{(d-1)}(t)}{\log(1-t)}\right]$$

with the generalized relationship between explicit coefficients of different orders

$$\delta_{(d),0} = \delta_{(d-1),1}, \quad \delta_{(d),k} = \delta_{(d-1),k+1} - \sum_{i=0}^{k-1}\left(\frac{\delta_{(d),i}}{k-i+1}\right), \quad k = 1, 2, \ldots. \tag{5}$$

Hence, the generating function formulae for the set of implicit integration coefficients is as established in an almost similar manner as its explicit counterpart and can be written as follows:

$$\Gamma^*_{(d)}(t) = \frac{1}{(d-1)!}\left[\frac{(1-t)-(d-1)!\Gamma^*_{(d-1)}(t)}{\log(1-t)}\right]$$

with coefficients as follows

$$\delta^*_{(d),0} = \sum_{i=0}^{1}\delta^*_{(d-1),i}, \quad \delta^*_{(d),k} = \sum_{i=0}^{k+1}\delta^*_{(d-1),i} - \sum_{i=0}^{k-1}\delta^*_{(d),i}L_{1,k+1-i}, \quad k = 1, 2, \ldots \tag{6}$$

given that $L_{1,k+1-i}$ is the Lagrange coefficient. If large integration are involved, calculation of the integration coefficients can be time consuming. This is avoided with a recursive relationship between the coefficients. Similar to the calculations in [5], the general relationship between the explicit dth order coefficients and implicit dth coefficients are given by

$$\sum_{i=0}^{k}\delta^*_{(d),i} = \delta_{(d),k}. \tag{7}$$

3. Determining order and step size

A key component, for developing a VOS algorithm is determining when an integration is considered as acceptable for varying the order. Even though the strategies selected for VOS has a definitive effect on the efficiency of the algorithm, the VOS acceptance criteria is determining factor which regulates its reliability. The implementation of a variable order multistep method relies solely on the back values stored. When varying the order, the number of previous values stored determines the possibility of increasing the order and by simply by eliminating a certain number of back values allows us to reduce the current order. Experience suggests that order strategies which does not favour lower orders when are deemed efficient for solving nonstiff problems. Order strategies implemented in the current study adopts similar to strategies developed in [6]. Let denote the calculated step size as h and the final step size as h_{end}. For a standard estimate of h_{end}, safety factor R is multiplied to the current step size of h such that $h_{end} = Rh$ in order to reduce steps rejected. To avoid issues regarding convergence and stability [6] suggested restrictions on the ratio of successive step size. This is to ensure stability of the method. Considering in a PECE algorithm, we adopt the doubling or halving the step size code from [1] which is implemented in a step size changing technique from [7]. The step changing algorithm applied in the current work is illustrated in Algorithm 1 and Algorithm 2 (source: [8]).

Algorithm 1 Doubling the step size algorithm.

```
Begin
  $H_{\min} := 0.8H_{\min}$
  If$(H_{\min} \geqslant 2)$
    Begin
      $H := 2H_{\text{old}}$
    End
  If$(H := 2H_{\text{old}})$
    Begin
      For $I := 1$, to $N$ step $1$
        Begin
          $\nabla^{k-1} f_{n+1} := \frac{1}{2}\nabla^{k-1} f_{n+1}$
          For $T := 1$, to $K - 2$ step $1$
            Begin
              For $M := T$, to $K - 2$ step $1$
                Begin
                  $\nabla^{m} f_{n+1} := 2\big(\nabla^{m} f_{n+1} - \nabla^{m+1} f_{n+1}\big)$
                End
              $\nabla^{k-1} f_{n+1} := 2\nabla^{k-1} f_{n+1}$
            End
          $\nabla^{k-1} f_{n+1} := 2\nabla^{k-1} f_{n+1}$
        End
    End
End
```

Algorithm 2 Halving the step size algorithm.

```
Begin
  $Errors := Errors + 1$
  $H := 0.5H_{\text{old}}$
  $X := X_{\text{old}}$
  For $I := 1$, to $N$ step $1$
    Begin
      $\nabla^{k-1} f_{n+1} := \frac{1}{2}\nabla^{k-1} f_{n+1}$
        For $T := K - 2$, to $1$ step $-1$
          Begin
            $\nabla^{k-1} f_{n+1} := \frac{1}{2}\nabla^{k-1} f_{n+1}$
            For $M := K - 2$, to $T$ step $-1$
              Begin
                  $\nabla^{m} f_{n+1} := \frac{1}{2}\big(\nabla^{m} f_{n+1} + \nabla^{m+1} f_{n+1}\big)$
              End
          End
    End
End
```

4. Error estimation

The current study refers to the works of Hall and Watt [2] to establish the estimated local error for each integration step. Estimation for the local error begins with denoting the set of predictors in following form:

$$\begin{aligned} {}^{pr}y^{(d)}_{n+1} &= \sum_{i=0}^{k-1} \delta_{(0),i}\nabla^i f_n, \\ \dots \qquad & \dots \qquad \dots, \\ {}^{pr}y_{n+1} &= \sum_{i=0}^{d-1} \frac{h}{i!} y_n^{(i)} + h^d \sum_{i=0}^{k-1} \delta_{(d),i}\nabla^i f_n, \end{aligned} \tag{8}$$

because the current work implements a $P_kEC_{k+1}E$ algorithm, we are able to denote the corrector in the form of

$$\begin{aligned} {}^{cr}y^{(d)}_{n+1} &= \sum_{i=0}^{k-1} \delta^*_{(0),i}\nabla^i_{pr} f_{n+1}, \\ \dots \qquad & \dots \qquad \dots, \\ {}^{cr}y_{n+1} &= \sum_{i=0}^{d-1} \frac{h}{i!} y_n^{(i)} + h^d \sum_{i=0}^{k-1} \delta^*_{(d),i}\nabla^i_{pr} f_{n+1}, \end{aligned} \tag{9}$$

with ∇^i_{pr} as the i-th backward difference using $f\big(x_{n+1}, \widetilde{Y}^{pr}_{n+1}\big)$ for f_{n+1}. Due to (7), the corrector can be simplified to the following formulation for computational purpose

$$\begin{aligned} {}^{cr}y^{(d)}_{n+1} &= {}^{pr}y^{(d)}_{n+1} + \delta^*_{(0),i}\nabla^i_{pr} f_{n+1}, \\ \dots \qquad & \dots \qquad \dots, \\ {}^{cr}y_{n+1} &= {}^{pr}y_{n+1} + h^d\delta^*_{(d),i}\nabla^i_{pr} f_{n+1}. \end{aligned} \tag{10}$$

The local truncation error (LTE) presented in this study follows the standard Milne error estimate, which is represented by

$$\begin{aligned} \widetilde{E}_k &= \delta^*_{(0),k}\nabla^k_{pr} f_{n+1}, \\ \dots \quad & \dots \qquad \dots, \\ \widetilde{E}^{(d)}_k &= h^d\delta^*_{(d),k}\nabla^k_{pr} f_{n+1}. \end{aligned} \tag{11}$$

To control the order and step size, an appropriate p for $\widetilde{E}^{(d-p)}_k$ as mentioned in [3] and resembling the proof from [9] the asymptotic validity can be established using

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

5. Last step coefficient

As previously mentioned, the 1PVOSBD method generally requires computing the integration coefficients only once. This assertion is true except in particular last steps. When implementing the 1PVOSBD algorithm, the current step size does not always correspond with the final step, h_{end}. For these specific cases, calculating the coefficients is required where we denote the last step size as $h_{end} = rh$, $r > 0$ (refer to [5]). This yields the coefficients with the following relationship

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

6. Numerical results

Over recent years, research focusing on the second order periodic initial and boundary value problems with oscillating and periodic solutions have been investigated by various authors (see [10–16]). The efficiency of the 1PVOSBD by will be verified by comparing numerical approximations with the DI method established by [3]. Numerical results obtained are from popular orbital problems test problems which will inherently highlights the benefit of the 1PVOSBD method. The results for Problems 1–3, are the approximated solutions obtained using direct integration and backward difference method respectively. The current work evaluates the maximum error of all computed solution. The error type chosen to estimate the solutions can be found in [17]. The following indicates abbreviations used throughout the current section:

FAIL: the number of failed steps,	1PVOSBD: backward difference,
STEPS: total steps,	MAXERR: the overall maximum error
TOL: the tolerance used,	MAXERR1: the maximum error (Equation 1),
DI: direct integration,	MAXERR2: the maximum error (Equation 2),

We proceed with the following test problems and numerical results.

Orbit Problem 1: Two-body problem (source [9]).

$$y_1''(t) = \frac{-y_1(t)}{r}, \quad y_2''(t) = \frac{-y_2(t)}{r}, \quad r = \left(y_1^2(t) + y_2^2(t)\right)^{\frac{3}{2}},$$

where $y_1|_0 = 1$, $y_1'|_0 = 0$, $y_2|_0 = 0$, $y_2'|_0 = 1$ with the analytical solution $y_1(t) = \cos t$, $y_2(t) = \sin t$.

Table 1. Approximation for Problem 1.

TOL	MTD	STEPS	FAIL	MAXERR1	MAXERR2	MAXERR
10^{-2}	DI	95	5	1.00000(0)	9.99583(−1)	1.00000(0)
10^{-2}	1PVOSBD	132	8	1.00000(0)	9.98054(−1)	1.00000(0)
10^{-4}	DI	137	2	1.11352(−2)	1.27865(−2)	1.27865(−2)
10^{-4}	1PVOSBD	136	1	4.64941(−2)	4.02291(−2)	4.64941(−2)
10^{-6}	DI	179	2	1.01987(−3)	1.55778(−3)	1.55778(−3)
10^{-6}	1PVOSBD	181	1	1.12302(−3)	1.32391(−3)	1.32391(−3)
10^{-8}	DI	210	0	1.34432(−4)	2.68910(−4)	2.68910(−4)
10^{-8}	1PVOSBD	205	0	2.00681(−5)	2.31411(−5)	2.31411(−5)
10^{-10}	DI	366	1	2.84946(−7)	5.69993(−7)	5.69993(−7)
10^{-10}	1PVOSBD	252	0	3.22339(−8)	4.64242(−8)	4.64242(−8)

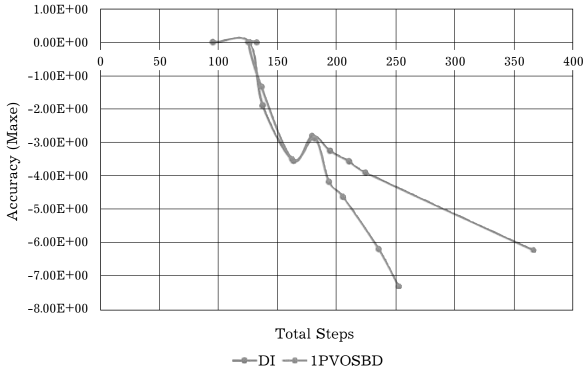

Fig. 1. Accuracy of 1PVOSBD and DI method.

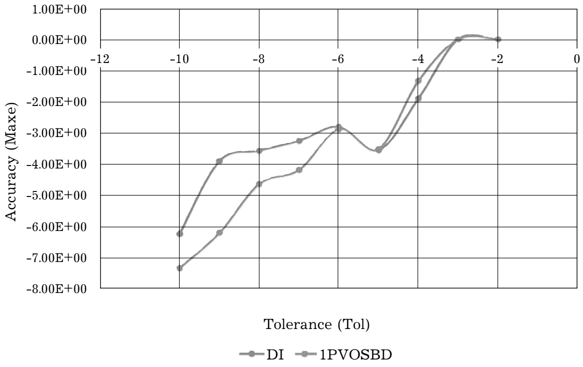

Fig. 2. Efficiency of 1PVOSBD and DI method.

Orbit Problem 2: Stiefel and Bettis (source: [18]).

$$y''(t) = -y(t) + \varepsilon\, e^{it}, \quad y(0) = 1, \quad y'(0) = 0.9995i, \quad y \in \mathbb{C}$$

can be written in the equivalent form

$$\begin{gathered} y_1''(t) = -y_1(t) + \varepsilon \cos t, \quad y_2''(t) = -y_2(t) + \varepsilon \sin t, \\ y_1|_0 = 1, \quad y_1'|_0 = 0, \quad y_2|_0 = 0, \quad y_2'|_0 = 0.9995, \end{gathered}$$

and $\varepsilon = 0.0001$ with the analytical solution

$$y_1(t) = \cos t + \frac{1}{2}\varepsilon\, t \sin t, \quad y_2(t) = \sin t + \frac{1}{2}\varepsilon\, t \cos t \quad \in \mathbb{R}.$$

Table 2. Approximation for Problems 2 and 3.

TOL	MTD	STEPS	FAIL	MAXERR	STEPS	FS	MAXERR
10^{-2}	DI	1176	1	1.06797(0)	1178	1	1.00035(0)
10^{-2}	1PVOSBD	1173	0	1.06695(0)	1173	0	1.00032(0)
10^{-4}	DI	2488	4	9.60862(−3)	1951	11	1.00054(0)
10^{-4}	1PVOSBD	2706	0	1.98095(−5)	2706	0	1.98263(−5)
10^{-6}	DI	3145	1	3.13723(−4)	3146	1	3.1712(−4)
10^{-6}	1PVOSBD	3144	0	1.65277(−6)	3150	0	1.17117(−6)
10^{-8}	DI	3650	1	6.71619(−5)	3650	1	6.72156(−5)
10^{-8}	1PVOSBD	3652	0	1.19971(−7)	3651	0	1.20103(−7)
10^{-10}	DI	6615	2	1.37306(−7)	6616	2	1.40038(−7)
10^{-10}	1PVOSBD	4242	0	2.12600(−8)	4246	0	1.92072(−8)

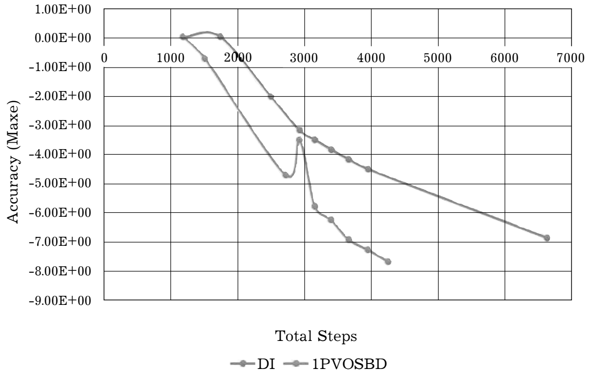

Fig. 3. Accuracy of 1PVOSBD and DI method.

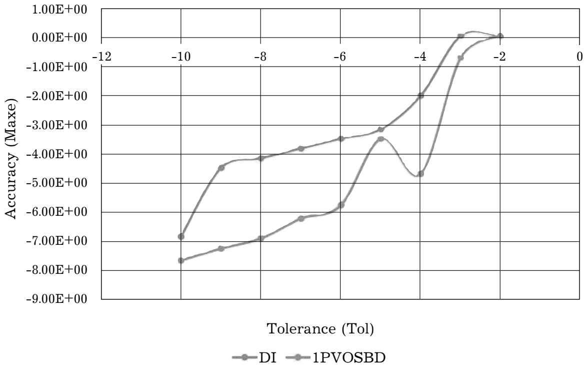

Fig. 4. Efficiency of 1PVOSBD and DI method.

Orbit Problem 3: Franco and Palacios (source: [19]).

$$y''(t) = -y(t) + \varepsilon\, e^{i\psi t}, \quad y|_0 = 1, \quad y'|_0 = i, \quad y \in \mathbb{C},$$

can be written in the equivalent form

$$\begin{gathered} y_1''(t) = -y_1(t) + \varepsilon \cos \psi t, \quad y_2''(t) = -y_2(t) + \varepsilon \sin \psi t, \\ y_1|_0 = 1, \quad y_1'|_0 = 0, \quad y_2|_0 = 0, \quad y_2'|_0 = 1, \end{gathered}$$

and $\varepsilon = 0.0001$ and $\psi = 0.01$ with the analytical solution

$$y_1(t) = \frac{1-\varepsilon-\psi^2}{1-\psi^2}\cos t + \frac{\varepsilon}{1-\psi^2}\varepsilon\, t\cos\psi t, \quad y_2(t) = \frac{1-\varepsilon-\psi^2}{1-\psi^2}\sin t + \frac{\varepsilon}{1-\psi^2}\varepsilon\, t\sin\psi t.$$

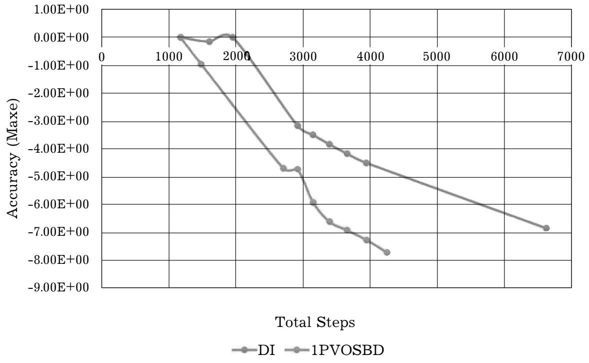

Fig. 5. Accuracy of 1PVOSBD and DI method.

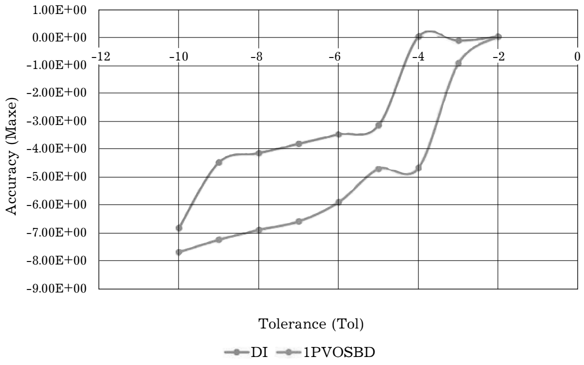

Fig. 6. Efficiency of 1PVOSBD and DI method.

7. Discussion and conclusion

When solving orbital problems, the DI method occurs some difficulties due to its division component, which may cause the divided difference to be small enhancing the round-off errors, especially with trigonometric solutions. This becomes evident when dealing with small tolerances (as shown in Figs.3 and 4) that both 1PVOSBD and DI are competitive for greater tolerance. From $TOL = 10^{(-6)}$ the 1PVOSBD breaks away and shows to be more efficient. The overall numerical results demonstrate the advantages of the 1PVOSBD over the DI method when solving orbital problems. In terms of accuracy, the 1PBDVSO is more accurate than the DI method and shows to be more stable, especially when facing problems with a higher level of difficulty. Figures 3, 5, and 7 illustrate the efficiency of both methods by the undermost curve. Hence, by comparison of efficiency and accuracy, we justify our conclusion that the method of choice would be the 1PVOSBD. This study has proposed a 1PVOSBD method as proven to be an efficient ODE solver due to reduced computational cost with minimal to no loss of accuracy. For future works, the 1PVOSBD method can be fitted with a block formulation to significantly lower computational costs.

[1] Krogh F. T. A variable-step, variable-order multistep method for the numerical solution of ordinary differential equations. Proc. of the IFIP Congress in Information Processing. **68**, 194 (1968).

[2] Hall G., Watt J. M. Modern numerical methods for ordinary differential equations. Clarendon Press (1976).

[3] Suleiman M. B. Generalised multistep Adams and backward differentiation methods for the solution of stiff and non-stiff ordinary differential equations. University of Manchester PhD Thesis (1979).

[4] Rasedee A. F. N. Direct method using backward difference for solving higher-order ordinary differential equations. University Putra of Malaysia PhD Thesis (2009).

[5] Rasedee A. F. N., Suleiman M. B., Ibrahim Z. B. Solving nonstiff higher-order odes using variable order step size backward difference directly. Mathematical Problems in Engineering. **2014**, Article ID 565137 (2014).

[6] Shampine L. F., Gordon M. K. Computed solutions of ordinary differential equations. W. H. Freeman (1975).

[7] Lambert J. D. Computational methods in ordinary differential equations. John Wiley & Son (1973).

[8] Rasedee A. F. N., Hamzah S. R., Ishak N., Mohd Ijam H., Suleiman M. B., Ibrahim Z. B., Abdul Sathar M. H., Ramli N. A., Kamaruddin N. S. Variable order variable stepsize algorithm for solving nonlinear Duffing oscillator. Journal of Physics: Conference Series. **890**, 012045 (2017).

[9] Rasedee A. F. N., Suleiman M. B., Ibrahim Z. B. Solving nonstiff higher order odes using variable order step size backward difference directly. Mathematical Problems in Engineering. **2014**, Article ID 565137 (2014).

[10] Simos T. E. Dissipative trigonometrically-fitted methods for linear second-order IVPs with oscillating solution. Applied Mathematics Letters. **17** (5), 601–607 (2004).

[11] Mohd Ijam H., Suleiman M. B., Rasedee A. F. N., Senu N., Ahmadian A., Salahshour S. Solving nonstiff higher-order ordinary differential equations using 2-point block method directly. Abstract and Applied Analysis. **2014**, Article ID 867095 (2014).

[12] Mohd Ijam H., Ibrahim Z. B., Suleiman M. B., Senu N., Rasedee A. F. N. Order and stability of 2-point block backward difference method. AIP Conference Proceedings. **1974**, 020054 (2018).

[13] Shokri L., Mehdizadeh Khalsaraei M., Atashyar A. A new two-step hybrid singularly P-stable method for the numerical solution of second-order IVPs with oscillating solutions. Iranian Journal of Mathematical Chemistry. **11** (2), 113–132 (2020).

[14] Wu X., Wang B., Mei L. Oscillation-preserving algorithms for efficiently solving highly oscillatory second-order ODEs. Numerical Algorithms. **86**, 693–727 (2021).

[15] Rasedee A. F. N., Abdul Sathar M. H., Hamzah S. R., Ishak N., Wong T. Z., Koo L. F., Ibrahim S. N. I. Block variable order step size multistep method for solving higher order ordinary differential equations directly. Journal of King Saud University-Science. **33** (3), 101376 (2021).

[16] Rasedee A. F. N., Abdul Sathar M. H., Othman K. I., Hamzah S. R., Ishak N. Two-Point Approximating non linear higher order ODEs by a three point block algorithm. Plos One. **16** (2), e0246904 (2021).

[17] Rasedee A. F. N., Abdul Sathar M. H., Deraman F., Mohd Ijam H., Suleiman M. B., Saaludin N., Rakhimov A. 2 point block backward difference method for solving Riccati type differential problems. AIP Conference Proceedings. **1775**, 030005 (2016).

[18] Stiefel E., Bettis D. G. Stabilization of Cowell's method. Numerische Mathematik. **12**, 154–175 (1969).

[19] Franco J. M., Palacios M. High-order P-stable multistep methods. Journal of Computational and Applied Mathematics. **30** (1), 1–10 (1990).

Метод змінного порядку кроку для розв'язування орбітальних задач із періодичними розв'язками

Раседі А. Ф. Н.[1], Джамалудін Н. А.[2], Наджиб Н.[1],
Абдул Сатар М. Х.[3], Вонг Т. Дж.[4], Коо Л. Ф.[4]

[1] *Економічний факультет і Муамалат,*
Університет ісламських наук Малайзії,
71800 Нілай, Негері Сембілан, Малайзія
[2] *Центр оборонних досліджень,*
Національний університет оборони Малайзії,
Кем Сунгай Бесі, 57000, Куала-Лумпур, Малайзія
[3] *Центр фундаментальних досліджень сільськогосподарської науки,*
Університет Путра Малайзія,
43400 UPM Серданг, Селангор, Малайзія
[4] *Кафедра науки і техніки,*
Факультет гуманітарних наук, менеджменту та науки,
Університет Путра Малайзії,
Бінтулу Саравак Кампус, 97008, Бінтулу, Саравак, Малайзія

Існуючі чисельні техніки зі змінним розміром кроку для розв'зування системи звичайних диференціальних рівнянь (ЗДР) вищого порядку вимагають безпосереднього обчислення коефіцієнтів інтегрування при кожній зміні кроку. У цьому дослідженні запропоновано розмір кроку змінного порядку, який дозволяє безпосереднє розв'язування орбітальних рівнянь вищого порядку. Запропоновано алгоритм, за яким обчислюються коефіцієнти інтегрування лише один раз на початку і, за необхідності, один раз наприкінці. Точність чисельного наближення підтверджено на відомих орбітальних диференціальних рівняннях. Для зменшення обчислювальних витрат для алгоритму предиктор-корректор отримано зв'язок між коефіцієнтами інтегрування різних порядків. Ефективність запропонованого методу підтверджується графічним поданням точності на усіх кроках оцінки.

Ключові слова: *прикладна математика, зворотна різниця, звичайні диференціальні рівняння, багатокроковість, змінний порядок кроку.*