Block variable order step size method for solving higher order orbital problems

Ahmad Fadly Nurullah Rasedee, Hazizah Mohd Ijam, Mohammad Hasan Abdul Sathar, Norizarina Ishak, Muhamad Azrin Nazri, Nur Shuhada Kamarudin, and Nur Ainna Ramli

Citation: AIP Conference Proceedings **1905**, 030028 (2017);
View online: https://doi.org/10.1063/1.5012174
View Table of Contents: http://aip.scitation.org/toc/apc/1905/1
Published by the American Institute of Physics

Block Variable Order Step Size Method For Solving Higher Order Orbital Problems

Ahmad Fadly Nurullah Rasedee[1,a)], Hazizah Mohd Ijam[2], Mohammad Hasan Abdul Sathar[2,3], Norizarina Ishak[4], Muhamad Azrin Nazri[1], Nur Shuhada Kamarudin[1] and Nur Ainna Ramli[1]

[1]*Faculty of Economics and Muamalat, Universiti Sains Islam Malaysia.*
[2]*Institute for Mathematical Research, Universiti Putra Malaysia.*
[3]*Centre of Foundation Studies for Agricultural Science, Universiti Putra Malaysia.*
[4]*Faculty of Science and Technology, Universiti Sains Islam Malaysia.*

[a)]Corresponding author: fadlynurullah@usim.edu.my

Abstract. Previous numerical methods for solving systems of higher order ordinary differential equations (ODEs) directly require calculating the integration coefficients at every step. This research provides a block multi step method for solving orbital problems with periodic solutions in the form of higher order ODEs directly. The advantage of the proposed method is, it requires calculating the integration coefficients only once at the beginning of the integration is presented. The derived formulae is then validated by running simulations with known higher order orbital equations. To provide further efficiency, a relationship between integration coefficients of various order is obtained.

INTRODUCTION

Various real-life applications ranging from the complexity of engineering to the volatility of finance can be modeled into mathematical equations, specifically differential equations. In this research, we deal with higher order ordinary differential problems with periodic solutions.

Consider the higher order ODEs of the form

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

with the initial conditions $\widetilde{Y}(\alpha) = \widetilde{\zeta}$, where

$$\widetilde{Y}(\alpha) = (y, y', \ldots, y^{(d-1)}) \qquad \text{and} \qquad \widetilde{\zeta} = (\zeta, \zeta', \ldots, \zeta^{(d-1)}), \tag{2}$$

in the interval $\alpha \leq x \leq \beta$.

Referring to works by authors such as [1], [2], [3], [4] and [5], we developed a two-point block backward difference formulation with a variable order step size (VOS) code for solving higher order ODEs directly. Initially, Suleiman in [4] devised a VOS algorithm to solve stiff and nonstiff higher order ODEs directly in divided difference form known as the direct integration method (DI) which was shown to be efficient in terms of accuracy and calculation cost. The only drawback to DI is the tedious calculations of integration coefficients at every step change. Since then, the research on solving higher order ODEs directly has been expanded by authors such as [6], [7], [8], [9] and etc.

The current research, proposes a two point block method in the form of a predictor (explicit) - corrector (implicit) formulae. Because in a backward difference formulation, the integration coefficients are calculated only once at the beginning and by obtaining a relationships between the predictor and corrector of different blocks, we are able to reduce the computational costs. Next, we proceed with the formulation of the backward difference integration coefficients.

Proceedings of the 13th IMT-GT International Conference on Mathematics, Statistics and their Applications (ICMSA2017)
AIP Conf. Proc. 1905, 030028-1–030028-6; https://doi.org/10.1063/1.5012174
Published by AIP Publishing. 978-0-7354-1595-9/$30.00

DERIVATION OF THE PREDICT-CORRECT BACKWARD DIFFERENCE METHOD

By integrating (1) once and denoting the numbered of blocks by b we have

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+b}} f(x, \widetilde{Y})dx, \qquad b = 1, 2. \tag{3}$$

Substituting the the Newton-Gregory backward difference polynomial,

$$P_n(x) = \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n, \qquad s = \frac{x - x_n}{h} \tag{4}$$

and changing the limit of integration by replacing $dx = hds$ gives

$$y^{(d-1)}(x_{n+1}) = y^{(d-1)}(x_n) + \int_0^b \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n \; hds \tag{5}$$

The equation above can be denoted by

$$y^{(d-1)}(x_{n+1}) = y^{(d-1)}(x_n) + h \sum_{i=0}^{k-1} \gamma_{b,1,i} \nabla^i f_n \; ds \qquad \text{where} \qquad \gamma_{b,1,i} = (-1)^i \int_0^b \begin{pmatrix} -s \\ i \end{pmatrix} ds. \tag{6}$$

Next, we define $G_{b,1}(t)$ as the generating function for the coefficients $\gamma_{b,1,i}$ as follows

$$G_{b,1}(t) = \sum_{i=0}^{\infty} \gamma_{b,1,i} t^i. \tag{7}$$

By substituting $\gamma_{b,1,i}$ from (6) into (7), the generating function $G_{b,1}(t)$ can be written as

$$G_{b,1}(t) = \int_0^b e^{-s \log(1-t)} dt. \tag{8}$$

Solving the integration above, allows the generating function to be represented as

$$G_{b,1}(t) = -\left[\frac{(1-t)^{-b}}{\log(1-t)} - \frac{1}{\log(1-t)} \right]. \tag{9}$$

Then, by expanding the generating function in terms of $\gamma_{b,1,k}$ provides the following integration coefficients

$$\gamma_{b,1,k} = 1 - \sum_{i=0}^{k-1} \left(\frac{\gamma_{b,1,i}}{k - i + 1} \right), \quad k = 1, 2, \ldots, \quad \gamma_{b,1,0} = b. \tag{10}$$

The dth order generating function is obtained by integrating (1) d times and by mathematical induction the general solution has the following form

$$y(x_{n+1}) = y(x_n) + (bh)y'(x_n) + \ldots + \frac{(bh)^{(d-1)}}{(d-1)!} y^{(d-1)}(x_n) + \frac{h^d}{d!} \sum_{i=0}^{k-1} \gamma_{b,d,i} \nabla^i f_n \; ds. \tag{11}$$

Similarly to $G_{b,1}(t)$, the d-th generating function, $G_{b,d}(t)$ gives

$$G_{b,d}(t) = \frac{1}{(d-1)!} \left[\frac{b^{d-1}}{\log(1-t)} - \frac{(d-1)! G_{b,d-1}(t)}{\log(1-t)} \right] \tag{12}$$

and leads to the following general representation of the explicit coefficients

$$\gamma_{b,d,0} = \gamma_{b,d-1,1}, \quad \gamma_{b,d,k} = \gamma_{b,d-1,k+1} - \sum_{i=0}^{k-1} \left(\frac{\gamma_{b,d,i}}{k-i+1} \right), \quad k = 1, 2, \ldots. \tag{13}$$

The generalized implicit formulae for the corrector is obtained by changing the limit of integration then followed by the similar steps as used to obtain the explicit generating function (see [9]). The implicit generating function will have the following representation

$$G^*_{b,d}(t) = \frac{1}{(d-1)!} \left[\frac{b^{d-1}(1-t)^b}{\log(1-t)} - \frac{(d-1)! G^*_{b,d-1}(t)}{\log(1-t)} \right]. \tag{14}$$

To prevent redundancy in calculating the integration coefficients, a recurring relationship between the d-th order explicit and implicit coefficients are given by

$$\sum_{i=0}^{k} (k-i+1)\gamma^*_{b,d,i} = \gamma_{b,d,k} \tag{15}$$

which has similarity to the relationship obtained in [10].

DETERMINING ORDER AND STEP SIZE

Accepting of the results of an integration step is vital when implementing a VOS algorithm. This is because when establishing a VOS algorithm, two criteria are taken into consideration, efficiency and reliability. The order and step size strategy used determines the efficiency of the algorithm whereas, the acceptance criteria determine its reliability. In a VOS algorithm, the order can easily be increased or decreased simply by including or disregarding the amount of back values used. Experience dictates that when implementing an Adams based code for solving nonstiff problems, the order strategy used should be unbiased to selecting lower order. For purpose of this research, order strategies as suggested in [11] is adopted to the algorithm.

Lets denote the calculated step size as h and the final step size as h_{end}. To reduce the number of rejected steps and to obtain an acceptable estimate of h_{end}, we include a safety factor denoted by R which is multiplied with h such that $h_{end} = Rh$. To overcome known concerns relating with stability of the variable step size technique, a ratio restriction of successive steps size is implemented (refer Shampine and Gordon [11]). We conclude the formulation of the predict-correct formulae by adopt the doubling and halving the step size algorithm as described in [12] with the implementation of the step size changing technique from [13].

ERROR ESTIMATION

Similar techniques conducted in [3], we begin our estimation with the following predictor

$$\begin{aligned} {}^{pr}y^{(d)}_{n+b} &= \sum_{i=0}^{k-1} \gamma_{b,0,i} \nabla^i f_n \\ {}^{pr}y^{(d-1)}_{n+b} &= y^{(d-1)}_n + h \sum_{i=0}^{k-1} \gamma_{b,0,i} \nabla^i f_n \\ \vdots \quad & \quad \vdots \qquad\qquad \vdots \\ {}^{pr}y_{n+b} &= \sum_{i=0}^{d-1} \frac{(bh)}{i!} y^{(i)}_n + h^d \sum_{i=0}^{k-1} \gamma_{b,d,i} \nabla^i f_n, \end{aligned} \tag{16}$$

where the coefficients is independent of k. Next, is the corrector taking the form of

$$\begin{aligned} {}^{cr}y_{n+b}^{(d)} &= \sum_{i=0}^{k} \gamma_{b,d,i}^{*} \nabla_{pr}^{i} f_{n+b} \\ {}^{cr}y_{n+b}^{(d-1)} &= y_{n}^{(d-1)} + h \sum_{i=0}^{k} \gamma_{b,d,i}^{*} \nabla_{pr}^{i} f_{n+b} \\ &\vdots \\ {}^{cr}y_{n+b} &= \sum_{i=0}^{d-1} \frac{(bh)}{i!} y_{n}^{(i)} + h^{d} \sum_{i=0}^{k} \gamma_{b,d,i}^{*} \nabla_{pr}^{i} f_{n+b}. \end{aligned} \tag{17}$$

Because of the relationship between explicit and implicit coefficients as shown in (15), the corrector can be expressed as

$$\begin{aligned} {}^{cr}y_{n+b}^{(d)} &=^{pr} y_{n+b}^{(d)} + \gamma_{b,0,i}^{*} \nabla_{pr}^{i} f_{n+b} \\ {}^{cr}y_{n+b}^{(d-1)} &=^{pr} y_{n+b}^{(d-1)} + h\gamma_{b,1,i}^{*} \nabla_{pr}^{i} f_{n+b} \\ &\vdots \\ {}^{cr}y_{n+b} &=^{pr} y_{n+b} + h^{d} \gamma_{b,d,i}^{*} \nabla_{pr}^{i} f_{n+b}. \end{aligned} \tag{18}$$

The local truncation error (LTE) is represented by

$$\begin{aligned} \widetilde{E}_{k} &= \gamma_{b,0,k}^{*} \nabla_{pr}^{k} f_{n+b} \\ \widetilde{E}_{k}^{1} &= h\gamma_{b,1,k}^{*} \nabla_{pr}^{k} f_{n+b} \\ &\vdots \\ \widetilde{E}_{k}^{d} &= h^{d} \gamma_{b,d,k}^{*} \nabla_{pr}^{k} f_{n+b}. \end{aligned} \tag{19}$$

by manner of Milne error estimate. As described in [14], for an efficient order and step size algorithm a suitable $\widetilde{E}_{k}^{(d-p)}$ is dependent on the appropriate p. Due to the $P_{k}EC_{k+1}E$ algorithm applied in this research, the asymptotic validity can be established using

$$\widetilde{E}_{k+1}^{d-p} = h^{d-p} \gamma_{b,d-1,k+1}^{*} \nabla^{k+1} f_{n+b}. \tag{20}$$

NUMERICAL RESULTS

Numerical solutions for solving orbital problems in the form of second order ODEs have been investigated by authors such as [15], [16] and [17]. In the current work, the 2PBVOS method is tested with known orbital problems. Numerical results are compared against the Direct Integration (DI) method established by Suleiman in [4]. We compare the accuracy and total steps taken between both methods to determine the efficiency of the 2PBVOS method. Error calculations are done with respect to absolute error, relative error and mixed error tests (see [10]). Notations below indicates

FSTPS: failed steps,
STPS: total steps,
TOL: the tolerance used,
,
MAXERR: the overall maximum error,
DI: direct integration,
2PBVOS: two-point block backward difference.

Orbit Problem 1: Two-body problem

$$\begin{aligned} y_1''(x) &= \frac{-y_1(x)}{r}, \qquad y_1(0) = 1, \quad y_1'(0) = 0, \\ y_2''(x) &= \frac{-y_2(x)}{r}, \qquad y_2(0) = 0, \quad y_1'(0) = 1, \\ &0 \le x \le 16\pi \end{aligned} \tag{21}$$

where

$$r = (y_1^2(x) + y_2^2(x))^{\frac{3}{2}} \tag{22}$$

with the analytical solution

$$y_1(x) = \cos x, \qquad y_2(x) = \sin x. \tag{23}$$

Orbit Problem 2: Stiefel and Bettis [18]

$$y''(x) = -y(x) + \varepsilon e^{it}, \qquad y(0) = 1, \quad y'(0) = 0.9995i, \quad y \in \mathbb{C}, \qquad 0 \leq x \leq 1000, \tag{24}$$

can be written the equivalent form

$$\begin{aligned} y_1''(x) &= -y_1(x) + \varepsilon \cos x, \qquad y_1(0) = 1, \quad y_1'(0) = 0, \\ y_2''(x) &= -y_2(x) + \varepsilon \sin x, \qquad y_2(0) = 0, \quad y_2'(0) = 0.9995, \end{aligned} \tag{25}$$

where $\varepsilon = 0.001$ with the analytical solution

$$y(x) = y_1(x) + iy_2(x), \quad y_1, y_2 \in \mathbb{R}, \tag{26}$$

$$y_1(x) = \cos x + \frac{1}{2}\varepsilon x \sin x, \qquad y_2(x) = \sin x + \frac{1}{2}\varepsilon x \cos x. \tag{27}$$

TOL	MTD	STPS	FSTPS	MAXERR
10^{-2}	DI	95	5	1.00000(0)
10^{-2}	2PBVOS	43	3	1.00000(0)
10^{-4}	DI	137	2	1.27865(−2)
10^{-4}	2PBVOS	81	2	9.67312(−2)
10^{-6}	DI	179	2	1.55778(−3)
10^{-6}	2PBVOS	150	2	3.62825(−3)
10^{-8}	DI	210	0	2.68910(−4)
10^{-8}	2PBVOS	329	8	7.13106(−5)
10^{-10}	DI	366	1	5.69993(−7)
10^{-10}	2PBVOS	398	15	3.67720(−6)

TABLE 1: Numerical Results for Problem 1

TOL	MTD	STPS	FSTPS	MAXERR
10^{-2}	DI	1176	1	1.06797(0)
10^{-2}	2PBVOS	922	5	1.04644(0)
10^{-4}	DI	2488	4	9.60862(−3)
10^{-4}	2PBVOS	1073	0	7.98562(−2)
10^{-6}	DI	3145	1	3.13723(−4)
10^{-6}	2PBVOS	2471	1	2.33381(−4)
10^{-8}	DI	3650	1	6.71619(−5)
10^{-8}	2PBVOS	5701	31	3.02409(−5)
10^{-10}	DI	6615	2	1.37306(−7)
10^{-10}	2PBVOS	6623	16	6.17905(−7)

TABLE 2: Numerical Results for Problem 2

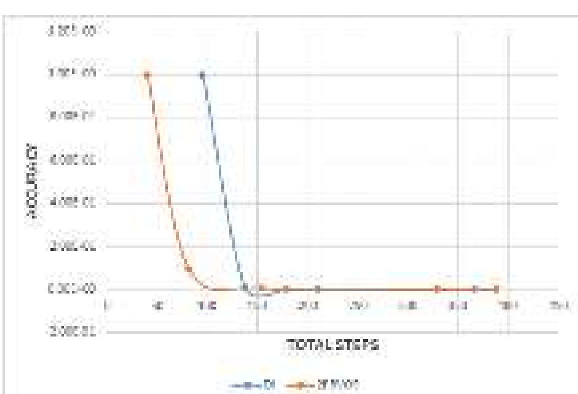

FIGURE 1: Efficiency of 2PBVOS and DI method for Problem 1

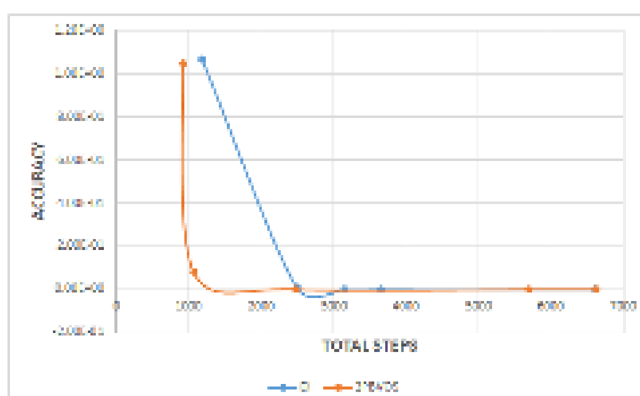

FIGURE 2: Efficiency of 2PBVOS and DI method for Problem 2

DISCUSSION AND CONCLUSION

The division component in a divided difference formulation might effect the round off errors of the DI method especially with trigonometric solutions. This might be the cause of difference in total steps. As shown in Table 1 and 2, the 2PBVOS method requires significantly less number steps, for bigger tolerances without loss of accuracy. When dealing with finer tolerances, both methods show to be competitive.

Figure 1 and 2 represent analysis of accuracy against number of total steps. As clearly illustrated in both figures, the 2PBVOS method has the undermost curve compared to the DI method for large tolerances. Being that the undermost curve denotes the efficiency of the method (see [14]), we can conclude that the 2PBVOS method show to be more sufficient then the DI method

ACKNOWLEDGMENTS

The research conducted by this article has been supported by Universiti Putra Malaysia under Grant Putra (GP), project number GP-IPS/2017/9526600, Ministry of Education (MoE) under the Fundamental Research Grant Scheme (FRGS), project number USIM/FRGS/FEM/055002/51517 and Universiti Sains Islam Malaysia (USIM) under the Short Term Grant Scheme, project number PPP/FST-0117/051000/11417.

REFERENCES

1. C. W. Gear, Mathematics of Computation **21**, 146–156 (1967).
2. G. Hall and J. M. Watt, *Modern numerical methods for ordinary differential equations* (Clarendon Press, Oxford, 1976).
3. G. Hall and M. B. Suleiman, IMA: Journal of Numerical Analysis **1**, 427–438 (1981).
4. 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, U.K. 1979.
5. M. B. Suleiman, Applied Mathematics and Computation **33**, 197–219 (1989).
6. Z. Omar and M. B. Suleiman, International Journal of Simulation and Process Modelling **2**, 227–231 (2006).
7. Z. Abdul Majid, N. Z. Mokhtar, and M. B. Suleiman, Mathematical Problems in Engineering **2012** (2012).
8. Z. B. Ibrahim, K. I. Othman, and M. Suleiman, Sci **39**, 502–510 (2012).
9. A. F. N. Rasedee, M. B. Suleiman, and Z. B. Ibrahim, Mathematical Problems in Engineering **2014** (2014).
10. M. B. Suleiman, Z. B. Ibrahim, and A. F. N. Rasedee, Mathematical Problems in Engineering **2011**, p. 18 pages (2011).
11. L. F. Shampine and M. K. Gordon, *Computed solutions of ordinary differential equations* (W. H. Freeman, San Francisco, 1975).
12. F. T. Krogh, Siam: Journal on Numerical Analysis **10**, 949–965 (1973).
13. J. D. Lambert, *Computational methods in ordinary differential equations* (John Wiley and Sons, New York, 1973).
14. A. F. N. Rasedee, *Direct method using backward difference for solving higher order ordinary differential equations*, Master’s thesis, University Putra of Malaysia, Selangor, Malaysia (2009).
15. Simos, T. E. , Num. Algor. **17**, 601–607 (2004).
16. K. Tselios and T. E. Simos, Journal of Computational and Applied Mathematics **175**, 173–181 (2005).
17. D. Papadopoulos and T. Simos, Appl. Math. Inf. Sci **7**, 433–437 (2013).
18. E. Stiefel and D. G. Bettis, Numerische Mathematik **13**, 154–175 (1969).