Sains Malaysiana 35(2) 2006: 63-68

Direct Integration Implicit Variable Steps Method for Solving Higher Order Systems of Ordinary Differential Equations Directly

(Kaedah Kamiran Terus Langkah Berubah Tersirat Bagi Menyelesaikan Sistem Persamaan Terbitan Peringkat Tinggi Secara Langsung)

ZANARIAH ABDUL MAJID & MOHAMED BIN SULEIMAN

ABSTRACT

In this paper, a direct integration implicit variable step size method in the form of Adams Moulton Method is developed for solving directly the second order system of ordinary differential equations (ODEs) using variable step size. The existing multistep method involves the computations of the divided differences and integration coefficients in the code when using the variable step size or variable step size and order. The idea of developing this method is to store all the coefficients involved in the code. Thus, this strategy can avoid the lengthy computation of the coefficients during the implementation of the code as well as to improve the execution time. Numerical results are given to compare the efficiency of the developed method with the 1-point method of variable step size and order code (1PDVSO) in Omar (1999).

Keywords: *implicit method; variable steps method; ordinary differential equations*

ABSTRAK

Dalam makalah ini, suatu kaedah kamiran terus dengan saiz langkah berubah tersirat dalam bentuk Kaedah Adams Moulton dibangunkan bagi menyelesaikan secara terus sistem persamaan peringkat dua menerusi saiz langkah berubah. Kaedah multilangkah yang sedia ada melibatkan pengiraan beza pembahagi dan pekali kamiran dalam kod apabila menggunakan saiz langkah berubah atau saiz langkah berubah dan peringkat. Ide di sebalik kaedah ini ialah untuk menyimpan kesemua pekali yang terlibat dalam kod. Maka strategi ini boleh menghindarkan pengiraan berpanjangan pekali berkaitan semasa implementasi kod tersebut disamping memperbaiki masa pelaksanaan. Hasil berangka diberikan untuk membandingkan keberkesanan kaedah yang telah dibangunkan itu dengan kaedah 1-titik bagi saiz langkah berubah dan kod peringkat (1PDVSO) dalam Omar (1999).

Kata kunci: *kaedah tersirat; kaedah langkah berubah; persamaan terbitan biasa*

INTRODUCTION

In this paper, we consider solving directly the second order non-stiff initial value problems (IVPs) for systems of second order ODEs of the form

$$y'' = f(x, y, y'), \quad y(a) = y_0, \quad y'(a) = y'_0, \quad x \in [a, b] \tag{1}$$

Eq. (1) arises from many physical phenomena in a wide variety of applications especially in engineering such as the motion of rockets or satellites, fluid dynamic, electric circuit and other areas of application. The approach for solving the system of higher order ODEs directly has been suggested by several researchers such as Gear (1971), Lambert (1993), Omar (1999) and Suleiman (1979). The current multistep method for variable step (VS) or variable step and order (VSVO) technique for solving the systems of second order ODEs as described by the above researchers will involve tedious computations of divided difference and recurrence relation in computing the integration coefficients. A system of higher order can be reduced to a system of first order equations and then solved using first order ODEs methods. This approach is very well established but it obviously will enlarge the system of first order ODEs.

The 1-point method of variable step size and order code (1PDVSO) in Omar (1999) solve the higher order ODEs directly. This code will involve tedious computations of divided differences and recurrence relation in computing the integration coefficients. We will explain briefly the computations of the integration coefficients involved in

The interpolation polynomial which interpolates the values of a function *F* at the points in terms of divided differences, has the form,

$$P_{k,n}(x) = F_{[n]} + (x - x_n)\, F_{[n,n-1]} + \ldots + (x-x_n)(x - x_{n-1})\ldots(x - x_{n-k+2})\, F[_{n,n-1,\ldots,n-k+1}] \tag{2}$$

where the i – th divided differences is defined by

$$F[n,n\text{-}1,\ldots n\text{-}i] = \frac{F[n,n\text{-}1,\ldots n\text{-}i+1] - F[n\text{-}1,n\text{-}2,\ldots n\text{-}1]}{(x_n - x_{n-i})} \tag{3}$$

Define $P_{n,i}(x) = (x - x_n)(x - x_{n-1}) \cdots (x - x_{n-i+1})$ for $i = 1,2, \ldots, k$ and $P_{n,o}(x) = 1$

Also define $t > 0$, to be the *t*-fold integral and the integration coefficients can be obtained by using the following definition:

$$g_{i,t} = \underbrace{\int_{x_n}^{x_{n+1}} \int_{x_n}^{x} \cdots \int_{x_n}^{x}}_{t \text{ times}} P_{n,i}(x)\, dx dx \cdots dx \tag{5}$$

and

$$g_{i,0} = P_{n,i}(x_{n+1}) \quad \text{for } i = 1,2, \ldots, k. \tag{6}$$

Substitute (4) into (5) and then solve using integration by parts yields

$$g_{i,t} = (x_{n+1} - x_{n-i+1})\, g_{i-1,t} - t\, g_{i-1,t+1} \tag{7}$$

Fo

$$g_{0,t} = \underbrace{\int_{x_n}^{x_{n+1}} \int_{x_n}^{x} \cdots \int_{x_n}^{x}}_{t \text{ times}} dx dx \cdots dx = \frac{(x_{n+1} - x_n)^t}{t.}, \tag{8}$$

see Omar (1999) and Suleiman (1979) for detail.

Equations (7) and (8) can be used to compute the required coefficients in triangular arrays as in Table 1, starting with the first row from left to right.

The actual number of coefficients required will depend on the problem and the order of the method being solved. in Table 1 is defined as where is the current -step method used and is the order of the differential equation. The maximum value of is denoted by

TABLE 1. Integration coefficients

t \ i	0	1	2	.	.	.
0	g [illegible]	g [illegible]	g [illegible]	.	.	.
1	g [illegible]	g [illegible]	g [illegible]	.	.	g [illegible]
2	g [illegible]	g [illegible]	g [illegible]			
.	.	.	.			
.	.	.				
.	.					
k [illegible] $+1$	g [illegible]					

The aim of this paper is to investigate the performance of the direct integration implicit variable step size method (1PDI) presented in the simple form of the Adams Moulton method for solving (1) using variable step size. The method is in a simple form but we intend for efficiency and economically. The idea of the code is to avoid tedious and repetitive computation of the integration coefficients as in Table 1 that can be very costly. Hence, the developed code will store all the coefficients of the method. However because of the step size variations, many formulae need to be stored and may cause a high initial overhead. As the computational work increases the advantage of the method will be obvious when the execution time is compared with the 1PDVSO code in Omar (1999).

METHOD

FORMULATION OF THE DIRECT INTEGRATION IMPLICIT VARIABLE STEPS METHOD

In Figure 1, the computed step has the step size *h* but in each previous step the length of the interval is *rh, qh* and *ph*. The values of *r, q* and *p* are varying; therefore the interval in Figure 1 is not equally apart.

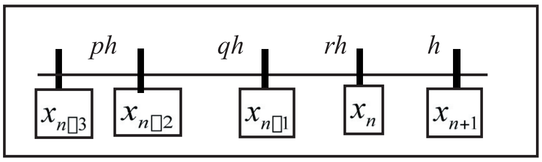

FIGURE 1. 1-Point Implicit Method

The point, at can be obtained by integrating (1) once, i.e

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

Therefore

$$y'(x_{n+1}) = y'(x_n) + \int_{x_n}^{x_{n+1}} f(x, y, y')\,dx . \qquad (9)$$

The function in (9) will be approximated using Lagrange interpolating polynomial and the interpolation points involved are ... . Taking and by replacing ds , the value of can be obtained by integrating (9) over the interval using MAPLE, refer to Char et al. (1992) and the following corrector formulae can be obtained,

$$y'(x_{n+1}) = y'(x_n) + h \Bigg(\frac{-(3+5q+10\,(r+q\;\;+r^2))}{60\;\; p(p+q)(p+q+r)(1+p+q+r)} f_{n-3}$$

$$\frac{[3+5(p+q)+10\,(r+\;p\;\;+q\;\;+r^2)]}{60\;p\;(q+r)(1+q+r)} f_{n-2} - \frac{[3+5(p+2q)+10\,(\;\;+q^2+r+p\;\;+2q\;\;+r^2)]}{60\;q\;(p+q)(1+r)} f_{n-1}$$

$$+ \frac{(3+5(p+2q+3r)+10\,(p\;\;+q^2+2p\;\;+4q\;\;)+60\,(pqr+q^2r+r^2+p\;\;^2+r^2+2q\;\;^2))}{60\;r(q+r)(p+q+r)} f_n$$

$$+ \frac{([illegible]\;+5\,(p+2q+3r)+10\,(p\;\;+q^2+2p\;\;+4q\;\;)+30\,(pqr+q^2r+2r^2+p\;\;^2+r^3+2q\;\;^2))}{60\,(1+r)(1+q+r)(1+p+q+r)} f_{n+1} \Bigg) . \qquad (10)$$

Now, integrating (1) twice gives

$$\int_{x_n}^{x_{n+1}} \int_{x_n}^{x} y''(x)\,dx\,dx = \int_{x_n}^{x_{n+1}} \int_{x_n}^{x} f(x, y, y')\,dx\,dx \qquad (11)$$

Therefore

$$y(x_{n+1}) - y(x_n) - hy'(x_n) = \int_{x_n}^{x_{n+1}} (x_{n+1} - x) f(x, y, y')\,dx \qquad (12)$$

and replace the in (11) with the same interpolation polynomial through the points to as in (10). Taking and by replacing and integrate (12) over the interval using MAPLE and the following corrector formulae can be obtained,

$$y(x_{n+1}) - y(x_n) - hy'(x_n) = h^2 \left\{ \frac{-(1+2q+4r+5r(q+r))}{60\, p(p+q)(p+q+r)(1+p+q+r)} f_{n-3} \right.$$

$$+ \frac{(1+2(p+q+2r)+5(p\ \ +q\ \ +r^2))}{60\ q\ (q+r)(1+q+r)} f_{n-2} - \frac{(1+2(p+2q+2r)+5(q\ \ +q^2+p\ \ +2q\ \ +r^2))}{60\ q\ (p+q)(1+r)} f_{n-1}$$

$$+ \frac{(1+2(p+2q+3r)+5(q\ \ +q^2+2p\ \ +3r^2)+0\ (q\ \ +pqr+q^2r+p\ ^2+r^2+2q\ ^2))}{60\ r(q+r)(p+q+r)} f_n$$

$$\left. + \frac{(2+3(p+2q+3r)+5(q\ \ +q^2+2p\ \ +3r^2)+0\ (2q\ \ +pqr+q^2r+p\ ^2+r^3+2q\ ^2))}{60\ (1+r)(1+q+r)(1+p+q+r)} f_{n+1} \right\}. \quad (13)$$

The method is the combination of predictor of order 4 and the corrector of order 5. The predictor formulae were similarly derived where the interpolation points involved are .... .

VARIABLE STEP SIZE STRATEGY

The step size strategy used in the code is a modified version of Shampine (1975). The choices for the next step size will be restricted to half, double or the same as the previous step size and the successful step size will remain constant for at least two blocks before considering it to be doubled. The step size strategy helps to minimize the choices of *r, q* and *p*.

Those possible step size ratios of *r, q* and *p* have ten combinations. In case of successful step size, the choices during constant step size are (*r*=1, *q*=1, *p*=1), (*r*=1, *q*=2, *p*=2), (*r*=1, *q*=1, *p*=2), (*r*=1, *q*=0.5, *p*=0.5) or (*r*=1, *q*=1, *p*=0.5). When the step size is doubled the possible ratios are (*r*=0.5, *q*=0.5, *p*=1), (*r*=0.5, *q*=0.5, *p*=0.25) or (*r*=0.5, *q*=0.5, *p*=0.5). In case of step size failure the possible ratios are (*r*=2, *q*=2, *p*=1) or (*r*=2, *q*=2, *p*=2). Substituting the common ratios of *r, q* and *p* in (10) and (13) will give the corrector formulae for the direct integration implicit variable steps method. For example, the corrector formulae when (r=1, *q*=2, *p*=2) are as follows:-

$$y'_{n+1} = y'_n + \frac{h}{14400}(5285 f_{n+1} + 11648 f_n - 2895 f_{n-1} + 415 f_{n-2} - 3\ \ f_{n-3})$$

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{7200}(735 f_{n+1} + 3472 f_n - 690 f_{n-1} + 9\ \ f_{n-2} - 1\ \ f_{n-3})$$

RESULTS AND DISCUSSION

In order to study the efficiency of the developed code, we present some numerical experiments for the following three problems.

The 1PDI and 1PDVSO were applied to the following test problems:

Problem 1:

$y'' = 2y^3,\quad y(0) = 1,\quad y'(0) = -1,\quad x \in [0,5]$

Solution: $y(x) = \dfrac{1}{x+1}$

Source: Roberts Jr.(1979)

Problem 2:

$y_1'' = -y_2' + \cos x,\quad y_1(0) = -1,\quad y_1'(0) = -1,$

$y_2'' = y_1 + \sin x,\quad y_2(0) = 1,\quad y_2'(0) = 0,\quad x \in [0,4\pi]$

Solution $y_1(x) = -\cos x - \sin x,\quad y_2(x) = \cos x.$

Source: Bronson (1973)

Problem 3:

$y_1'' = \dfrac{-y_1}{r},\quad y_1(0) = 1,\quad y_1'(0) = 0,$

$y_2'' = -\frac{y_2}{r}$, $y_2(0) = 0$, $y_2'(0) = 1$,

$r = \sqrt{y_1^2 + y_2^2}$ $x \in [0, 5]$

Solution: $y_1(x) = \cos x$, $y_2(x) = \sin x$.

Source: Suleiman (1989)

The following notations are used in the tables:

TOL Tolerance

MTD Method employed

TS Total steps taken

FS Total failure step

MAXE Magnitude of the maximum error of the computed solution

AVERR Average error

TIME The execution time taken in microseconds

1PDI Implementation of the direct integration implicit variable step size method

1PDVSO Implementation of the direct integration implicit variable step size and order method using the integration coefficients of in Omar (1999)

The errors calculated are defined as

$abs\left(y^{(s+1)}_{n+1} - y^{(s)}_{n+1}\right)$ $s = 0,1,2\ldots$

where is the t-th component of the approximate y and for this case we let $t = 1$. The absolute error test corresponds to A=1, B=0, the mixed test corresponds to A=1, B=1 and finally A=0, B=1 corresponds to the relative error test. The mixed

TABLE 2. Comparison between the IPDI and IPDVSO for solving problem 1

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

error tests were used for all the problems. The maximum error and average error are defined as follows:-

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

TABLE 4. Comparison between the IPDI and IPDVSO for solving problem 3

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

the numerical results for the three problems when solved using 1PDI and 1PDVSO. In most cases the maximum error of 1PDI is comparable or one order less than 1PDVSO. The results indicated that the total number of steps taken by 1PDVSO is much lesser than 1PDI at larger tolerances. This is expected since 1PDVSO is a variable step and order method with various order of k, and therefore the method will generally have larger step size and hence lesser total steps. However, at smaller tolerances, the total steps of 1PDI are better than 1PDVSO. The code 1PDI is also better in terms of execution times compared to 1PDVSO even though the total steps taken in 1PDI are larger than the steps taken by 1PDVSO in the tested problems. Therefore, we can conclude that the cost of computing the divided differences and integration coefficients in the 1PDVSO code is the major disadvantage when permitting random variations in the step size, thus the computational cost increases when the codes are implemented in variable step size and order.

TABLE 5. The ratios execution times of the IPDVSO method to the IPDI Method for solving problem 1 to 3

TOL	PROB 1	PROB 2	PROB 3
[illegible]	1.55	1.22	1.19
[illegible]	1.7[illegible]	1.4[illegible]	1.57
[illegible]	2.[illegible]	2.[illegible]1	1.46
[illegible]	2.[illegible]	2.[illegible]5	2.15
[illegible]	3.[illegible]	2.29	2.2[illegible]

In Table 5, the ratios of the execution times are greater than one show that the 1PDI code is efficient than 1PDVSO. Hence, these demonstrate the advantage of the 1PDI code in the form of standard multistep method because the cost per step is cheaper.

In general, we have shown the efficiency of the one-point direct integration implicit code presented as in the form of Adams Moulton method with variable step size is suitable for solving second order ODES.

REFERENCES

Bronson, R. 1973. *Modern Introductory Differential Equation: Schaum's Outline Series*. USA: McGraw-Hill Book Company.

Char, B.W., Geddes, K.O., Gonnet, G.H., Leong, B.L., Monagan, M.B. and Watt, S.M. 1992, *First Leaves: A Tutorial Introduction to Maple V*, Waterloo Maple Publishing, Springer-Verlag.

Gear, C.W. 1971. *Numerical Initial Value Problems in Ordinary Differential Equations*. New Jersey: Prentice Hall, Inc.

Lambert, J.D. 1993. *Numerical Methods For Ordinary Differential Systems. The Initial Value Problem*. New York: John Wiley & Sons, Inc.

Omar, Z. 1999. *Developing Parallel Block Methods For Solving Higher Order ODEs Directly*, Ph.D. Thesis, University Putra Malaysia, Malaysia.

Roberts Jr, C.E. 1979. *Ordinary Differential Equation: A Computational Approach*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Shampine, L.F. and Gordon, M.K. 1975. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. H. Freeman and Company, San Francisco.

Suleiman, M.B. 1979. *Generalised Multistep Adams and Backward Differentiation Methods for the Solution of Stiff and Non-Stiff Ordinary Differential Equations*. Ph.D. Thesis. University of Manchester.

Suleiman, M.B. 1989. *S*olving Higher Order ODEs Directly by the Direct Integration Method. *Applied Mathematics and Computation* 33: 197-219.

Zanariah Abdul Majid
Institut Perguruan Teknik
Jalan Yaacob Latif
Bandar Tun Razak
56000 Kuala Lumpur
MALAYSIA

Mohamed Suleiman
Mathematics Department
Faculty of Science
Universiti Putra Malaysia
43400 Serdang, Selangor D.E.
MALAYSIA