MATEMATIKA, 2006, Volume 22, Number 2, pp. 137–146

Performance of 4-Point Diagonally Implicit Block Method for Solving Ordinary Differential Equations

[1]**Zanariah Abdul Majid** & [2]**Mohamed Suleiman**
[1]Mathematics Department ,Technical Teachers' Training College,
Jln Yaacob Latif, Bandar Tun Abdul Razak, 56000 Kuala Lumpur, Malaysia.
[2]Mathematics Department, Faculty Science, Universiti Putra Malaysia,
43400 Serdang, Selangor Darul Ehsan, Malaysia.
e-mail: [1]zanamajid@hotmail.com, [2]m1suleiman@yahoo.com.my

Abstract This paper describes the development of a 4-point diagonally implicit block method for solving first order Ordinary Differential Equations (ODEs) using variable step size. This method will estimate the solutions of Initial Value Problems (IVPs) at four points simultaneously. The method developed is suitable for the numerical integration of non stiff and mildly stiff differential systems. The performances of the 4-point block method are compared in terms of maximum error and total number of steps to the non block method 1PVSO.

Keywords 4-Point, Implicit Block Method, Block Method, Ordinary Differential Equations

1 Introduction

Block method for numerical solution had been proposed by several researchers such as Rosser [8], Shampine and Watts [9], Worland [12], Chu and Hamilton [3] and Omar [7]. A block method will computes simultaneously the solution values at several distinct points on the x-axis in the block.

The present codes of changing the step size with multistep method involved tedious computations of divided difference and recurrence relation in computing the integration coefficients. (See Omar [7] and Suleiman [11] for detail). In this paper, we are interested in the numerical integration of IVPs of the form

$$y' = f(x, y), \qquad y(a) = y_0 \qquad a \leq x \leq b. \tag{1}$$

There are many existing methods for solving the ODEs as in (1) but those methods will only approximate the numerical solutions at one point sequentially. Therefore we need a faster method that can give faster solution to the problem. The aim of this paper is to introduce the 4-point block code in Majid [6] presented as in the simple form of Adams Moulton Method for solving (1) using variable step size. The method will avoid using the divided difference and integration coefficients that can be very costly. Hence, the codes will store all the coefficients of the formulae.

2 4-Point Diagonally Implicit Block Formulae

In Figure 1, the interval $[a,b]$ is divided into a series of blocks with each block containing 4 steps for the 4-point block method. The solutions of y_{n+1}, y_{n+2}, y_{n+3} and y_{n+4} at the points x_{n+1}, x_{n+2}, x_{n+3} and x_{n+4} respectively with step size h were approximated simultaneously in a block using the five back values at the points x_n, x_{n-1}, x_{n-2}, x_{n-3} and x_{n-4} of the previous block with step size rh. The method will compute four points concurrently using one earlier block that has four steps in the block. The method is called diagonally implicit because the coefficients of the upper triangular matrix entries are zero.

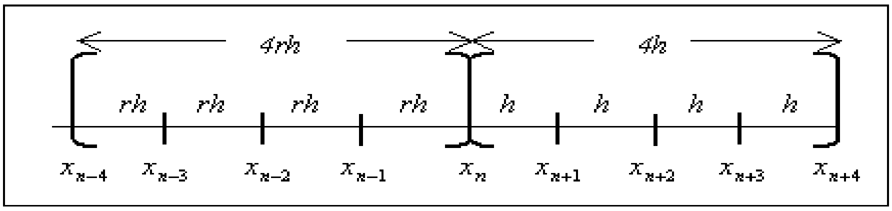

Figure 1: 4-Point 1 Block Method

The formulae of the 4-point 1 block diagonally implicit method were derived using Lagrange interpolation polynomial. The interpolation points involved for y_{n+1} are $(x_{n-4}, f_{n-4}), ..., (x_{n+1}, f_{n+1})$. The first point y_{n+1} will be derive by integrating (1) and therefore gives

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

which is equivalent to

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

The function $f(x,y)$ in (2) will be replaced by the Lagrange polynomial which interpolates at the set of points $\{x_{n-4}, x_{n-3}, x_{n-2,}x_{n-1}, x_n, x_{n+1}\}$. The procedure is by taking $s = \frac{x - x_{n+1}}{h}$; replacing $dx = h\ ds$ and changing the limit of integration from -1 to 0 in (2). Evaluating the integrals using **MAPLE**, refer to Char et al. [2] will gives the formula of the first point in terms of r as follows,

First point:

$$\begin{aligned} & y(x_{n+1}) \\ & = y(x_n) + h\left[\frac{(4r+1)\left(9r^2+8r+2\right)}{12(3r+1)(2r+1)(r+1)} f_{n+1} + \frac{\left(2+30r+175r^2+500r^3+720r^4\right)}{1440r^4} f_n \right. \\ & - \frac{(6r+1)\left(2+15r+40r^2\right)}{360r^4(r+1)} f_{n-1} + \frac{\left(2+24r+95r^2+120r^3\right)}{240r^4(2r+1)} f_{n-2} - \\ & \left. \frac{\left(2+21r+70r^2+80r^3\right)}{360r^4(3r+1)} f_{n-3} + \frac{\left(15r^2+10r+2\right)}{1440r^4} f_{n-4}\right] \end{aligned} \tag{3}$$

Now taking $x_{n+2} = x_n + 2h$ and integrating f in (2) from x_n to x_{n+2}. The interpolation points involved for the second point y_{n+2} are $(x_{n-4}, f_{n-4}), ..., (x_{n+2}, f_{n+2})$. The process is the same as the derivation of the first point. Evaluating the integrals using MAPLE gives the formula of the second point,

Second point:

$$
\begin{aligned}
& y(x_{n+2}) \\
& = y(x_n) + h\left[\frac{\left(224r + 441r^2 + 40 + 84r^4 + 350r^3\right)}{42(2r+1)(3r+2)(r+1)(r+2)} f_{n+2}\right. \\
& + \frac{8\left(8 + 56r + 147r^2 + 175r^3 + 84r^4\right)}{21(4r+1)(3r+1)(2r+1)(r+1)} f_{n+1} + \frac{\left(-140r - 245r^2 + 420r^4 - 24\right)}{1260r^4} f_n \\
& + \frac{4\left(12 + 63r + 91r^2\right)}{315r^4(r+1)(r+2)} f_{n-1} - \frac{\left(24 + 112r + 133r^2\right)}{210r^4(2r+1)(r+1)} f_{n-2} + \frac{4(7r+4)(7r+3)}{315r^4(3r+1)(3r+2)} f_{n-3} \\
& \left. - \frac{\left(24 + 84r + 77r^2\right)}{1260r^4(4r+1)(2r+1)} f_{n-4}\right]. \qquad (4)
\end{aligned}
$$

The third point y_{n+3} and the fourth point y_{n+4} will be derived similarly as the process above. The interpolation points involved for y_{n+3} and y_{n+4} are $(x_{n-4}, f_{n-4}), ..., (x_{n+3}, f_{n+3})$ and $(x_{n-4}, f_{n-4}), ..., (x_{n+4}, f_{n+4})$ respectively. Both points can be obtained by integrating (2) over the interval $[x_n, x_{n+3}]$ and $[x_n, x_{n+4}]$ respectively using **MAPLE** and the following corrector formulae of y_{n+3} and y_{n+4} in terms of r can be obtained:-

Third point:

$$
\begin{aligned}
& y(x_{n+3}) \\
& = y(x_n) + h\left[\frac{\left(261 + 588r + 392r^2 + 84r^3\right)}{112(r+1)(2r+3)(r+3)} f_{n+3}\right. \\
& + \frac{27\left(135 + 576r + 882r^2 + 560r^3 + 112r^4\right)}{448(2r+1)(3r+2)(r+1)(r+2)} f_{n+2} \\
& - \frac{9\left(297 + 1080r + 1470r^2 + 1400r^3 + 1680r^4\right)}{560(r+1)(2r+1)(3r+1)(4r+1)} f_{n+1} \\
& + \frac{\left(297 + 1080r + 1470r^2 + 1400r^3 + 1680r^4\right)}{4480r^4} f_n \\
& - \frac{9(2r+1)\left(99 + 126r + 112r^2\right)}{560r^4(r+1)(r+2)(r+3)} f_{n-1} + \frac{27\left(99 + 288r + 266r^2 + 112r^3\right)}{2240r^4(2r+1)(r+1)(2r+3)} f_{n-2} \\
& \left. - \frac{\left(297 + 756r + 588r^2 + 224r^3\right)}{560r^3(3r+1)(3r+2)(r+1)} f_{n-3} + \frac{9\left(33 + 28r + 14r^2\right)}{4480r^4(2r+1)(4r+1)} f_{n-4}\right]. \qquad (5)
\end{aligned}
$$

Fourth point:

$$
\begin{aligned}
y\left(x_{n+4}\right)=y\left(x_n\right)+h&\left[\frac{2\left(4000+10560r+9765r^2+3675r^3+441r^4\right)}{945\left(r+1\right)\left(3r+4\right)\left(r+2\right)\left(r+4\right)}f_{n+4}\right. \\
&+\frac{128\left(160+480r+510r^2+225r^3+36r^4\right)}{405\left(r+1\right)\left(2r+3\right)\left(r+3\right)\left(4r+3\right)}f_{n+3} \\
&+\frac{8\left(-320-480r+210r^2+525r^3+126r^4\right)}{315\left(r+1\right)\left(3r+2\right)\left(r+2\right)}f_{n+2} \\
&+\frac{128\left(160+480r+630r^2+525r^3+252r^4\right)}{945\left(r+1\right)\left(2r+1\right)\left(3r+1\right)\left(4r+1\right)}f_{n+1}+\frac{2\left(-224-600r-525r^2+441r^4\right)}{2835r^4}f_n \\
&+\frac{128\left(112+270r+195r^2\right)}{945r^4\left(r+1\right)\left(r+2\right)\left(r+3\right)\left(r+4\right)}f_{n-1}-\frac{8\left(224+480r+285r^2\right)}{315r^4\left(2r+1\right)\left(r+1\right)\left(2r+3\right)\left(r+2\right)}f_{n-2} \\
&+\frac{128\left(16+30r+15r^2\right)}{405r^4\left(3r+1\right)\left(3r+2\right)\left(r+1\right)\left(3r+4\right)}f_{n-3} \\
&\left.-\frac{2\left(224+360r+165r^2\right)}{945r^4\left(4r+1\right)\left(2r+1\right)\left(r+1\right)\left(4r+3\right)\left(r+1\right)}f_{n-4}\right]. \qquad (6)
\end{aligned}
$$

The predictor formulae were derived similarly and the order is one less. The step size strategy in the code is a modified version of Shampine and Gordon [10]. The choices for the next step size will be restricted to half, double or the same as the current step size. The successful step size will be allowed to double the step size at most two blocks or remain constant for at least two blocks.

In the code developed, when the next successful step size is doubled, the ratio r is 0.5 and if the next successful step size remain constant, r is 1. In case of step size failure, r is 2. Taking $r = 1, 2$ and 0.5 in (2), (3), (4) and (5) will produce the first, second, third and fourth points of the corrector formulae for the 4-point 1 block diagonally implicit method.

3 Absolute Stability

For a method to be of practical importance it must have a region of absolute stability which will ensure that the method will be able to solve at least for the mildly stiff problems. Here we will discussed the absolute stability of the 4-point 1 block diagonally implicit method (4P1DI) derived earlier using a linear first order test problem,

$$y' = f = \lambda y \qquad (7)$$

The stability region is investigated when the step size is constant, doubled and halved for the method. The test equation (7) is substituted into the corrector formulae of the block method. Setting the determinant of the corrector formulae written in matrix form to zero will give the stability polynomial. The stability polynomial of the 4-point 1 block diagonally implicit method at $r = 1, 2, 0.5$ are as follows, For $r = 1$ we have,

$$
\begin{aligned}
Q_1\left(\overline{h}\right) = {} & t^8\left(1 - \tfrac{2201909}{1814400}\overline{h} + \tfrac{30273877343}{54867456000}\overline{h}^2 - \tfrac{308040964291}{2765319782400}\overline{h}^3 + \tfrac{4660151299}{553063956480}\overline{h}^4\right) + \\
& t^7\left(-1 - \tfrac{173959}{129600}\overline{h} - \tfrac{27785856419}{3657830400}\overline{h}^2 + \tfrac{12214000212727}{8641624320000}\overline{h}^3 - \tfrac{1489558174942897}{345664972800000}\overline{h}^4\right) + \\
& t^6\left(\tfrac{524053}{362880}\overline{h} - \tfrac{94997600323}{54867456000}\overline{h}^2 - \tfrac{9280464283337}{115221657600000}\overline{h}^3 + \tfrac{18567284336021}{19203609600000}\overline{h}^4\right) + \\
& t^5\left(-\tfrac{1606484099}{10973491200}\overline{h}^2 - \tfrac{883848736217}{4320812160000}\overline{h}^3 - \tfrac{15713664812651}{172832486400000}\overline{h}^4\right) + \\
& t^4\left(-\tfrac{7006548641}{9876142080000}\overline{h}^3 - \tfrac{30541773371}{49380710400000}\overline{h}^4\right) + \tfrac{12041501}{115221657600000}\overline{h}^4 t^3 = 0
\end{aligned}
\tag{8}
$$

For $r = 2$ we have,

$$
\begin{aligned}
Q_2\left(\overline{h}\right) = {} & t^8\left(1 - \tfrac{390389}{297675}\overline{h} + \tfrac{5141883893}{8001504000}\overline{h}^2 - \tfrac{11718654239}{84015792000}\overline{h}^3 + \tfrac{49307639719}{4356374400000}\overline{h}^4\right) + \\
& t^7\left(-1 - \tfrac{5142030287}{2011806720}\overline{h} - \tfrac{13917162413647}{3621252096000}\overline{h}^2 - \tfrac{102092874793689167}{55894026101760000}\overline{h}^3 - \tfrac{890091753032073749}{479091652300800000}\overline{h}^4\right) + \\
& t^6\left(-\tfrac{200888867}{1437004800}\overline{h} - \tfrac{33176068161349}{304185176064000}\overline{h}^2 + \tfrac{1853940061285363}{42926612046151680}\overline{h}^3 + \tfrac{14185005483525081361}{16097479517306880000}\right) + \\
& t^5\left(-\tfrac{38171926871}{12167407042560}\overline{h}^2 - \tfrac{20527852145506099}{7154435341025280000}\overline{h}^3 - \tfrac{65818261516994621}{9198559724175360000}\overline{h}^4\right) + \\
& t^4\left(-\tfrac{244509809187731}{21463306023075840000}\overline{h}^3 - \tfrac{4534234387582093}{64389918069227520000}\overline{h}^4\right) - \tfrac{297160486963}{64389918069227520000}\overline{h}^4 t^3 = 0
\end{aligned}
\tag{9}
$$

Finally, for $r = 0.5$ we have,

$$
\begin{aligned}
Q_3\left(\overline{h}\right) = {} & t^8\left(1 - \tfrac{351373483}{314344800}\overline{h} + \tfrac{288670408457}{616115808000}\overline{h}^2 - \tfrac{1355129070299}{15526118361600}\overline{h}^3 + \tfrac{21513951671}{3528663264000}\overline{h}^4\right) + \\
& t^7\left(-1 + \tfrac{4886737}{352800}\overline{h} - \tfrac{224869887833089}{5545042272000}\overline{h}^2 + \tfrac{8908191874265417}{317579693760000}\overline{h}^3 - \tfrac{394216564731636641}{262003247352000000}\overline{h}^4\right) + \\
& t^6\left(-\tfrac{164377762}{9823275}\overline{h} - \tfrac{2179871724829}{115521714000}\overline{h}^2 - \tfrac{4188043185673}{108301606875}\overline{h}^3 + \tfrac{2325175906995547}{145557359640000}\overline{h}^4\right) + \\
& t^5\left(-\tfrac{32028975326}{3094331625}\overline{h}^2 - \tfrac{119275461129926}{6823001233125}\overline{h}^3 - \tfrac{43616723698406}{4093800739875}\overline{h}^4\right) + \\
& t^4\left(-\tfrac{227946967904}{758111248125}\overline{h}^3 - \tfrac{674614306972}{2274333744375}\overline{h}^4\right) + \tfrac{42830214016}{102345018496875}\overline{h}^4 t^3 = 0
\end{aligned}
\tag{10}
$$

where $\overline{h} = h\lambda$ and the stability regions are shown in Figure 2, 3 and 4 respectively.

The stability region is inside the boundary of the dotted points. The stability region is larger when the step size is half ($r = 2$) compared to the step size being double ($r = 0.5$) or constant ($r = 1$). This is expected because the region should get larger with smaller step sizes. The smallest stability region is when the step size being double for the method.

4 Numerical Results

In order to study the efficiency of the new method, we present some numerical experiments for the following problems:

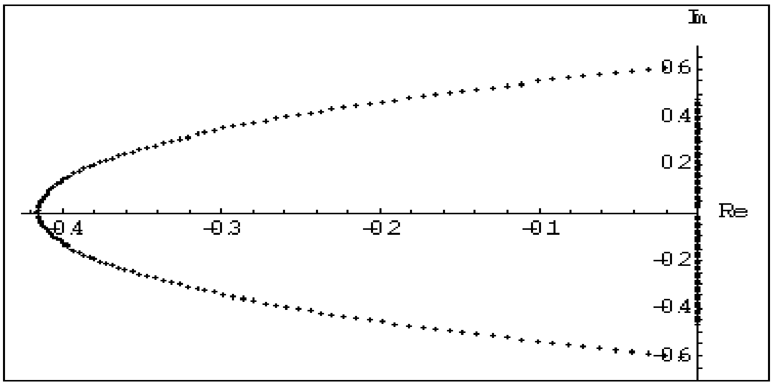

Figure 2: Stability Region for 4-Point 1 Block Diagonally Implicit Method when $r = 1$

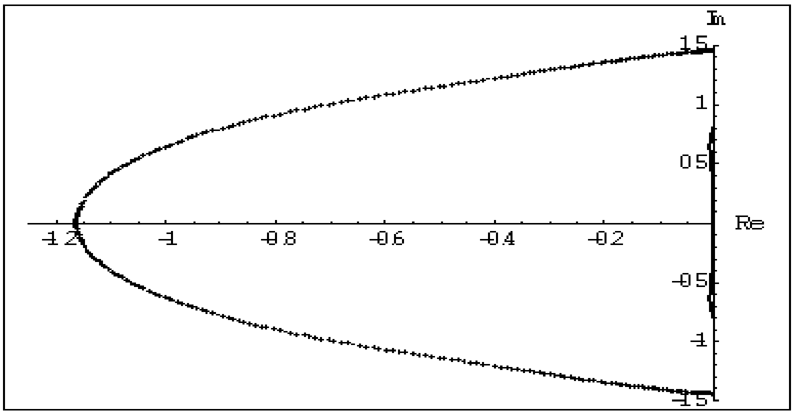

Figure 3: Stability Region for 4-Point 1 Block Diagonally Implicit Method when $r = 2$

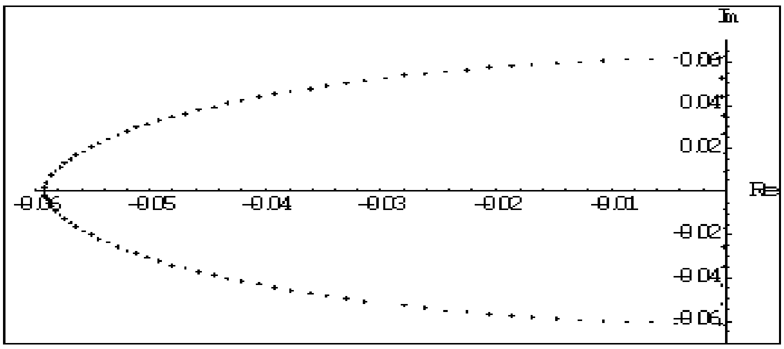

Figure 4: Stability Region for 4-Point 1 Block Diagonally Implicit Method when $r = 0.5$

Problem 1: Negative exponential problem. (Mildly stiff)
$y_1' = -0.5y_1$,
$y_1(0) = 1$,
$x \in [0, 20]$
Exact Solution: $y_1(x) = e^{-0.5x}$
Source: Burden [1]

Problem 2: Nonlinear Krogh's problem (Non stiff)
$y_i' = -\beta_i y_i + y_i^2$ $i = 1, 2, 3, 4$ $y_i(0) = -1$, $x \in [0, 20]$,
$\beta_1 = 0.2, \beta_2 = 0.2, \beta_3 = 0.3, \beta_4 = 0.4$
Exact Solution: $y_i(x) = \dfrac{\beta_i}{1 + c_i e^{\beta_i x}}$, $c_i = -(1 + \beta_i)$
Source: Johnson and Barney [5]

Problem 3: A two-body orbit problem (Mildly stiff)
$y_1' = y_3$, $y_2' = y_4$, $y_3' = -\dfrac{y_1}{r^3}$, $y_4' = -\dfrac{y_2}{r^3}$, $r = \sqrt{y_1^2 + y_2^2}$, $x \in [0, 20]$
$y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 0$, $y_4(0) = 1$,
Exact Solution: $y_1(x) = \cos x$, $y_2(x) = \sin x$, $y_3(x) = -\sin x$,
$y_4(x) = \cos x$.
Source: Hairer, et al. [4]

The following notations are used in the tables:

TOL	Tolerance
MTD	Method employed
TS	Total steps taken
FS	Total failure steps
MAXE	Magnitude of the maximum error of the computed solution
RSTEP	The ratio steps of 4P1DI compared to 1PVSO
1PVSO	Implementation of the 1-point implicit method (non block) using variable step and order in Omar [7]
4P1DI	Implementation of the 4 point 1 block diagonally implicit method using variable step size

The errors calculated are defined as

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

where $(y)_t$ is the t- th component of the approximate y. $A = 1$, $B = 0$ corresponds to the absolute error test, $A = 1$, $B = 1$ corresponds to the mixed test and finally $A = 0$, $B = 1$ corresponds to the relative error test. The mixed error test is used for all the above problems.

The maximum error is defined as follows:-

$$\text{MAXE} = \max_{1 \le i \le SSTEP} \left(\max_{1 \le i \le N} (E_i)_t \right)$$

where N is the number of equations in the system and $SSTEP$ is the number of successful steps. In the code, we iterate the corrector to convergence. The convergence test employed were

$$\left\| y_{n+4}^{(s+1)} - y_{n+4}^{(s)} \right\| < 0.1 \times \text{TOL}$$

and s is the number of iterations. The error controls for the code was at the fourth point in the block because in general it had given us better results.

Table 1: Numerical results of 1PVSO and 4P1DI Methods for Solving Problem 1

TOL	MTD	TS	FS	MAXE	RSTEP
10^{-2}	1PVSO	25	0	1.7373e-3	1.47
	4P1DI	17	0	3.5471e-4	1.00
10^{-4}	1PVSO	37	0	4.6562e-5	1.85
	4P1DI	20	0	7.6178e-5	1.00
10^{-6}	1PVSO	71	0	2.0686e-7	2.54
	4P1DI	28	0	7.0034e-8	1.00
10^{-8}	1PVSO	146	0	2.0533e-9	4.17
	4P1DI	35	0	3.3234e-9	1.00
10^{-10}	1PVSO	351	0	1.1537e-11	7.16
	4P1DI	49	0	1.2141e-10	1.00

Table 2: Numerical results of 1PVSO and 4P1DI Methods for Solving Problem 2

TOL	MTD	TS	FS	MAXE	RSTEP
10^{-2}	1PVSO	31	0	2.5451e-3	1.24
	4P1DI	25	0	2.2388e-4	1.00
10^{-4}	1PVSO	49	0	3.6125e-5	1.32
	4P1DI	37	0	1.3158e-6	1.00
10^{-6}	1PVSO	86	0	1.4942e-7	2.15
	4P1DI	40	0	1.2305e-8	1.00
10^{-8}	1PVSO	213	0	2.6028e-9	3.67
	4P1DI	58	0	2.5485e-10	1.00
10^{-10}	1PVSO	505	0	2.7385e-11	5.74
	4P1DI	88	0	6.6143e-12	1.00

Table 3: Numerical results of 1PVSO and 4P1DI Methods for Solving Problem 3

TOL	MTD	TS	FS	MAXE	RSTEP
10^{-2}	1PVSO	44	0	9.9994e-1	1.42
	4P1DI	31	0	1.3095e-2	1.00
10^{-4}	1PVSO	83	0	5.1513e-3	2.08
	4P1DI	40	0	5.6526e-4	1.00
10^{-6}	1PVSO	181	0	1.3466e-3	3.77
	4P1DI	48	0	2.0997e-5	1.00
10^{-8}	1PVSO	434	0	1.2342e-5	4.62
	4P1DI	94	0	4.6609e-8	1.00
10^{-10}	1PVSO	1044	0	1.4101e-7	9.00
	4P1DI	116	0	9.9087e-9	1.00

The codes were written in C language and executed on DYNIX/ptx operating system. The results for the three given problems when solved using the 1PVSO and 4P1DI are presented in the tables below.

From Table 1 through 3, it is observed that the method 4P1DI required less number of steps compared to the method 1PVSO when solving the same given problems. These results are expected since the four point block method would approximate the solutions at four points simultaneously. The ratios of steps (RSTEP) are greater than one show that 4P1DI is efficient compared to 1PVSO. In fact, in most cases the ratios are greater than three, which indicates a clear advantage of method 4P1DI over 1PVSO. In terms of maximum error, method 4P1DI is comparable or better compared to 1PVSO.

5 Conclusion

In this paper, we have shown the efficiency of the developed 4-point block method presented as in the simple form of Adams Moulton Method using variable step size is suitable for solving ODEs. The method has shown the superiority in terms of total steps and maximum error over the non block method 1PVSO.

References

[1] R.L. Burden and J.D.Faires, *Numerical Analysis, Seventh Edition,* Wadsworth Group. Brooks/Cole, (2001), 342-353.

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

[3] M.T. Chu, and H. Hamilton, *Parallel Solution of ODE's by Multiblock Methods,* SIAM J. Sci Stat. Comput. 8 (1987), 342-354.

[4] E.Hairer, S.P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I,* Berlin: Springer-Verlag, 1993.

[5] A.I. Johnson, and J.R. Barney, *Numerical Solution of Large Systems of Stiff Ordinary Differential Equations in a Modular Simulation Framework.* In: L.Lapidus and W.E.Schiesser, *Numerical Methods For Differential Systems ed.*, Academic Press Inc, New York , pp 97-124, 1976.

[6] Z.A.Majid, *Parallel Block Methods For Solving Ordinary Differential Equations*, Ph.D. Thesis, University Putra Malaysia, Malaysia, 2004.

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

[8] J.B.Rosser, *Rungge Kutta For All Season,* SIAM Rev., Vol 9, (1976), 417-452.

[9] L.F.Shampine, and H.F.Watts, *Block Implicit One-Step Methods,* Math. Comp. 23, (1969), 731-740.

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

[11] 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, 1979.

[12] P.B.Worland, *Parallel Method for the Numerical Solution of Ordinary Differential Equations*, IEEE Transaction on Computer, (1976) 1045-1048.