

Numerical Solution of Higher Order Ordinary Differential Equations by Direct Block Code

¹Waeleh, N., ²Z.A. Majid, ³F. Ismail and ⁴M. Suleiman

^{1,2,3,4}Department of Mathematics, Faculty of Science,

^{2,3,4}Institute for Mathematical Research,

University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract: Problem statement: This study is concerned with the development of a code based on 2-point block method for solving higher order Initial Value Problems (IVPs) of Ordinary Differential Equations (ODEs) directly. **Approach:** The block method was developed based on numerical integration and using interpolation approach which is similarly as Adams Moulton type. Furthermore, the proposed method is derived in order to solve higher order ODEs in a single code using variable step size and implemented in a predictor corrector mode. This block method will act as simultaneous numerical integrator by computing the numerical solution at two steps simultaneously. **Results:** The numerical results for the direct block method were superior compared to the existing block method. **Conclusion:** It is clearly proved that the code is able to produce good results for solving higher order ODEs.

Key words: Higher order ODEs, variable step size, predictor corrector, block method

INTRODUCTION

The mathematical formulation of physical phenomena in science and engineering often leads to IVPs of ODEs. This type of problem can be formulated either in terms of first order ODEs or higher order ODEs. For instance, this application will often used in beam theory, electric circuits, control theory, mechanical system and celestial mechanics. Thus, this study will concern on solving directly higher order non-stiff IVPs of ODEs of the form:

$$\begin{aligned} y^d &= f(x, y, \dots, y^{d-1}), \\ y^i(a) &= \eta_i, 0 \leq i \leq d-1, \quad x \in [a, b] \end{aligned} \quad (1)$$

The conventional methods of solving higher order ODEs will reduce such problems to a system of first order equations. This approach is cumbersome and will increase computational time as well as consume a lot of human effort. Thus, several researchers have concerned themselves with study to solve Eq. 1 directly such as Awoyemi (2003); Majid (2004); Awoyemi (2005); Majid and Suleiman (2006); Jator (2010); Jain *et al.* (1977) Kayode and Awoyemi (2010). Awoyemi (2005) has proposed a multiderivative collocation method for direct solution of fourth order IVPs of ODEs while Majid and Suleiman (2006) have introduced a direct

integration implicit variable step method for solving higher order systems of ODEs. Majid (2004) has developed the 2-point block method for solving first and second order ODEs using variable step size. It was noted that Jator (2010) had used the application of a self starting linear multistep method for solving second order IVPs directly.

Block methods for numerical solution of higher order ODEs have been proposed by several researchers such as Chu and Hamilton (1987); Fatunla (1991) and Jator (2010). Chu and Hamilton (1987) have proposed multi-block methods for parallel solution of ODEs and Fatunla (1991) has presented a zero stable block method for second order ODEs. The uniqueness of block method is that in each application, the solution value will be computed simultaneously at several distinct points. There are several existence numerical methods for handling higher order ODEs directly but those methods will compute the numerical solutions at one point sequentially. Henceforth, we need a method that can give faster solution of the problem.

In this study, we are going to extend the study done in Majid (2004) by implemented the 2-point block method to solve ODEs up to order five in a single code.

MATERIALS AND METHODS

Formulation of the method: In 2-point block method, the closed finite interval [a,b] is divided into a series of

Corresponding Author: Waeleh, N., Department of Mathematics, Faculty of Science, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

blocks that contained the interpolation points involved in the derivation of the block method.

According to Fig. 1, the method will generate the numerical solution at two points simultaneously. The values of y_{n+1} and y_{n+2} at the points x_{n+1} and x_{n+2} respectively are simultaneously computed in a block with step size h using the same back values which is the values at the point x_n, x_{n-1} and x_{n-2} with step size rh . The formulae of 2-point block method are derived by integrating Eq. 1 d times as follows Eq. 2:

Let $x_{n+v} = x_n + vh$, where $v = 1$ or 2 .

$$\int_{x_n}^{x_{n+v}} \int_{x_n}^x \dots \int_{x_n}^x y^d(x, y, \dots, y^{d-1}) dx dx \dots dx = \int_{x_n}^{x_{n+v}} \int_{x_n}^x \dots \int_{x_n}^x f(x, y, \dots, y^{d-1}) dx dx \dots dx \quad (2)$$

which leads to the general formula below:

$$y_{n+v}^{d-g} = \sum_{k=0}^{g-1} \frac{(vh)^k}{k!} y_n^{(d-g+k)} + \int_{x_n}^{x_{n+v}} \frac{(x_{n+v} - x)^g}{(g-1)!} f(x, y, \dots, y^{d-1}) dx \quad (3)$$

where, g is the number of times which Eq. 3 is integrated over the corresponding interval. Lagrange interpolation polynomial is used to approximate the function of $f(x, y, \dots, y^{d-1})$ in Eq. 3 and the interpolation points involved for the corrector formulae are $(x_{n-2}, f_{n-2}), \dots, (x_{n+2}, f_{n+2})$. Let $s = \frac{x - x_{n+2}}{h}$ and $dx = hds$ will be substitute into Eq. 3. By taking $d = 5$ in Eq. 1, the approximate solution of y_{n+1} and y_{n+2} will be obtained by integrating Eq. 1 once, twice, thrice, four times and five times over the interval $[x_n, x_{n+1}]$ and $[x_n, x_{n+2}]$ respectively. Finally, this integral will be evaluated using MAPLE and the corrector formulae in terms of r will be obtained.

The same approaches were employed in the derivation of the predictor formulae for $y, y', y'', y''', y^{(iv)}$ at the points x_{n+1} and x_{n+2} respectively and the interpolation points involved are $(x_{n-3}, f_{n-3}), \dots, (x_n, f_n)$. Hence, it will produce the predictor formulae in terms of q and r . For the sake of simplification, the general corrector formula of 2-point block method is developed in the manner shown in Eq. 4:

$$y_{n+v}^{d-g} = \sum_{k=0}^{g-1} \frac{(vh)^k}{k!} y_n^{(d-g+k)} + \frac{h^g}{(g-1)!} \sum_{j=-2}^2 \beta_{v,j}^g f_{n+j} \quad (4)$$

$\beta_{v,j}^g$ in Eq. 4 stands for the coefficients of the formulae and tabulated in Table 1-3 for $r = 1, r = 2$ and $r = 0.5$.

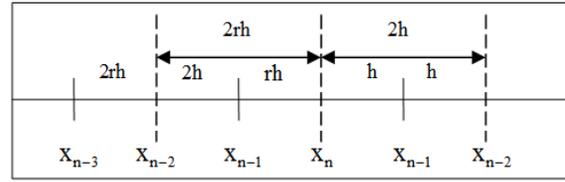


Fig. 1: 2-point block method

Table 1: Coefficients when $r = 1$

$r=1$	$\beta_{i,-2}^g$	$\beta_{i,-1}^g$	$\beta_{i,0}^g$	$\beta_{i,1}^g$	$\beta_{i,2}^g$
$g = 1$	$\frac{11}{720}$	$-\frac{74}{720}$	$\frac{456}{720}$	$\frac{346}{720}$	$-\frac{19}{720}$
$g = 2$	$\frac{11}{1440}$	$-\frac{76}{1440}$	$\frac{582}{1440}$	$\frac{220}{1440}$	$-\frac{17}{1440}$
$g = 3$	$\frac{23}{10080}$	$-\frac{162}{10080}$	$\frac{1482}{10080}$	$\frac{370}{10080}$	$-\frac{33}{10080}$
$g = 4$	$\frac{61}{120960}$	$-\frac{436}{120960}$	$\frac{4638}{120960}$	$\frac{860}{120960}$	$-\frac{83}{120960}$
$g = 5$	$\frac{65}{725760}$	$-\frac{470}{725760}$	$\frac{5700}{725760}$	$\frac{838}{725760}$	$-\frac{85}{725760}$
$r=1$	$\beta_{2,-2}^g$	$\beta_{2,-1}^g$	$\beta_{2,0}^g$	$\beta_{2,1}^g$	$\beta_{2,2}^g$
$g = 1$	$-\frac{1}{90}$	$\frac{4}{90}$	$\frac{24}{90}$	$\frac{124}{90}$	$\frac{29}{90}$
$g = 2$	$\frac{1}{90}$	$-\frac{8}{90}$	$\frac{78}{90}$	$\frac{104}{90}$	$\frac{5}{90}$
$g = 3$	$\frac{9}{630}$	$-\frac{64}{630}$	$\frac{516}{630}$	$\frac{384}{630}$	$-\frac{5}{630}$
$g = 4$	$\frac{16}{1890}$	$-\frac{112}{1890}$	$\frac{912}{1890}$	$\frac{464}{1890}$	$-\frac{20}{1890}$
$g = 5$	$\frac{80}{22680}$	$-\frac{560}{22680}$	$\frac{4800}{22680}$	$\frac{1840}{22680}$	$-\frac{112}{22680}$

The order of this developed method is calculated in a block form as proposed by Fatunla (1991). The 2-point block method for ODEs can be written in a matrix differentiation Eq. 5 below:

$$\alpha Y_m = h\beta Y_m' + h^2\lambda Y_m'' + \dots + h^d \xi F_m \quad (5)$$

where, α, β, λ and ξ are the coefficients of the 2-point block method. By applying the formulae for the constants C_q in Fatunla (1991), the order and error constant of the method will be obtained and the formulae are defined by Eq. 6:

$$C_0 = \sum_{j=0}^k \alpha_j$$

$$C_1 = \sum_{j=0}^k (j\alpha_j - \beta_j)$$

$$C_2 = \sum_{j=0}^k \left(\frac{j^2}{2!} \alpha_j - j\beta_j - \lambda_j \right)$$

$$\vdots$$

$$\vdots$$

$$C_q = \sum_{j=0}^k \left(\frac{j^q}{q!} \alpha_j - \frac{j^{q-1}}{(q-1)!} \beta_j - \dots - \frac{j^1}{(1)!} \xi_j \right) \quad (6)$$

Table 2: Coefficients when r = 2

r = 2	$\beta_{1,-2}^g$	$\beta_{1,-1}^g$	$\beta_{1,0}^g$	$\beta_{1,1}^g$	$\beta_{1,2}^g$
g = 1	$\frac{37}{14400}$	$\frac{335}{14400}$	$\frac{7455}{14400}$	$\frac{7808}{14400}$	$\frac{565}{14400}$
g = 2	$\frac{19}{14400}$	$\frac{175}{14400}$	$\frac{4965}{14400}$	$\frac{2656}{14400}$	$\frac{265}{14400}$
g = 3	$\frac{81}{201600}$	$\frac{755}{201600}$	$\frac{25995}{201600}$	$\frac{9344}{201600}$	$\frac{1065}{201600}$
g = 4	$\frac{109}{1209600}$	$\frac{1025}{1209600}$	$\frac{41475}{1209600}$	$\frac{11216}{1209600}$	$\frac{1375}{1209600}$
g = 5	$\frac{47}{2903040}$	$\frac{445}{2903040}$	$\frac{20685}{2903040}$	$\frac{4480}{2903040}$	$\frac{575}{2903040}$
r = 2	$\beta_{2,-2}^g$	$\beta_{2,-1}^g$	$\beta_{2,0}^g$	$\beta_{2,1}^g$	$\beta_{2,2}^g$
g = 1	$\frac{1}{900}$	$\frac{5}{900}$	$\frac{285}{900}$	$\frac{1216}{900}$	$\frac{295}{900}$
g = 2	$\frac{1}{450}$	$\frac{10}{450}$	$\frac{345}{450}$	$\frac{544}{450}$	$\frac{20}{450}$
g = 3	$\frac{8}{3150}$	$\frac{75}{3150}$	$\frac{2220}{3150}$	$\frac{2112}{3150}$	$\frac{65}{3150}$
g = 4	$\frac{14}{9450}$	$\frac{130}{9450}$	$\frac{3930}{9450}$	$\frac{2656}{9450}$	$\frac{170}{9450}$
g = 5	$\frac{14}{22680}$	$\frac{130}{22680}$	$\frac{4170}{22680}$	$\frac{2176}{22680}$	$\frac{182}{22680}$

Table 3: Coefficients when r=0.5

r = 0.5	$\beta_{1,-2}^g$	$\beta_{1,-1}^g$	$\beta_{1,0}^g$	$\beta_{1,1}^g$	$\beta_{1,2}^g$
g = 1	$\frac{145}{1800}$	$\frac{704}{1800}$	$\frac{1635}{1800}$	$\frac{755}{1800}$	$\frac{31}{1800}$
g = 2	$\frac{70}{1800}$	$\frac{352}{1800}$	$\frac{975}{1800}$	$\frac{220}{1800}$	$\frac{13}{1800}$
g = 3	$\frac{285}{25200}$	$\frac{1472}{25200}$	$\frac{4740}{25200}$	$\frac{695}{25200}$	$\frac{48}{25200}$
g = 4	$\frac{185}{75600}$	$\frac{976}{75600}$	$\frac{3585}{75600}$	$\frac{385}{75600}$	$\frac{29}{75600}$
g = 5	$\frac{155}{362880}$	$\frac{832}{362880}$	$\frac{3435}{362880}$	$\frac{289}{362880}$	$\frac{23}{362880}$
r = 0.5	$\beta_{2,-2}^g$	$\beta_{2,-1}^g$	$\beta_{2,0}^g$	$\beta_{2,1}^g$	$\beta_{2,2}^g$
g = 1	$\frac{20}{225}$	$\frac{64}{225}$	$\frac{15}{225}$	$\frac{320}{225}$	$\frac{71}{225}$
g = 2	$\frac{10}{225}$	$\frac{64}{225}$	$\frac{240}{225}$	$\frac{250}{225}$	$\frac{14}{225}$
g = 3	$\frac{220}{3150}$	$\frac{1152}{3150}$	$\frac{3390}{3150}$	$\frac{1740}{3150}$	$\frac{2}{3150}$
g = 4	$\frac{400}{9450}$	$\frac{2048}{9450}$	$\frac{6000}{9450}$	$\frac{2000}{9450}$	$\frac{52}{9450}$
g = 5	$\frac{400}{22680}$	$\frac{2048}{22680}$	$\frac{6240}{22680}$	$\frac{1520}{22680}$	$\frac{64}{22680}$

Table 4: Error constant for corrector formulae when r=1

D	1	2	3	4	5
C_{p+d}	$\begin{bmatrix} 11 \\ 1440 \\ -1 \\ 90 \end{bmatrix}$	$\begin{bmatrix} 37 \\ 10080 \\ 1 \\ 315 \end{bmatrix}$	$\frac{367}{20160}$	$\begin{bmatrix} 419 \\ 302400 \\ 16 \\ 675 \end{bmatrix}$	$\begin{bmatrix} 293 \\ 302400 \\ 188 \\ 4725 \end{bmatrix}$

The method is of order p if $C_q = 0$, $q = 0(1) p + d - 1$, $C_{p+d} \neq 0$. Thus, by implementing this approach to the 2-point block method, we found that the predictor is of order four and corrector is of order five. The error constant for the corrector formulae when $r = 1$ will be in matrix form as shown in Table 4.

Implementation of the method: A single code of the PECE scheme has been implemented with variable step size to study the computational time and human effort saving in using a direct integration method. The developed code starts by calculating the initial step size then finding the initial points in the starting block of the method. In order to evaluate the initial three starting points, the Euler method was adopted in the code as a generator of the method. Hence, the Euler method will be used only once at the beginning of the code. Once the points for starting block are calculated, then the block method can be applied until the end of interval.

A test for checking the end of the interval is made in order to reach the end of the interval precisely and it will be functional at each step of integration. The strategy is by limiting the choices of the next step size to half, double or remains constant as the previous step size. At each step of the integration, if the approximated solutions fulfilled the desired accuracy, therefore the step is called as successive step.

Hence, the choices for the next step size will be doubled or constant which specified by step size controller. Otherwise the step is called failure step and the next step size becomes half.

The possible ratios for the next constant step size are $(r = 1, q = 1)$, $(r = 1, q = 2)$ and $(r = 1, q = 0.5)$. At each doubled step size the ratios are $(r = 0.5, q = 0.5)$ and $(r = 0.5, q = 0.25)$. In the case of failure step size, the ratio is $(r = 2, q = 2)$. A step failure happens due to the Local Truncation Error (LTE) exceeding the given tolerance. This corrector formulae show that, the code consist the formula of y, y', y'', y''' and $y_{(iv)}$. Hence, the developed algorithm could be use for solving first order up to fifth order problem of ODEs directly in a single code. The algorithm was written in C language.

The approximation value of y_{n+1} and y_{n+2} are using predictor-corrector mode, $(P_k E) (C_{k+1} E)^u$ where P_k and C_{k+1} indicate the predictor of order k and corrector of order k+1 respectively and E indicate the evaluation of

the function. In the code, the corrector will be iterated until it is converge and the convergence test employed was Eq. 7:

$$\left| y_{n+2}^{(u)} - y_{n+2}^{(u-1)} \right| < 0.1 \times \text{TOL} \quad (7)$$

where, u is the number of iterations.

The LTE will be obtained by comparing the absolute difference of the corrector formula derived of order k with the similar corrector formula of order k+1 at the point x_{n+2} . If the $\text{LTE} < \text{TOL}$, hence the successful step achieved and the next step size will be obtained using the step size increment formula as follows Eq. 8:

$$h_{\text{new}} = \delta \times h_{\text{old}} \times \left(\frac{\text{TOL}}{\text{LTE}} \right)^{\frac{1}{k}} \quad (8)$$

Where:

δ = A safety factor

k = The order of the corrector formulae while h_{new} and h_{old} are the step size for current and previous block respectively

Finally, the errors in the code are calculated by the difference between an estimation value y_{n+2} resulting from using 2-point block method and the exact solution. The formula was defined as Eq. 9:

$$(e_i)_t = \left| \frac{(y_i)_t - (y(x_i))_t}{A + B(y(x_i))_t} \right| \quad (9)$$

where, $(y_i)_t$ is the t-th component of the approximation y. A = 1, B = 0 referred to the absolute error test, while A = 1, B = 1 will be used for mixed error test and A = 0, B = 1 correspond to the relative error test.

RESULTS

The results of numerical tests will be presented in order to illustrate the performance of 2-point block method. The following notations are used in the table:

TOL:	Tolerance
MTD:	Method used
TS:	Total steps taken
FS:	Total failure step
FCN:	Total function calls
MAXE:	Magnitude of maximum error of the computed solution
TIME:	The execution time taken in microseconds

2PRVS: Algorithm of 2-point fully implicit block method by reducing the problem to first order ODEs in Majid (2004)

2PDVS: Implementation of 2-point block method in this study by solving the problem directly

Problem 1:

$$y_1'' = -y_2', y_2'' = -y_1', \quad x \in [0,10]$$

$$y_1(0) = 0, y_1'(0) = \frac{1}{1-e^{-1}}, y_2(0) = 1, y_2'(0) = \frac{1}{1-e^{-1}}$$

Solution: $y_1(x) = \frac{1-e^{-x}}{1-e^{-1}}, y_2(x) = \frac{2-e^{-1}-e^{-x}}{1-e^{-1}}$

Problem 2:

$$y''' = -4y' + x, \quad x \in [0,4\pi]$$

$$y(0) = 0, y'(0) = 0, y''(0) = 1$$

Solution: $y(x) = \left(\frac{3}{16} \right) (1 - \cos 2x) + \frac{1}{8} x^2$.

Problem 3:

$$y^{(iv)} = -xy - (8 + 7x + x^3)e^x, \quad x \in [0,20]$$

$$y(0) = 0, y'(0) = 1, y''(0) = 0, y'''(0) = 3$$

Solution: $y(x) = x(1-x)e^x$.

Problem 4:

$$y^{(v)} = 2y'y'' - yy^{(iv)} - y'y''' - 8x + (x^2 - 2x - 3)e^x, \quad x \in [0,2]$$

$$y(0) = 1, y'(0) = 1, y''(0) = 3, y'''(0) = 1, y^{(iv)}(0) = 1$$

Solution: $y(x) = e^x + x^2$.

DISCUSSION

We are going to compare the numerical results obtained by solving the tested problems using a direct integration approach with the results using a reduction approach. It is clearly shown that the 2PDVS has superiority in terms of computational time and total number of steps taken especially as the tolerance getting smaller. This indicates the major advantage of direct integration method compared to reduce to d-th order equation to d sets of first order equations. Table 5-8 show the maximum error of 2PRVS is better and comparable compared to 2PDVS.

Table 5: Numerical results for solving Problem 1

TOL	MTD	TS	FS	FCN	MAXE	TIME
	2PRVS	18	0	122	1.7336(-4)	1264
10^{-2}	2PDVS	17	0	100	1.0051(-3)	1058
10^{-4}	2PDVS	29	0	189	2.3232(-6)	2829
10^{-6}	2PRVS	25	0	151	1.2155(-5)	1917
10^{-8}	2PRVS	55	0	345	1.3735(-8)	4947
	2PDVS	38	0	235	4.6491(-7)	2839
10^{-10}	2PRVS	114	0	719	1.7562(-10)	9205
	2PDVS	65	0	403	1.3722(-8)	4255
	2PRVS	261	0	1603	2.1100(-12)	18338
	2PDVS	119	0	743	4.1326(-10)	6875

Table 6: Numerical results for solving Problem 2

TOL	MTD	TS	FS	FCN	MAXE	TIME
10^{-2}	2PRVS	63	0	314	3.3578(-3)	2121
10^{-4}	2PDVS	36	0	158	8.3249(-4)	997
	2PRVS	133	0	685	3.8276(-5)	5590
10^{-6}	2PDVS	54	0	251	1.4502(-5)	1865
	2PRVS	319	0	1647	3.7843(-7)	13019
10^{-8}	2PDVS	100	0	425	6.9612(-7)	2884
	2PRVS	446	0	2601	5.9506(-9)	24255
10^{-10}	2PDVS	173	0	767	9.1414(-9)	4525
	2PRVS	1096	0	6423	3.0347(-11)	51681
	2PDVS	289	0	1365	3.3935(-10)	7206

Table 7: Numerical results for solving Problem 3

TOL	MTD	TS	FS	FCN	MAXE	TIME
10^{-2}	2PRVS	120	0	492	2.0093(-4)	4014
	2PDVS	48	0	204	1.2153(-4)	1228
10^{-4}	2PRVS	288	0	1167	2.9049(-6)	12263
	2PDVS	83	0	345	8.2525(-7)	2510
10^{-6}	2PRVS	704	0	2841	3.5053(-8)	31563
	2PDVS	145	0	593	1.7508(-7)	4365
10^{-8}	2PRVS	1742	0	7033	3.5724(-10)	78949
	2PDVS	234	0	949	4.3465(-9)	6969
10^{-10}	2PRVS	4333	0	17501	1.7159(-11)	196775
	2PDVS	457	0	1841	4.3459(-10)	11820

Table 8: Numerical results for solving Problem 4

TOL	MTD	TS	FS	FCN	MAXE	TIME
10^{-2}	2PRVS	17	0	76	4.4178(-4)	1165
	2PDVS	12	0	60	4.3851(-6)	844
10^{-4}	2PRVS	29	0	125	1.9629(-6)	2594
	2PDVS	17	0	81	4.0341(-6)	1543
10^{-6}	2PRVS	57	0	237	1.4320(-8)	4590
	2PDVS	22	0	103	1.7326(-6)	2067
10^{-8}	2PRVS	122	0	497	1.5090(-10)	8545
	2PDVS	26	0	121	4.7280(-7)	3025
10^{-10}	2PRVS	283	0	1141	1.5924(-12)	17440
	2PDVS	40	0	173	7.5175(-9)	3742

Nevertheless, 2PDVS is still able to obtain the desired accuracy within the given tolerance. Furthermore, 2PDVS is economical to be implemented than 2PRVS since the number of functions to be evaluated by 2PDVS is much lesser than 2PRVS especially at finer tolerance. Therefore, by solving higher order ODEs directly will reduce the computational cost and will inevitable effects on the execution time.

CONCLUSION

In this study, we have shown the efficiency of the developed two point block method presented as in the form of Adams Moulton method using variable step size which is suitable for solving higher order ODEs directly.

ACKNOWLEDGMENT

The researchers would like to thank the Ministry of Science, Technology and Innovation (MOSTI) for their National Science Fellowship (NSF) scholarship.

REFERENCES

- Awoyemi, D.O., 2003. A P-stable linear multistep method for solving general third order ordinary differential equations. *Int. J. Comput. Math.*, 80: 985-991. DOI: 10.1080/0020716031000079572
- Awoyemi, D.O., 2005. Algorithmic collocation approach for direct solution of fourth-order initial-value problems of ordinary differential equations. *Int. J. Comput. Math.*, 82: 321-329. DOI: 10.1080/00207160412331296634
- Chu, M.T. and H. Hamilton, 1987. Parallel solution of ODE's by multiblock methods. *SIAM J. Sci. Stat. Comput.*, 8: 342-353. DOI: 10.1137/0908039
- Fatunla, S.O., 1991. Block methods for second order ODEs. *Int. J. Comput. Math.*, 41: 55-63. DOI: 10.1080/00207169108804026
- Jain, M.K., S.R.K. Iyengar and J.S.V. Saldanha, 1977. Numerical solution of a fourth-order ordinary differential equation. *J. Eng. Math.*, 11: 373-380. DOI: 10.1007/BF01537095
- Jator, S.N., 2010. Solving second order initial value problems by a hybrid multistep method without predictor. *Applied Math. Comput.*, 217: 4036-4046. DOI: 10.1016/j.amc.2010.10.010
- Kayode, S.J. and D.O. Awoyemi, 2010. A multiderivative collocation method for 5th order ordinary differential equations. *J. Math. Stat.*, 6: 60-63. DOI: 10.3844/jmssp.2010.60.63
- Majid, Z.A., 2004. Parallel block methods for solving ordinary differential equations. Ph.D. Thesis, University Putra Malaysia.
- Majid, Z.A. and M.B. Suleiman, 2006. Direct integration implicit variable steps method for solving higher order systems of ordinary differential equations directly. *Sains Malaysiana*, 35: 63-68.