Matematika, 2003, Jilid 19, Bil. 2, hlm. 91–100

2-Point Implicit Block One-Step Method Half Gauss-Seidel For Solving First Order Ordinary Differential Equations

Zanariah bt Abdul Majid[1], **Mohamed bin Suleiman**[1], **Fudziah bt Ismail**[1] & **Mohamed bin Othman**[2].
Mathematics Department[1],
Faculty of Science and Environment Studies,
Communication Technology and Network Department[2],
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia,
43400 UPM, Serdang, Malaysia.

Abstract In this paper, a 2 point implicit block one-step method for solving a system of ordinary differential equations (ODEs) using variable step size is developed. This method will estimate the solutions of initial value problems at 2 point simultaneously based on equidistant block method. Numerical results are given to compare the efficiency of the new method and that of the 2 point implicit block one-step method by Rosser (1967).

Keywords Block Method, One-Step Method, Ordinary Differential Equations

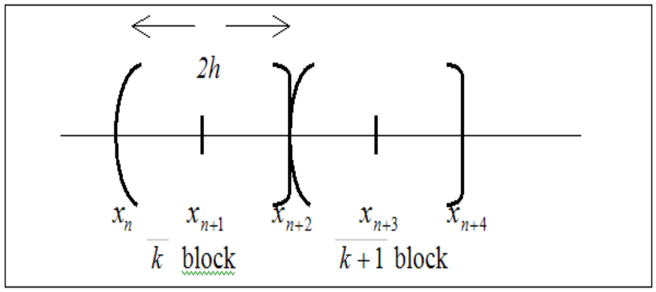
1 Introduction

Differential equations are often used to model problems in science and engineering. In most practical problems, these differential equations are highly nonlinear and cannot be solved analytically. Hence, we need an appropriate numerical integration method to solve the problems. This work is the early investigation of our research in solving ODEs using block method. Block method for numerical solutions of first order and higher order ODEs have been proposed by several researchers such as Milne (1953) who used them only as a starting value for predictor-corrector algorithm, Rosser (1967) developed Milne's proposal into a set of implicit formulas, Shampine and Watts (1969), Worland (1976), Franklin (1978), Burrage (1993), Sommeijer (1993), and Omar (1999). In this paper, the form of initial value problem (IVP) for a system of first order ODEs

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

where a and b are finite is solved using the 2 point implicit block one-step method.

Figure 1: 2 Point Method



In Figure 1, the interval $[a, b]$ is divided into a series of blocks with each block containing 2 points and 2 steps. The 2 point method will simultaneously produces two new equally spaced solution values within a block. The solution at the point x_n is used to start the k block while the solution at the point x_{n+2} is the last point in the k block will be used to start the $k+1$ block and the same process continues for the next block.

The evaluation information from the previous step in a block can be called from other steps of the same block. This method will be formulated in terms of linear multistep method but the method is equivalent to one step method.

Rosser (1967) introduced the 2 point implicit block one-step method based on the integration formula which is basically of the Newton-Cotes type. The values of y_{n+1} and y_{n+2} were approximated by integrating Equation (1) over the interval $[x_n, x_{n+1}]$ and $[x_n, x_{n+2}]$ respectively.

In this paper, the 2 point implicit block method will be formulated based on Newton backward divided difference formula. The new method used the closest point in the interval to integrate Equation (1) in order to approximate y_{n+1} and y_{n+2}. The approximation of y_{n+1} and y_{n+2} is by integrating Equation (1) over the interval $[x_n, x_{n+1}]$ and $[x_{n+1}, x_{n+2}]$ respectively.

2 2-Point Implicit Block One-Step Method

The two values, y_{n+1} and y_{n+2} are simultaneously found in a block. Let $x_{n+1} = x_n + h$, therefore,

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

or

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

We replace $f(x, y)$ in Equation (2) with polynomial interpolation as follows,

$$P_{n+2}(x) = \sum_{m=0}^{k} (-1)^m \begin{pmatrix} -s \\ m \end{pmatrix} \nabla^m f_{n+2} \quad \text{where} \quad s = \frac{x - x_{n+2}}{h}.$$

By replacing $dx = hds$ and changing the limit of integration, we obtain

$$y\left(x_{n+1}\right) = y\left(x_n\right) + h\sum_{m=0}^{k}\gamma_m\nabla^m f_{n+2} \quad \text{where} \quad \gamma_m = (-1)^m\int_{-2}^{-1}\begin{pmatrix} -s \\ m \end{pmatrix} ds \tag{3}$$

The generating function $G\left(t\right)$ for the γ_m is given by,

$$G(t) = \sum_{m=0}^{\infty}\gamma_m t^m = \sum_{m=0}^{\infty}(-t)^m\int_{-2}^{-1}\begin{pmatrix} -s \\ m \end{pmatrix} ds = \int_{-2}^{-1}\left[\sum_{m=0}^{\infty}(-t)^m\begin{pmatrix} -s \\ m \end{pmatrix}\right] ds$$

We see that

$$G\left(t\right) = \frac{-t\left(1-t\right)}{\ln\left(1-t\right)} \quad \text{or} \quad \sum_{m=0}^{\infty}\gamma_m t^m\left(\frac{\ln(1-t)}{-t}\right) = 1-t$$

and it follows that

$$(\gamma_0 t^0 + \gamma_1 t^1 + \gamma_2 t^2 + \gamma_3 t^3 + \cdots)(1 + \frac{t}{2} + \frac{t^2}{3} + \frac{t^3}{4} + \cdots) = (1-t).$$

Grouping and comparing coefficients yields

$$\gamma_0 = 1, \quad \gamma_1 = -1 - \frac{\gamma_0}{2}, \quad \gamma_m = -\sum_{r=0}^{m-1}\frac{\gamma_r}{m+1-r} \quad m = 2, 3, 4, \ldots \tag{4}$$

and the values of γ_m when $m = 0, 1$ and 2 are as follows,

$$\gamma_0 = 1, \quad \gamma_1 = -\frac{3}{2}, \quad \gamma_2 = \frac{5}{12}.$$

Formulae (3) can be written in the form

$$y_{n+1} = y_n + h\sum_{m=0}^{k}\beta_{k,m}f_{n+2-m} \tag{5}$$

where k is the number of interpolation points and

$$\beta_{k,m} = (-1)^m\sum_{r=m}^{k}\begin{pmatrix} r \\ m \end{pmatrix}\gamma_r. \tag{6}$$

Let $k = 2$ in Equation (5) and Equation(6), will give the formulae of the first point in the 2 point block as follows,

$$y_{n+1} = y_n + \frac{h}{12}(5f_n + 8f_{n+1} - f_{n+2}).$$

Now taking $x_{n+2} = x_{n+1} + h$ and integrating f once from x_{n+1} to x_{n+2} in Equation (2) gives

$$y\left(x_{n+2}\right) = y\left(x_{n+1}\right) + \int_{x_{n+1}}^{x_{n+2}}\sum_{m=0}^{k}(-1)^m\begin{pmatrix} -s \\ m \end{pmatrix}\nabla^m f_{n+2}dx \tag{7}$$

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

$$y\left(x_{n+2}\right) = y\left(x_{n+1}\right) + h\sum_{m=0}^{k}\delta_m\nabla^m f_{n+2} \quad \text{where} \quad \delta_m = (-1)^m\int_{-1}^{0}\begin{pmatrix} -s \\ m \end{pmatrix} ds. \tag{8}$$

The generating function $M(t)$ for the δ_m is given by,

$$M(t) = \sum_{m=0}^{\infty}\delta_m t^m = \sum_{m=0}^{\infty}(-t)^m\int_{-1}^{0}\begin{pmatrix} -s \\ m \end{pmatrix} ds = \int_{-1}^{0}\left[\sum_{m=0}^{\infty}(-t)^m\begin{pmatrix} -s \\ m \end{pmatrix}\right] ds$$

We see that $M(t) = \dfrac{-t}{\ln(1-t)}$ or $\displaystyle\sum_{m=0}^{\infty}\delta_m t^m\left(\frac{\ln(1-t)}{-t}\right) = 1$
which can be written as

$$(\delta_0 + \delta_1 t + \delta_2 t^2 + \delta_3 t^3 + \cdots)(1 + \frac{t}{2} + \frac{t^2}{3} + \frac{t^3}{4} + \cdots) = 1.$$

Grouping and comparing the terms yields,

$$\delta_0 = 1, \quad \delta_m = -\sum_{r=0}^{m-1}\frac{\delta_m}{m+1-r} \quad m = 1, 2, \ldots \tag{9}$$

The value of δ_m, when $m = 0, 1$ and 2 are as follows

$$\delta_0 = 1, \quad \delta_1 = -\frac{1}{2}, \quad \delta_2 = -\frac{1}{12}.$$

Formulae (8) can be written in the form

$$y_{n+2} = y_{n+1} + h\sum_{m=0}^{k}\alpha_{k,m} f_{n+2-m} \tag{10}$$

where k is the number of interpolation points and

$$\alpha_{k,m} = (-1)^m\sum_{r=m}^{k}\begin{pmatrix} r \\ m \end{pmatrix}\delta_r. \tag{11}$$

Let $k = 2$ in Equation (10) and Equation (11), will give the formulae of the second point in the 2 point block as follows,

$$y_{n+2} = y_{n+1} + \frac{h}{12}(5f_{n+2} + 8f_{n+1} - f_n).$$

Hence, the formulae of the 2 point implicit block method are

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{12}(5f_n + 8f_{n+1} - f_{n+2}) \\ \text{and} \quad y_{n+2} &= y_{n+1} + \frac{h}{12}(5f_{n+2} + 8f_{n+1} - f_n). \end{aligned} \tag{12}$$

3 Implementation of 2-Point Implicit Block One-Step Method

We will approximate y_{n+1} and y_{n+2} in Equation (12) by the iteration process as follows:

$$y_{n+m,r} = y_n + mhf_n \quad m = 1, 2. \quad r = 0. \tag{13}$$

$$y_{n+1,r+1} = y_n + \frac{h}{12}(5f_n + 8f_{n+1,r} - f_{n+2,r}) \tag{14}$$

$$y_{n+2,r+1} = y_{n+1,r} + \frac{h}{12}(-f_n + 8f_{n+1,r} + 5f_{n+2,r}) \quad r = 0, 1, 2, 3 \tag{15}$$

Define Equation (13) as the initial approximation, each $y_{n+m,r}$ is an approximation to y_{n+m} of order $r+2$. Hence, $f_{n+m,r}$ is an approximation to f_{n+m} of order $r+2$. Since $f_{n+m,r}$ are multiplied by coefficients of order h in Equation (14) and Equation (15), and it turns out that $y_{n+m,r+1}$ will be an approximation of order $r+3$. At $r = 1$ will give method of order 3 and if $r > 1$ can improves the accuracy but still at the same order. In the program, we use $r = 2$ and the convergent test will be

$$\|y_{n+2,r+1} - y_{n+2,r}\| < 0.1 \times \quad \text{TOLERANCE.}$$

4 2-Point Implicit Block One-Step Method Half Gauss Seidel

In Equation (14) and Equation(15), the approach is similar to the Jacobi iteration. At the $r+1$th iteration, the approximate value of $y_{n+1,r}$ in Equation (15) is from the previous iteration and the order is one less. Hence, we replace the algorithm by Equation (16) as follows,

$$\begin{aligned} y_{n+1,r+1} &= y_n + \frac{h}{12}(5f_n + 8f_{n+1,r} - f_{n+2,r}) \\ y_{n+2,r+1} &= y_{n+1,r+1} + \frac{h}{12}(-f_n + 8f_{n+1,r} + 5f_{n+2,r}) \quad r = 0, 1, 2 \end{aligned} \tag{16}$$

In Equation (16), the approximate value of $y_{n+1,r+1}$ is from the same iteration to replace $y_{n+1,r}$ from the previous iteration and this is the Gauss Seidel style. We observed that the numerical results are much better.

5 Stability region

The stability of the 2-point implicit block one step method derived in the previous section on a linear first order problem when the method is applied to the test equation

$$y' = f = \lambda y \tag{17}$$

The formula of the 2-point block one step method is given by Equation (13)–(15). For $r = 0$, substitute $f_{n+1,0}$ and $f_{n+2,0}$ from Equation (13) into the right hand side of Equation (14) and Equation (15). When $r = 1$, substitute $f_{n+1,1}$ and $f_{n+2,1}$ into the right hand side of Equation (14) and Equation (15) and the process continue. The characteristics polynomials of the method at $r = 0, 1, 2$ are as follows,

At $r = 0$,

$$t^2 - \left(2\overline{h}^2 + 2\overline{h} + 1\right) t = 0 \tag{18}$$

At $r = 1$,

$$t^2 - (\frac{4}{3}\overline{h}^3 + 2\overline{h}^2 + 2\overline{h} + 1)t = 0 \tag{19}$$

At $r = 2$,

$$t^2 - (\frac{2}{3}\overline{h}^4 + \frac{4}{3}\overline{h}^3 + 2\overline{h}^2 + 2\overline{h} + 1)t = 0 \tag{20}$$

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

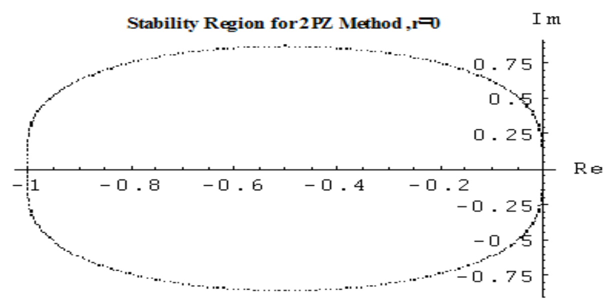


Figure 2: Stability Region for 2PZ at $r =$ 0.

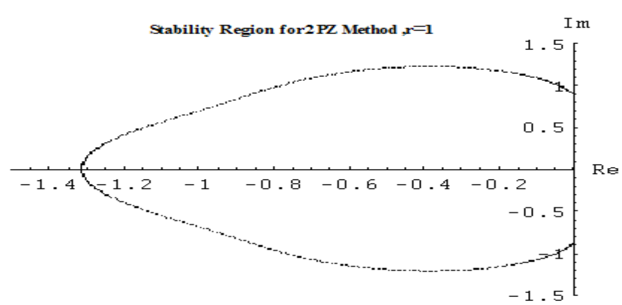


Figure 3: Stability Region for 2PZ at $r = 1$.

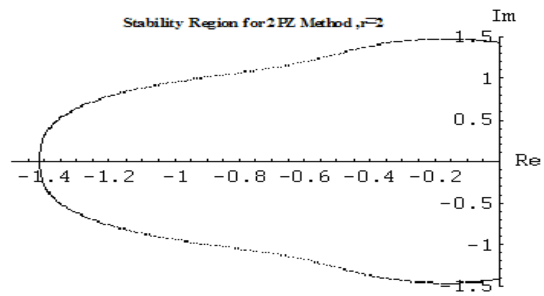


Figure 4: Stability Region for 2PZ at $r = 2$.

The stability region of method 2PZ is inside the boundary of the circle. It is observed from Figure 2-4 that the stability region is bigger as we increase the number of iteration.

6 Numerical Results

The tables below show the numerical results for the three given problems when solved using the method obtained from the previous section compare with the 2 point implicit block one-step method in Rosser (1967).

The following notations are used in the tables:

TOL	Tolerance
MTD	Method Employed
TSTEP	Total Steps Taken
FS	Total Failure Step
MAXERR	Magnitude of the maximum error of the computed solution
FCN	Total Function Calls
TIME	The execution time taken (in microseconds)
2PZ	Implementation of the 2 point implicit block one-step method by using Jacobi iteration
2PR	Implementation of the 2 point implicit block one-step method by Rosser using Jacobi iteration
2PZhG	Implementation of the 2 point implicit block one-step method by using half Gauss Seidel iteration

Tested Problems:

Problem 1: $y_1' = -Ay_1 - By_2, \quad y_2' = By_1 - Ay_2, \quad A = 1, \quad B = \sqrt{3}$

$$y_1(0) = 1, \quad y_2(0) = 0, \quad [0, 20]$$

Exact Solution:

$$y_1(x) = e^{-Ax} \cos Bx, \quad y_2(x) = e^{-Ax} \sin Bx$$

Source: Tam, H. W. (1992)

Problem 2: $y_1' = y_2, \quad y_2' = 2y_2 - y_1,$

$$y_1(0) = 0, \quad y_2(0) = 1, \quad [0, 20]$$

Exact Solution:

$$y_1(x) = xe^x, \quad y_2(x) = (1+x)e^x,$$

Source: Bronson (1973)

Problem 3:$y_1' = y_2, \quad y_2' = -y_3, \quad y_3' = y_4, \quad y_4' = y_2 + 2e^x$

$$y_1(0) = 0, \quad y_2(0) = -2, \quad y_3(0) = 0, \quad y_4(0) = 2, \quad [0, 10]$$

Exact Solution:

$$y_1(x) = -e^x + e^{-x}, \quad y_2(x) = -e^x - e^{-x}, \quad y_3(x) = e^x - e^{-x}, \quad y_4(x) = e^x + e^{-x},$$

Source: Bronson (1973)

Table 1: Comparison between the 2PZ, 2PR and 2PZhG methods for solving Problem 1

TOL	MTD	TSTEP	FS	MAXERR	FNC	TIME
10^{-2}	2PZ	48	6	4.91205(-4)	385	5818
	2PR	55	3	5.78554(-4)	441	6701
	2PZhG	51	9	4.26915(-4)	409	6007
10^{-4}	2PZ	175	3	1.09711(-5)	1401	22180
	2PR	172	2	1.04141(-6)	1377	21313
	2PZhG	143	2	5.43487(-6)	1145	17932
10^{-6}	2PZ	784	4	4.79824(-7)	6273	99539
	2PR	612	2	4.86775(-9)	4897	76125
	2PZhG	518	3	2.27365(-8)	4145	65027
10^{-8}	2PZ	3634	4	1.69271(-8)	29073	462879
	2PR	3406	3	1.43191(-11)	27249	423379
	2PZhG	1866	2	1.15143(-10)	14929	234709
10^{-10}	2PZ	17388	5	8.74602(-10)	139105	2211383
	2PR	12661	3	5.02730(-14)	101289	1576134
	2PZhG	10404	3	2.64557(-13)	83233	1310179

In all tested problems, the 2PZ is very inefficient and costly in terms of total number of steps and execution time especially when tested for finer tolerances. The maximum error of 2PZhG is comparable or one decimal places less than 2PR and still within the given

Table 2: Comparison between the 2PZ, 2PR and 2PZhG methods for solving Problem 2

TOL	MTD	TSTEP	FS	MAXERR	FNC	TIME
10^{-2}	2PZ	157	3	1.36866(-3)	1257	8572
	2PR	153	2	5.84725(-5)	1225	7881
	2PZhG	153	2	5.84725(-5)	1225	8072
10^{-4}	2PZ	535	5	1.75043(-4)	4281	29271
	2PR	955	3	1.92036(-7)	7641	49486
	2PZhG	524	3	5.80968(-7)	4193	27777
10^{-6}	2PZ	3122	7	5.09251(-6)	24977	171984
	2PR	3798	5	3.19441(-10)	30385	196135
	2PZhG	3094	4	7.64143(-10)	24753	164357
10^{-8}	2PZ	11100	8	4.700261(-7)	88801	610284
	2PR	20123	6	6.12502(-12)	160985	1040890
	2PZhG	11040	5	3.02368(-12)	88321	587304
10^{-10}	2PZ	62902	10	1.286325(-8)	503217	3461814
	2PR	92019	8	1.37597(-11)	736153	4734659
	2PZhG	62863	7	1.31379(-11)	502905	3345296

Table 3: Comparison between the 2PZ, 2PR and 2PZhG methods for solving Problem 3

TOL	MTD	TSTEP	FS	MAXERR	FNC	TIME
10^{-2}	2PZ	43	2	1.32609(-4)	345	5703
	2PR	39	1	4.59025(-5)	313	4965
	2PZhG	39	2	4.59025(-5)	313	5165
10^{-4}	2PZ	240	2	1.47953(-5)	1921	32548
	2PR	120	1	5.26513(-7)	961	15349
	2PZhG	120	1	5.26513(-7)	961	15967
10^{-6}	2PZ	1508	3	3.85967(-7)	12065	205049
	2PR	755	2	3.46349(-10)	6041	96778
	2PZhG	378	1	5.48902(-9)	3025	50426
10^{-8}	2PZ	4760	3	3.88082(-8)	38081	648384
	2PR	2381	2	3.42486(-12)	19049	305250
	2PZhG	2381	2	3.24632(-12)	19049	317678
10^{-10}	2PZ	30089	4	9.73023(-10)	240713	4093234
	2PR	15046	3	1.69442(-12)	120369	1928971
	2PZhG	7524	2	1.18945(-12)	60193	1003358

tolerances. At the same total number of steps, the execution times taken by the 2PR is slightly better than 2PZhG. This could be justified by the fact that the time spent on performing extra computations required in 2PZhG has affected the execution times. It could be observed that the reduction in the number of steps in the 2PZhG gives better execution time than the 2PR.

7 Conclusion

Method 2PZhG is more efficient than method 2PR and 2PZ as the tolerance getting smaller.

References

[1] R. Bronson, *Schaums Outline of Modern Introductory Differential Equation*, USA: Mc Graw-Hill (1993).

[2] K. Burrage, *Parallel Methods for Initial Value Problems*, Applied Numerical Mathematics, 11(1993), 5-25.

[3] M.A. Franklin, *Parallel Solution of Ordinary Differential Equations*, IEEE Transaction on Computer, c-27(5) (1978): 413-420.

[4] P. Henrici *Discrete Variable Methods in Ordinary Differential Equations,* New York: John Wiley & Sons, Inc.(1962)

[5] J.D. Lambert, *Computational Methods in Ordinary Differential Equations* , New York: John Wiley & Sons, Inc.(1973)

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

[7] W.E. Milne, *Numerical Solution of Differential Equations*, John Wiley, New York (1953)

[8] Zurni. Omar, *Developing Parallel Block Methods For Solving Higher Order ODEs Directly*, Thesis, University Putra Malaysia (1999), Malaysia.

[9] J.B. Rosser, *Runge Kutta For All Seasons*, SIAM Rev., Vol. 9(1967): 417-452.

[10] B.P. Sommeijer, *Parallelism in the Numerical Integration of Initial Value Problems*, CWI Tract 99, CWI, Amsterdam (1993).

[11] H.W. Tam, *Two Stage Parallel Methods for the Numerical Solution of Ordinary Differential Equations*, Siam J. Sci. Stat Comput.13 (5)(1992):1062-1084.

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

[13] Zanariah Abdul Majid et. al, *Kaedah Blok Tersirat Dua Titik Menggunakan Separa Gauss Seidel Bagi Menyelesaikan Persamaan Pembezaan Biasa Peringkat Pertama*, Prosiding Simposium Kebangsaan Sains Matematik Ke 9(2001). UKM: 371-376.