BULLETIN *of the*
MALAYSIAN MATHEMATICAL
SCIENCES SOCIETY
http://math.usm.my/bulletin

Bull. Malays. Math. Sci. Soc. (2) **29**(1) (2006), 23–31

3-Point Implicit Block Method for Solving Ordinary Differential Equations

[1]Zanariah Abdul Majid, [2]Mohamed Bin Suleiman and [3]Zurni Omar
[1,2]Mathematics Department, Faculty of Science,
Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[3]Faculty of Quantitative Science, Universiti Utara Malaysia,
06010 UUM, Sintok, Kedah, Malaysia
[1]zanamajid@hotmail.com, [2]m1suleiman@yahoo.com.my

Abstract. A 3-point implicit block method for solving system of first order ordinary differential equations (ODEs) is proposed. This method approximates the solutions of initial value problems at 3 points simultaneously using variable step size. The stability of the method is also studied. The numerical results show that the method is more efficient than the 3-point implicit block method developed by Rosser [4] in terms of the total number of steps and execution times.

2000 Mathematics Subject Classification: 65L06, 65L05

Key words and phrases: 3-point, implicit block method, ordinary differential equations.

1. Introduction

This paper considers initial value problems (IVPs) for a system of first order ODEs in the following form,

$$y' = f(x, y), \quad y(a) = y_0 \quad a \le x \le b \tag{1.1}$$

where a and b are finite. Scientific and technological problems often lead to mathematical modeling of real life applications such as the motion of projectiles or orbiting bodies, population growth, chemical kinetics and economic growth. Differential equations are often used to model the problems and most of time these equations do not have analytic solutions. Hence, an appropriate numerical method is required to solve the problems. Block methods for numerical solutions of first order ODEs have been proposed by several researchers such as Milne [2], Rosser [4], Shampine and Watts [5], Worland [7] and Omar [3]. Rosser [4] introduced the 3-point implicit block method based on the integration formulae which is basically of the Newton-Cotes type. The values of y_{n+1}, y_{n+2} and y_{n+3} were approximated by integrating (1.1) over the interval $[x_n, x_{n+1}]$, $[x_n, x_{n+2}]$ and $[x_n, x_{n+3}]$ respectively.

Received: July 17, 2004; *Revised:* September 19, 2005.

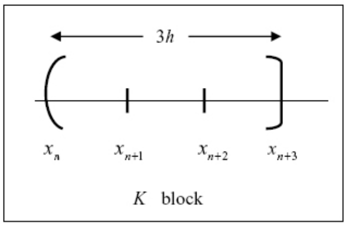

Figure 1. 3-point implicit block method

In this paper, we attempt to derive 3-point implicit block method based on Newton backward divided difference formulae. Unlike Rosser, we approximate y_{n+1}, y_{n+2} and y_{n+3} by integrating (1.1) over the interval $[x_n, x_{n+1}]$, $[x_{n+1}, x_{n+2}]$ and $[x_{n+2}, x_{n+3}]$ respectively.

2. Derivation of 3-point implicit block method

In 3-point implicit block method, the interval $[a, b]$ is divided into a series of blocks with each block containing 3 points (refer to Figure 1). The following strategy is employed to calculate the solutions at each block. The solution at the point x_n, which is the end point of $K-1$ block, is used to calculate the solutions of K block. Similarly, the solution at the end point of K block, which is at x_{n+3}, is used to calculate the solutions of $K+1$ block. The same process applied for calculating the next blocks until the end point $x = b$ is reached.

To approximate y_{n+1}, takes $x_{n+1} = x_n + h$ and integrate (1.1) gives

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

which is equivalent to

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y) \; dx. \tag{2.1}$$

Define $P_{n+3}(x)$ as the interpolation polynomial which interpolates $f(x, y)$ in (2.1) at the set of points $\{x_n, x_{n+1}, x_{n+2}, x_{n+3}\}$ as follows

$$P_{n+3}(x) = \sum_{m=0}^{k} (-1)^m \binom{-s}{m} \nabla^m f_{n+3}$$

where

$$s = \frac{x - x_{n+3}}{h} \text{ and } k = 3.$$

By replacing $dx = h\, ds$ and changing the limit of integration gives

$$y(x_{n+1}) = y(x_n) + h \sum_{m=0}^{k} \sigma_m \nabla^m f_{n+3} \tag{2.2}$$

where

$$\sigma_m = (-1)^m \int_{-3}^{-2} \begin{pmatrix} -s \\ m \end{pmatrix} ds.$$

By solving (2.2) will produce the formulae of the first point as follows

$$y_{n+1} = y_n + \frac{h}{24}(f_{n+3} - 5f_{n+2} + 19f_{n+1} + 9f_n)). \tag{2.3}$$

Now taking $x_{n+2} = x_{n+1} + h$ in equation (1.1), replacing $dx = h\ ds$ and changing the limit of integration from -2 to -1 gives,

$$y(x_{n+2}) = y(x_{n+1}) + h \sum_{m=0}^{k} \gamma_m \nabla^m f_{n+3} \tag{2.4}$$

where

$$\gamma_m = (-1)^m \int_{-2}^{-1} \begin{pmatrix} -s \\ m \end{pmatrix} ds.$$

Again by solving (2.4) will give the formulae for the second point as follows

$$y_{n+2} = y_{n+1} + \frac{h}{24}(-f_{n+3} + 13f_{n+2} + 13f_{n+1} - f_n). \tag{2.5}$$

Now, taking $x_{n+3} = x_{n+2} + h$ in (1.1), replacing $dx = h\ ds$ and changing the limit of integration from -1 to 0 gives

$$y(x_{n+3}) = y(x_{n+2}) + h \sum_{m=0}^{k} \delta_m \nabla^m f_{n+3}$$

where

$$\delta_m = (-1)^m \int_{-1}^{0} \begin{pmatrix} -s \\ m \end{pmatrix} ds. \tag{2.6}$$

Solving (2.6) will give the formulae for the third point as follows

$$y_{n+3} = y_{n+2} + \frac{h}{24}(9f_{n+3} + 19f_{n+2} - 5f_{n+1} + f_n) \tag{2.7}$$

3. Implementation of 3-point implicit block method

The values of y_{n+1}, y_{n+2} and y_{n+3} in (2.3), (2.5) and (2.7) will be approximated using the predictor-corrector schemes. If r corrections are needed, then the sequence of computations at any mesh point is $(PE)(C_0E)\dots(C_rE)$ where P and C indicate the application of the predictor and corrector formulae respectively and E indicate the evaluation of the function f.

The predictor equations:

$$\begin{aligned} P:&\quad y^p_{n+m,0} = y^c_n + mhf^c_n, \quad m = 1, 2, 3, \\ E:&\quad f^p_{n+m,0} = f(x_{n+m}, y^p_{n+m,0}). \end{aligned} \tag{3.1}$$

The corrector equations: For $r = 0$,

$$y^c_{n+1,1} = y^c_n + \frac{h}{24}(f^p_{n+3,0} - 5f^p_{n+2,0} + 19f^p_{n+1,0} + 9f^c_n) \tag{3.2}$$

$$C_0: \quad y^c_{n+2,1} = y^p_{n+1,0} + \frac{h}{24}(-f^p_{n+3,0} + 13f^p_{n+2,0} + 13f^p_{n+1,0} - f^c_n) \tag{3.3}$$

$$y^c_{n+3,1} = y^p_{n+2,0} + \frac{h}{24}(9f^p_{n+3,0} + 19f^p_{n+2,0} - 5f^p_{n+1,0} + f^c_n) \tag{3.4}$$

$$E: \quad f^c_{n+m,1} = f(x_{n+m}, y^c_{n+m,1}).$$

For $r = 1, 2, 3$,

$$y^c_{n+1,r+1} = y^c_n + \frac{h}{24}(f^c_{n+3,r} - 5f^c_{n+2,r} + 19f^c_{n+1,r} + 9f^c_n) \tag{3.5}$$

$$C_r: \quad y^c_{n+2,r+1} = y^c_{n+1,r} + \frac{h}{24}(-f^c_{n+3,r} + 13f^c_{n+2,r} + 13f^c_{n+1,r} - f^c_n) \tag{3.6}$$

$$y^c_{n+3,r+1} = y^c_{n+2,r} + \frac{h}{24}(9f^c_{n+3,r} + 19f^c_{n+2,r} - 5f^c_{n+1,r} + f^c_n) \tag{3.7}$$

$$E: \quad f^c_{n+m,r+1} = f(x_{n+m}, y^c_{n+m,r+1}) \tag{3.8}$$

Define (3.1) as the initial approximation and each $f^p_{n+m,0}$ is an approximation of order $O(h)$. Since $f^p_{n+m,0}$ are multiplied by the coefficients of order h in (3.2), (3.3) and (3.4), it turns out that $y^c_{n+m,r+1}$ will be an approximation of order $O(h^2)$. At $r = 2$, it would simulate method of order $O(h^4)$ and continue iterating until $r = 3$ can improves the accuracy but still within the same order. In the program, we only allowed the iteration up to $r = 3$ and the iteration can be terminate before the maximum r if the convergence test has satisfied,

$$\left\| y^c_{n+3,r+1} - y^c_{n+3,r} \right\| < 0.1 \times \text{ TOLERANCE.} \tag{3.9}$$

4. 3-point implicit block method in half Gauss Seidel

In (3.2)–(3.7), the approach is similar to the Jacobi iteration. At $r = 0$, the approximate value of $y^p_{n+1,0}$ in (3.3) and $y^p_{n+2,0}$ in (3.4) are from the predictor values. When $r = 1, 2, 3$, the approximate value of $y^c_{n+1,r}$ in (3.6) and $y^c_{n+2,r}$ in (3.7) is from the previous iteration and the order is one less. Hence, we replace the algorithm as follows, For $r = 0$,

$$y^c_{n+2,r+1} = y^c_{n+1,r+1} + \frac{h}{24}(-f^p_{n+3,r} + 13f^p_{n+2,r} + 13f^p_{n+1,r} - f^c_n) \tag{4.1}$$

$$y^c_{n+3,r+1} = y^c_{n+2,r+1} + \frac{h}{24}(9f^p_{n+3,r} + 19f^p_{n+2,r} - 5f^p_{n+1,r} + f^c_n) \tag{4.2}$$

For $r = 1, 2, 3$,

$$y^c_{n+2,r+1} = y^c_{n+1,r+1} + \frac{h}{24}(-f^c_{n+3,r} + 13f^c_{n+2,r} + 13f^c_{n+1,r} - f^c_n) \tag{4.3}$$

$$y^c_{n+3,r+1} = y^c_{n+2,r+1} + \frac{h}{24}(9f^c_{n+3,r} + 19f^c_{n+2,r} - 5f^c_{n+1,r} + f^c_n) \tag{4.4}$$

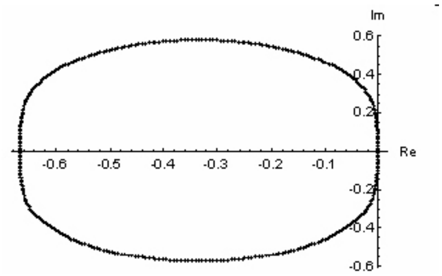

Figure 2. Stability Region for 3PZ at $r = 0$

In (4.1) and (4.3), the estimated value of $y^c_{n+1,r+1}$ is from the same iteration to replace $y^p_{n+1,r}$ in (3.3) and $y^c_{n+1,r}$ in (3.6). The same strategy follows in (4.2) and (4.4), taking the value of $y^c_{n+2,r+1}$ to replace $y^p_{n+2,r}$ in (3.4) and $y^c_{n+2,r}$ in (3.7). This strategy is again known as the Gauss Seidel style. We observed that the numerical results are much better.

5. Stability region

The stability of the 3-point implicit block method derived in the previous section on a linear first order problem when the method is applied to the test equation

(5.1) $$y' = f = \lambda y.$$

The formulae of the 3-point implicit block method are given by (3.1)–(3.7). For $r = 0$, substitute $f^p_{n+1,0}$, $f^p_{n+2,0}$ and $f^p_{n+3,0}$ from (3.1) into the right hand side of (3.2), (3.3) and (3.4). When $r = 1$, substitute $f^c_{n+1,1}$, $f^c_{n+2,1}$ and $f^c_{n+3,1}$ into the right hand side of (3.5)–(3.7) and the process continues. The characteristics polynomial of the 3-point implicit block method at $r = 0, 1, 2$ is as follows:
At $r = 0$,

(5.2) $$t^3 - (1 + 3\overline{h} + \frac{9}{2}\overline{h}^2)t^2 = 0$$

At $r = 1$,

(5.3) $$t^3 - (1 + 3\overline{h} + \frac{9}{2}\overline{h}^2 + \frac{9}{2}\overline{h}^3)t^2 = 0$$

At $r = 2$,

(5.4) $$t^3 - (1 + 3\overline{h} + \frac{9}{2}\overline{h}^2 + \frac{9}{2}\overline{h}^3 + \frac{51}{16}\overline{h}^4)t^2 = 0$$

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

The stability region of 3PZ is inside the boundary of the circle. At $r = 0$, the corrector equations are dependent on values taken from the predictor equations which are explicit formulae. But as the r increases, the corrector values have dominated the corrector equations which are implicit formulae. Therefore, it is observed from Figure 2 to 4 that the stability region is larger as the number of iteration increased.

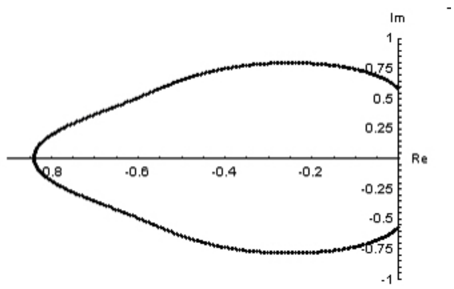

Figure 3. Stability Region for 3PZ at $r = 1$

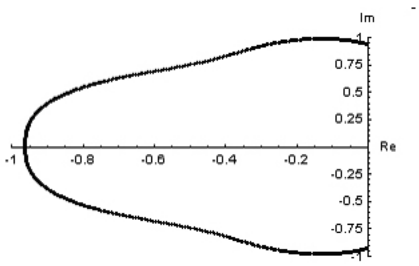

Figure 4. Stability Region for 3PZ at $r = 2$

6. Tested problems

The following problems were tested on the DYNIX/ptx operating system:

Problem 6.1. $y' = -y$ $y(0) = 1$, $[0, 20]$ Exact Solution: $y(x) = e^{-x}$.
Source: Artificial Problem

Problem 6.2. $y' = y$ $y(0) = 1$, $[0, 20]$ Exact Solution: $y(x) = e^{x}$.
Source: Artificial Problem

Problem 6.3. $y_1' = -Ay_1 - By_2$ $y_2' = By_1 - Ay_2$, $A = 1$, $B = \sqrt{3}$, $y_1(0) = 1$, $y_2(0) = 0$, $[0, 20]$ Exact Solution: $y_1(x) = e^{-Ax}\cos Bx$. $y_2(x) = e^{-Ax}\sin Bx$.
Source: Tam [6].

Problem 6.4. $y_1' = y_2$, $y_2' = 2y_2 - y_1$, $y_1(0) = 0$, $y_2(0) = 1$, $[0, 20]$ Exact Solution: $y_1(x) = xe^{x}$, $y_2(x) = (1 + x)e^{x}$,
Source: Bronson [1].

7. Numerical results

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
FCN	Total function calls
TIME	The execution time taken in microseconds
3PZ	Implementation of the 3 point implicit block method using Jacobi iteration
3PR	Implementation of the 3 point implicit block method by Rosser [4] using Jacobi iteration
3PZhG	Implementation of the 3 point implicit block method in half Gauss Seidel iteration
R_{TIME}	The ratio execution times of 3PZhG to the 3PR and 3PZ

The true solution is required in order to calculate the maximum error. It is defined as follows: Let

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

where the notation $(y)_t$ is the t th component of y. A=1, B=0 corresponds to the absolute error test, A=1, B=1 corresponds to mixed error test and finally A=0, B=1 corresponds to relative error test. The maximum error and average error are defined as follows:

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

and

$$AVERR = \frac{\sum_{i=1}^{SSTEP} \sum_{t=1}^{N} (e_i)_t}{(P)(N)(SSTEP)}$$

where N is the number of equations in the system, $SSTEP$ is the number of successful steps and P is the number of point. For Problems 2 and 4, the relative error test is used. The absolute and mixed error test is for Problem 1 and 3 respectively. At each step of integration, a test for checking the end of the interval is made. If b denotes the end of the interval then if $x + 3h \ge b$ then $h = \frac{(b-x)}{3}$, otherwise h remain as calculated. The technique above helped to reach the end point of the interval. After the successful convergence test of [3.5], local errors estimate Est at x_{n+3} will be performed to control the error for the block.

We compare the absolute difference of the corrector formula derived of order k and a similar corrector formula of order $k-1$. Therefore, we obtain

$$Est = \left| \frac{h}{24} (f_{n+3} - 3f_{n+2} + 3f_{n+1} - f_n) \right| \tag{7.1}$$

The value Est is compared with the prescribed tolerance TOL and the step is accepted if

$$Est < TOL$$

and rejected otherwise. The step size prediction in the next step is given by

$$h = fac \times h_{old} \times \left(\frac{TOL}{Est_k} \right)^{\frac{1}{p}},$$

and if $(h > 2 \times h_{old})$ then

$$h = 2 \times h_{old},$$

where *fac* is a safety factor and h_{old} is the step size from previous block. The safety factor used in the program was 0.5 because from computational tested it had reduced the number of steps rejected. The error controls for all the methods are at the third point in the block because in general it had given us better results.

The tables below show the numerical results for the four given problems when solved using the method obtained from the previous section compare with the 3-point implicit block method in Rosser [4].

Table 1. Comparison between the 3PZ, 3PR and 3PZhG methods for solving Problem 1.

TOL	MTD	TS	FS	MAXE	AVERR	FNC	TIME	R_{TIME}
10^{-2}	3PZ	45	2	7.05966(-5)	1.55040(-5)	676	2381	2.52
	3PR	20	1	3.72960(-5)	9.38930(-6)	301	944	1.27
	3PZhG	16	1	6.38350(-5)	1.70620(-5)	241	741	1.00
10^{-4}	3PZ	170	3	1.77719(-6)	5.42886(-7)	2551	8739	4.95
	3PR	49	1	3.59617(-7)	9.92531(-8)	736	2321	1.31
	3PZhG	38	1	7.55105(-7)	1.76192(-7)	571	1766	1.00
10^{-6}	3PZ	654	4	6.22500(-8)	2.46430(-8)	9811	33726	7.15
	3PR	126	1	4.94290(-9)	1.30479(-9)	1891	6010	1.27
	3PZhG	101	1	4.94290(-9)	1.82891(-9)	1516	4715	1.00
10^{-8}	3PZ	2362	5	2.40197(-9)	1.32897(-9)	35431	121909	9.48
	3PR	339	1	8.84362(-11)	2.15738(-11)	5086	16170	1.26
	3PZhG	274	1	8.84362(-11)	2.77222(-11)	4111	12857	1.00
10^{-10}	3PZ	7323	6	3.18638(-10)	1.42314(-10)	109846	377951	10.75
	3PR	1412	2	1.09967(-13)	3.33205(-14)	21181	67312	1.91
	3PZhG	750	1	1.87599(-12)	5.25147(-13)	11251	35162	1.00

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

TOL	MTD	TS	FS	MAXE	AVERR	FNC	TIME	R_{TIME}
10^{-2}	3PZ	79	2	3.29774(-4)	1.59790(-4)	1186	3870	2.13
	3PR	40	1	5.36207(-4)	2.71766(-4)	601	1892	1.04
	3PZhG	40	1	5.36207(-4)	2.71766(-4)	601	1816	1.00
10^{-4}	3PZ	390	3	6.85574(-5)	3.42863(-5)	5851	19201	4.33
	3PR	98	1	4.20484(-6)	2.10176(-6)	1471	4542	1.02
	3PZhG	98	1	4.20484(-6)	2.10176(-6)	1471	4437	1.00
10^{-6}	3PZ	1946	4	3.16305(-6)	1.58112(-6)	29191	95991	8.68
	3PR	244	1	3.28661(-8)	1.65205(-8)	3661	11246	1.02
	3PZhG	244	1	3.28661(-8)	1.65205(-8)	3661	11057	1.00
10^{-8}	3PZ	9761	5	1.28837(-7)	6.44181(-8)	146416	481690	17.44
	3PR	611	1	2.42991(-9)	1.21766(-9)	9166	28225	1.02
	3PZhG	611	1	2.42991(-9)	1.21766(-9)	9166	27612	1.00
10^{-10}	3PZ	49036	6	5.13356(-9)	2.56472(-9)	735541	2423761	34.91
	3PR	3066	2	6.39593(-12)	2.95087(-12)	45991	142476	2.05
	3PZhG	1533	1	7.83933(-11)	3.91244(-11)	22996	69420	1.00

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

TOL	MTD	TS	FS	MAXE	AVERR	FNC	TIME	R_{TIME}
10^{-2}	3PZ	89	3	1.22292(-4)	1.81607(-5)	1336	19162	2.73
	3PR	40	3	1.83581(-4)	1.42535(-5)	601	8215	1.17
	3PZhG	35	4	2.39153(-4)	3.15568(-5)	526	7016	1.00
10^{-4}	3PZ	355	4	3.32710(-6)	6.25342(-7)	5326	76727	4.68
	3PR	92	2	5.68483(-7)	1.14458(-7)	1381	19170	1.17
	3PZhG	80	3	4.21205(-6)	2.19494(-7)	1201	16400	1.00

10^{-6}	3PZ	1404	5	1.22241(-7)	2.90040(-8)	21061	304080	7.00
	3PR	246	2	7.68707(-9)	1.45027(-9)	3691	51440	1.18
	3PZhG	210	2	7.68707(-9)	1.81533(-9)	3151	43430	1.00
10^{-8}	3PZ	5252	6	4.77307(-9)	1.50056(-9)	78781	1137380	9.57
	3PR	663	2	1.51342(-10)	2.51263(-11)	9946	138867	1.17
	3PZhG	574	2	1.51341(-10)	2.95748(-11)	8611	118872	1.00
10^{-10}	3PZ	17424	7	4.02189(-10)	1.13529(-10)	261361	3772739	11.42
	3PR	2768	3	2.07639(-13)	4.13279(-14)	41521	580038	1.76
	3PZhG	1594	2	3.45373(-12)	5.89717(-13)	23911	330439	1.00

Table 4. Comparison between the 3PZ, 3PR and 3PZhG methods for solving Problem 4.

TOL	MTD	TS	FS	MAXE	AVERR	FNC	TIME	R_{TIME}
10^{-2}	3PZ	312	4	1.16502(-4)	6.40396(-5)	4681	27523	4.45
	3PR	79	2	2.05071(-5)	1.22404(-5)	1186	6365	1.03
	3PZhG	79	2	2.05071(-5)	1.22404(-5)	1186	6187	1.00
10^{-4}	3PZ	3099	6	1.45165(-6)	8.05481(-7)	46486	274484	17.87
	3PR	196	2	1.80050(-7)	3.92279(-9)	2941	15789	1.03
	3PZhG	196	2	1.80050(-7)	3.92279(-9)	2941	15360	1.00
10^{-6}	3PZ	26539	8	2.75066(-8)	8.46964(-9)	398086	2352855	39.69
	3PR	974	3	5.47812(-10)	3.18843(-10)	14611	78579	1.33
	3PZhG	755	3	3.24182(-9)	6.86935(-10)	11326	59279	1.00
10^{-8}	3PZ	80471	10	2.30879(-9)	1.19783(-9)	1207066	7141630	37.16
	3PR	4882	4	2.75961(-12)	1.30553(-12)	73231	394674	2.05
	3PZhG	2442	3	1.72488(-11)	9.68842(-12)	36631	192203	1.00
10^{-10}	3PZ	213245	13	3.53662(-10)	1.54355(-10)	3198676	18907864	39.16
	3PR	12257	4	3.11456(-12)	9.92319(-13)	183856	990858	2.05
	3PZhG	6130	3	1.62828(-12)	6.94792(-13)	91951	482847	1.00

8. Comment on the results and conclusion

In all tested problems, method 3PZ turns out very inefficient and costly in terms of total number of steps and execution time especially when tested for finer tolerances. The ratio execution times show that the 3PZhG is more efficient than 3PR in terms of total steps and execution time. The maximum error of 3PZhG is comparable or one decimal places less than 3PR and is still within the given tolerances. The ratios of the execution time for 3PZhG compared to 3PZ and 3PR are greater than 1 in all tested problems. Therefore, method 3PZhG is more efficient than method 3PR and 3PZ in terms of the total number of steps and execution times as the tolerance getting smaller.

References

[1] R. Bronson, *Modern Introductory Differential Equation*, Schaum's Outline Series, McGraw-Hill, USA, 1973.

[2] W. E. Milne, *Numerical Solution of Differential Equations*, Wiley, New York, 1953.

[3] Z. Omar, *Developing parallel block methods for solving higher order ODEs directly*, Thesis, University Putra Malaysia, Malaysia, 1999.

[4] J. B. Rosser, A Runge-Kutta for all seasons, *SIAM Rev.* **9** (1967), 417–452.

[5] L. F. Shampine and H. A. Watts, Block implicit one-step methods, *Math. Comp.* **23** (1969), 731–740.

[6] H.-W. Tam, Two-stage parallel methods for the numerical solution of ordinary differential equations, *SIAM J. Sci. Statist. Comput.* **13**(5) (1992), 1062–1084.

[7] P. B. Worland, Parallel methods for the numerical solution of ordinary differential equations, *IEEE Trans. Computers* **C-25**(10) (1976), 1045–1048.