*Journal of Quality Measurement and Analysis* *JQMA* 8(1) 2012, 29-41
*Jurnal Pengukuran Kualiti dan Analisis*

# TWO AND THREE POINT ONE-STEP BLOCK METHODS FOR SOLVING DELAY DIFFERENTIAL EQUATIONS

(Kaedah Blok Satu Langkah Dua dan Tiga Titik bagi Penyelesaian Persamaan Pembezaan Lengah)

HAZWANI MOHD RADZI[1], ZANARIAH ABDUL MAJID[2], FUDZIAH ISMAIL[3] & MOHAMED SULEIMAN[4]

*ABSTRACT*

In this paper, initial value problems of first order delay differential equations (DDEs) are solved using two and three point one step block method. Neville's interpolation will be implemented for the solutions of the delay argument. The general formulation of one-step block method for solving ordinary differential equations is adapted to solve DDEs. The P- and Q-stability are also discussed. Numerical results are given to illustrate the performance of those block methods for solving delay differential equations.

*Keywords:* Delay differential equations; variable step size; block method

*ABSTRAK*

Dalam makalah ini, masalah nilai awal persamaan pembezaan lengah peringkat pertama diselesaikan menggunakan kaedah blok satu langkah dua dan tiga titik. Interpolasi Neville digunakan bagi menyelesaikan sebutan lengah. Formulasi umum kaedah blok satu langkah bagi penyelesaian persamaan pembezaan biasa diadaptasi untuk menyelesaikan persamaan pembezaan lengah. Rantau kestabilan-P dan -Q juga dibincangkan. Hasil berangka disediakan untuk menggambarkan prestasi kedua-dua kaedah blok bagi penyelesaian persamaan pembezaan lengah.

*Kata kunci:* Persamaan pembezaan lengah; saiz langkah boleh berubah; kaedah blok

## 1. Introduction

Many real life phenomena in physics, biology, medicine, engineering and economics can be modeled by initial value problem (IVP) for ordinary differential equations (ODEs) of the type

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

where the function $y(x)$ is the state variable that represents some physical quantity that evolves over time. More realistic models should include some of the past states of the systems and the use of delay terms due to the recent rapid progress in the understanding of the important of differential equations with time delays.

There are so many processes in various area such as biology, medicine, chemistry, engineering, and economics that involve time delays. The equation that involve past state of the system is known as delay differential equation (DDE). In this paper, we are going to consider DDE of the form

$$\begin{aligned} y'(x) &= f(x, y, y(x-\tau)) \quad &for\ x > x_0 \\ y(x) &= \varphi(x) \quad &for\ x \le x_0 \end{aligned} \tag{2}$$

which has one delay term only. $\varphi(x)$ is the initial function, $\tau(x, y(x))$ is called the delay, $x - \tau(x, y(x))$ is called the delay argument and the value of is the solution of the delay term.

Generally, a DDE refers to both a retarded type of DDE (RDE) and a neutral type of DDE (NDE). In this paper, we only concerned DDE of the retarded type (RDE). One step block method is used to solve delay differential equations (DDEs) of the form (1).

The studies for delay differential equations have attracted some of the best mathematical minds and have led to important developments in the theory of special analytic functions. Several researchers such as Al-Mutib (1977), Oberle and Pesh (1981), Thompson (1990), Suleiman and Ismail (2001), and Ishak *et al.* (2008) have proposed the numerical solutions of DDEs. Most of the numerical methods for solving DDEs are adapted from the numerical methods for solving ODEs. In 1995, Ibrahim *et al.* has presented spline function approximation for solutions of initial-value problems in delay differential equations while Oberle and Pesh (1981) and Ismail *et al.* (2002) used Hermite interpolation to approximate the delay arguments. Later on, Mohamed *et al.* (2006) employed the polynomial spline functions to approximate the solution of a system of first-order DDEs.

In 2008, Ishak *et al.* implemented predictor-corrector scheme based on the generalised multistep methods using variable order variable step size techniques. The formulae are represented in divided difference form. The two point predictor-corrector block method used for solving delay differential equations produces the solutions at two points simultaneously. It is proved that the block method is more efficient when compared to the non-block method. Zhang and Chen (2010) suggested a block boundary value method (BBVMs) for the initial value problems of DDEs. The results showed that the derived BBVMs are effective and reliable to treat the DDEs.

## 2. Numerical Treatment of DDEs

Most numerical method for solving ordinary differential equations (ODEs) as in (1) can be adapted to solve DDE. In this paper, we implement two and three point one-step method proposed by Majid (2003; 2006) to solve DDEs. Those formulae in Majid *et al.* (2003; 2006) are as follows

Two Point One-Step Block Method:

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{12}(5f_n + 8f_{n+1} - f_{n+2}) \\ y_{n+2} &= y_{n+1} + \frac{h}{12}(-f_n + 8f_{n+1} + 5f_{n+2}) \end{aligned} \tag{3}$$

Three Point One-Step Block Method:

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{24}(9f_n + 19f_{n+1} - 5f_{n+2} + f_{n+3}) \\ y_{n+2} &= y_{n+1} + \frac{h}{24}(-f_n + 13f_{n+1} + 13f_{n+2} - f_{n+3}) \\ y_{n+3} &= y_{n+2} + \frac{h}{24}(f_n - 5f_{n+1} + 19f_{n+2} + 9f_{n+3}) \end{aligned} \tag{4}$$

Generally, the method (3) and (4) can be written as

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{j=0}^{2}\beta_j f(x_{n+j}, y_{n+j}) \\ y_{n+2} &= y_{n+1} + h\sum_{j=0}^{2}\alpha_j f(x_{n+j}, y_{n+j}) \end{aligned} \tag{5}$$

and

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{j=0}^{3}\partial_j f(x_{n+j}, y_{n+j}) \\ y_{n+2} &= y_{n+1} + h\sum_{j=0}^{3}\eta_j f(x_{n+j}, y_{n+j}) \\ y_{n+3} &= y_{n+2} + h\sum_{j=0}^{3}\omega_j f(x_{n+j}, y_{n+j}) \end{aligned} \tag{6}$$

where $x_0 < x_1 < x_2 < \ldots$ are the given grid points and $h_n = x_{n+1} - x_n, (n = 0,1,2,\ldots)$ denotes the corresponding step sizes. Delay differential equations of the form

$$\begin{aligned} y'(x) &= \lambda y(x) + \mu y(x-\tau), && x > x_0 \\ y(x) &= \phi(x), && x \le x_0 \end{aligned} \tag{7}$$

are solved at the point $x_{n+1}$ and $x_{n+2}$ for two point one-step block method and $x_{n+1}$, $x_{n+2}$ and $x_{n+3}$ for three point one-step block method. The following equations are obtained,

Two point one-step block method:

$$\begin{aligned} y_{n+1} &= y_n + h_n\sum_{j=0}^{2}\beta_j f(x_{n+j}, y_{n+j}, z_{n+j}) \\ y_{n+2} &= y_{n+1} + h_n\sum_{j=0}^{2}\alpha_j f(x_{n+j}, y_{n+j}, z_{n+j}) \end{aligned} \tag{8}$$

Three point one-step block method:

$$
\begin{aligned}
y_{n+1} &= y_n + h_n \sum_{j=0}^{3} \partial_j f(x_{n+j}, y_{n+j}, z_{n+j}) \\
y_{n+2} &= y_{n+1} + h_n \sum_{j=0}^{3} \eta_j f(x_{n+j}, y_{n+j}, z_{n+j}) \\
y_{n+3} &= y_{n+2} + h_n \sum_{j=0}^{3} \omega_j f(x_{n+j}, y_{n+j}, z_{n+j})
\end{aligned}
\tag{9}
$$

where $y_{n+1}$, $y_{n+2}$ and $y_{n+3}$ is the approximation to $y(x_{n+1})$, $y(x_{n+2})$ and $y(x_{n+3})$ respectively and $z_{n+j}$ is the approximation to $y(x_n - \tau)$, the delay term which is obtained by Neville's interpolation at the point $x = x_n - \tau$ using previous values of $y$ and $z_{n+j} = \varphi(x)$ whenever $x_n - \tau \le 0$.

The interpolation will be done by using the values of $y_n$, $x_n$ and $\tau$ and we will get $z_{n+j} = y(x_n - \tau)$. The order of interpolation used for the method in (7) and (8) should be at least at the same order with those methods in order to preserve the desired accuracy. The large errors may occur if a higher order method is used to integrate the problem while the linear interpolation is used to approximate the delayed variables. In our case, the order of Neville's interpolation used to approximate the delay terms is higher than the method used.

## 3. Implementation of the Method

We estimate the local error $E_k = u_n(x_{n+d}) - y_{n+d}$ at each step of the integration where $u_n$ is the solution of $u_n'(x) = f(x, u_n)$, $u_n(x_n) = y_n$ and $d = 2,3$. It is important that we control the error for the whole process of the code because the delay terms might require evaluating the numerical solution at any point prior to the current $x$. A general procedure for estimating the local errors is to compare the results of formulae of different orders.

Let $E_{d,k} = y_{n+d}(k+1) - y_{n+d}(k)$ denotes the estimated error in $y_{n+d}(k)$ at $x_{n+d}$ using the mode $P_k E C_{k+1} E$. Let $P_k$ and $C_{k+1}$ denote the explicit and implicit methods of order $k$ and $k+1$ respectively. The local error is potentially more accurate when using the corrector formulae. The convergence test involved the corrector formulae. The iterations are said to have converged when the iteration $y^{(i)}{}_{n+d}$ satisfies as follows

$$
| y^{(i)}{}_{n+d} - y^{(i-1)}{}_{n+d} | \; < \; 0.1 \times TOL \tag{10}
$$

where $i$ is the number of iterations. If the accuracy requirement is satisfied, the corrected difference for the estimate error is form to complete the step. Supposed that the integration step is accepted then the next step size has to be selected. In the program, we only allow constant or doubling the step size. If the integration step failed, it will reduce the step size by half. As a

result, it will repeat the integration step again until the end of the interval. The estimates for the maximum step sizes are as follows

$$\begin{aligned} &h_{next} = C \times \left( \frac{TOL}{2 \times LTE} \right)^{\left(\frac{1}{k+1}\right)} \times h_{old} \\ &if\,(h_{next} \geq 2 \times h_{old}) \text{ then } h_{next} = 2 \times h_{old} \\ &\text{else } h_{next} = h_{old} \end{aligned} \tag{11}$$

where C = 0.8 is a safety factor and $k+1$ is the order of the method. The purpose of having the safety factor is to avoid having too many rejected step. The algorithm when the step failure occurs is

$$h_{next} = \frac{1}{2} \times h_{old} \tag{12}$$

The test in Eq. (11) and Eq. (12) will allow the step to vary only by constant, half or doubling.

## 4. Stability Analysis

This section deals with the stability analysis of one step block method for the numerical solutions of delay differential equations (DDEs). We focused on finding the P- and Q-stability of the method applied to the following test equations,

$$\begin{aligned} y'(x) &= \lambda y(x) + \mu y(x-\tau), \qquad x > x_0 \\ y(x) &= \phi(x), \qquad x \leq x_0 \end{aligned} \tag{13}$$

and

$$\begin{aligned} y'(x) &= \mu y(x-\tau), \qquad x > x_0 \\ y(x) &= \phi(x), \qquad x \leq x_0 \end{aligned} \tag{14}$$

where $\lambda$ and $\mu$ are complex numbers. Consider a fixed step size $h$ such that $x_n = x_0 + nh$ and $mh = \tau$, $m \in I^+$ and letting $H_1 = h\lambda$ and $H_2 = h\mu$.

Definition 1. For a fixed step size $h$ and $\lambda, \mu \in R$ in (13), the region $R_P$ in the complex $H_1 - H_2$ plane is called the P-stability region if for any $(H_1, H_2) \in R_P$, the numerical solution of (13) vanishes as $x_n \to \infty$.

Definition 2. For a fixed step size $h$ and $\mu \in C$ in (14), the region $R_Q$ in the complex $H_2$-plane is called the Q-stability region if for any $H_2 \in R_Q$, the numerical solution of (14) vanishes as $x_n \to \infty$.

By letting $n = 2N$ in the two point block method and $n = 3N$ in the three point block method, we have

$$A_2 Y_{N+2} = A_1 Y_{N+1} + h \sum_{i=1}^{2} B_i F_{N+i} \tag{15}$$

Two point block method:

$$A_2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_2 = \frac{1}{12}\begin{bmatrix} 8 & -1 \\ 8 & 5 \end{bmatrix}, B_1 = \frac{1}{12}\begin{bmatrix} 0 & 5 \\ 0 & -1 \end{bmatrix}$$

$$Y_{N+2} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix}, Y_{N+1} = \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix}, F_{N+2} = \begin{bmatrix} f_{n+1} \\ f_{n+2} \end{bmatrix}, F_{N+1} = \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}$$

Three point block method:

$$A_2 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B_2 = \frac{1}{24}\begin{bmatrix} 19 & -5 & 1 \\ 13 & 13 & -1 \\ -5 & 19 & 9 \end{bmatrix}, B_1 = \frac{1}{24}\begin{bmatrix} 0 & 0 & 9 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix},$$

$$Y_{N+2} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \end{bmatrix}, Y_{N+1} = \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}, F_{N+2} = \begin{bmatrix} f_{n+1} \\ f_{n+2} \\ f_{n+3} \end{bmatrix}, F_{N+1} = \begin{bmatrix} f_{n-2} \\ f_{n-1} \\ f_n \end{bmatrix}$$

The elements of $B_i$ are the coefficients of the method itself. The P and Q stability polynomial are obtained by applying (15) to (13) and (14) respectively, where $h = \frac{\tau}{m}$, $m \in I^+$. Thus, the P-stability polynomial for the two and three point block method, $\psi(\varsigma)$ is given by

$$\psi(\varsigma) = \det\left[(A_2 - H_1B_2)\varsigma^{2+m} - (A_1 + H_1B_1)\varsigma^{1+m} - H_2\sum_{i=1}^{2} B_i\varsigma^i\right] \tag{16}$$

and the Q-stability polynomial $\pi(\varsigma)$ is given by the determinant of

$$(A_2\varsigma^{2+m} - A_1\varsigma^{1+m} - H_2\sum_{i=1}^{2} B_i\varsigma^i) \tag{17}$$

The P- and Q-stability regions for $m = 1$ for two and three point block method are shown in Figures 1 - 4 respectively.

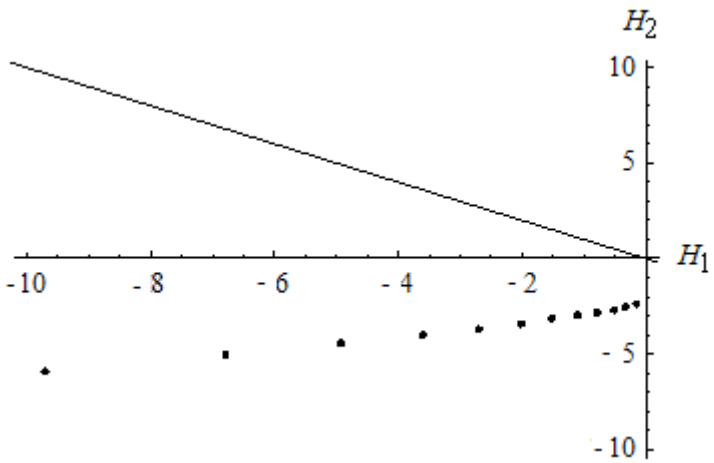


Figure 1: P-stability region for two point block method

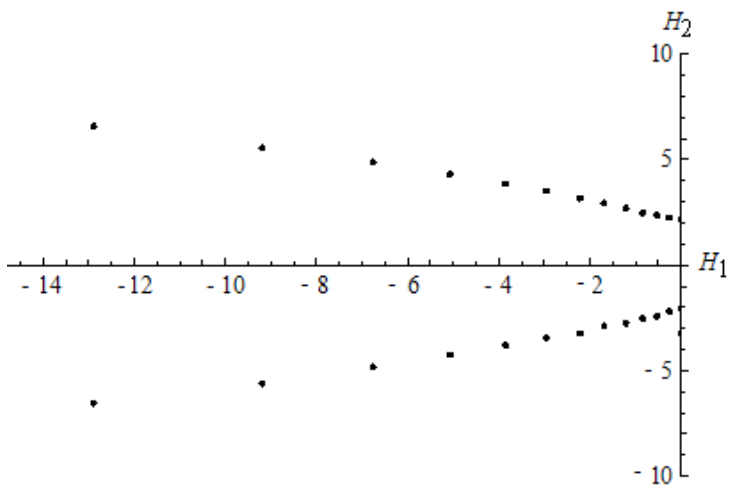


Figure 2: P-stability region for three point block method

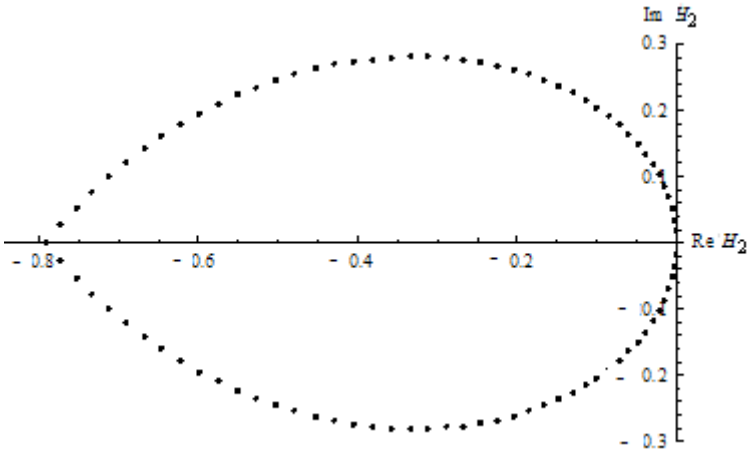


Figure 3: Q-stability region for two point block method

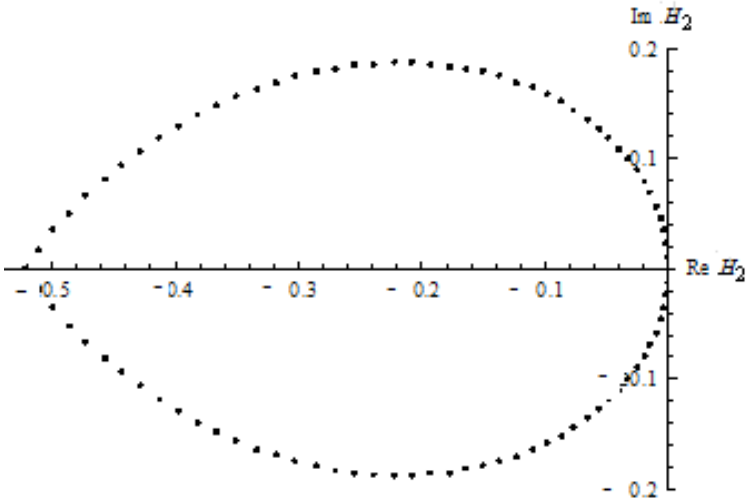


Figure 4: Q-stability region for three point block method

The P-stability regions of those methods lie inside the open ended region given in Figures 1 and 2. From the figures, it is observed that the regions are about the same when the method is implemented in two and three point one-step block method. The Q-stability regions of those methods lie inside the bounded region given in Figures 3 and 4. The region of Q-stability of the two point one-step block method is larger compared to the three point one-step block method.

## 5. Numerical Results

In order to study the efficiency of the developed codes, we present some numerical results for the following problems:

Problem 1

$$y'(x) = \cos(x) + y(y(x) - 2) \qquad 0 < x \le 10$$
$$y(x) = 1 \qquad x \le 0$$
$$\tau = x - y(x) + 2$$

Solution:
$$y(x) = \sin(x) + 1$$

Problem 2

$$y'(x) = -y(x - \frac{\pi}{2}) \qquad 0 < x \le 10$$
$$y(x) = \sin(x) \qquad x \le 0$$
$$\tau = \frac{\pi}{2}$$

Solution:
$$y(x) = \sin(x)$$

Problem 3

$$y_1'(x) = -y_1(x - \frac{\pi}{2}) \qquad \frac{\pi}{2} < x \le 10$$
$$y_2'(x) = -y_2(x - \frac{\pi}{2}) \qquad \frac{\pi}{2} < x \le 10$$
$$y_1(x) = \sin(x) \qquad x \le \frac{\pi}{2}$$
$$y_2(x) = \cos(x) \qquad x \le \frac{\pi}{2}$$
$$\tau = \frac{\pi}{2}$$

Solution
$$y_1(x) = \sin(x);\ y_2(x) = \cos(x);\ x \ge \frac{\pi}{2}$$

The numerical results obtained when the problems are solved using two and three point block method with Neville's interpolation to approximate the delay term are given in the Table 1-3.

| | |
|---|---|
| TOL | The chosen tolerance |
| STEP | The number of total steps |

| | |
|---|---|
| FSTEP | The number of failed steps |
| FN | The number of function evaluations |
| MAXE | The maximum error |
| MTD | Method employed |
| 2PS | The implementation of the two point one-step method for solving DDE |
| 3PS | The implementation of the three point one-step method for solving DDE |

Table 1: Comparison between the 2PS and 3PS methods for solving Problem 1

| TOL | MTD | STEP | FSTEP | FN | MAXE |
|---|---|---|---|---|---|
| $10^{-2}$ | 2PS | 28 | 1 | 158 | 1.2202e-3 |
| | 3PS | 19 | 0 | 156 | 7.4332e-4 |
| $10^{-4}$ | 2PS | 86 | 3 | 478 | 1.5845e-5 |
| | 3PS | 48 | 1 | 390 | 2.1487e-5 |
| $10^{-6}$ | 2PS | 304 | 6 | 1754 | 1.3168e-7 |
| | 3PS | 95 | 0 | 777 | 1.6189e-7 |
| $10^{-8}$ | 2PS | 1000 | 3 | 5918 | 6.9104e-10 |
| | 3PS | 253 | 3 | 2166 | 4.9942e-9 |
| $10^{-10}$ | 2PS | 4660 | 6 | 27848 | 1.8251e-11 |
| | 3PS | 589 | 4 | 5154 | 8.8202e-11 |

Table 2: Comparison between the 2PS and 3PS methods for solving Problem 2

| TOL | MTD | STEP | FSTEP | FN | MAXE |
|---|---|---|---|---|---|
| $10^{-2}$ | 2PS | 28 | 1 | 150 | 1.8636e-3 |
| | 3PS | 19 | 0 | 150 | 2.2937e-3 |
| $10^{-4}$ | 2PS | 89 | 3 | 492 | 2.6633e-5 |
| | 3PS | 43 | 0 | 336 | 1.0402e-4 |
| $10^{-6}$ | 2PS | 330 | 3 | 1912 | 3.5411e-7 |
| | 3PS | 95 | 0 | 768 | 8.2669e-7 |
| $10^{-8}$ | 2PS | 1406 | 9 | 8346 | 4.0224e-9 |
| | 3PS | 260 | 2 | 2217 | 1.0484e-8 |
| $10^{-10}$ | 2PS | 6337 | 9 | 37908 | 9.0128e-10 |
| | 3PS | 509 | 0 | 4422 | 7.4175e-10 |

Table 3: Comparison between the 2PS and 3PS methods for solving Problem 3

| TOL | MTD | STEP | FSTEP | FN | MAXE |
|---|---|---|---|---|---|
| $10^{-2}$ | 2PS | 24 | 0 | 132 | 2.2481e-3 |
| | 3PS | 18 | 0 | 168 | 2.0140e-3 |
| $10^{-4}$ | 2PS | 85 | 0 | 478 | 1.8285e-5 |
| | 3PS | 48 | 0 | 450 | 8.9451e-6 |
| $10^{-6}$ | 2PS | 338 | 0 | 1974 | 1.8665e-7 |
| | 3PS | 89 | 0 | 852 | 6.4954e-7 |

*To be continued...*

*.....Continuation*

| | | | | | |
|---|---|---|---|---|---|
| $10^{-8}$ | 2PS | 1541 | 0 | 9172 | 7.6762e-10 |
| | 3PS | 241 | 0 | 2541 | 1.7544e-9 |
| $10^{-10}$ | 2PS | 7440 | 0 | 44546 | 7.7745e-10 |
| | 3PS | 813 | 0 | 9024 | 7.8024e-10 |

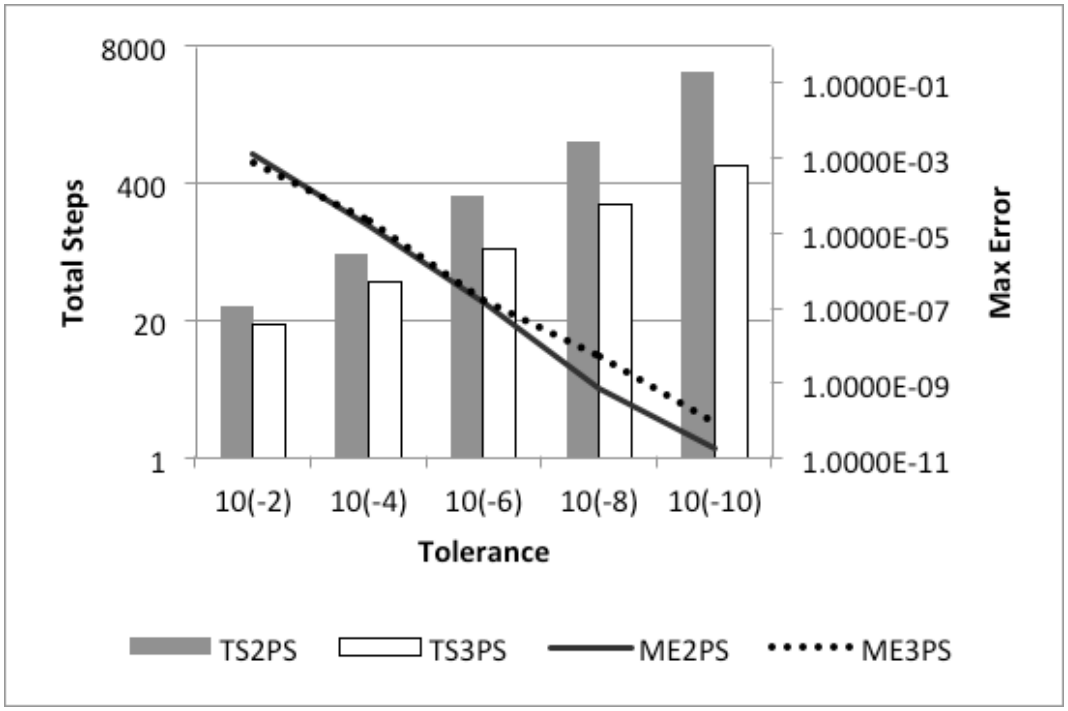


Figure 5: Comparison between the 2PS and 3PS methods for solving Problem 1

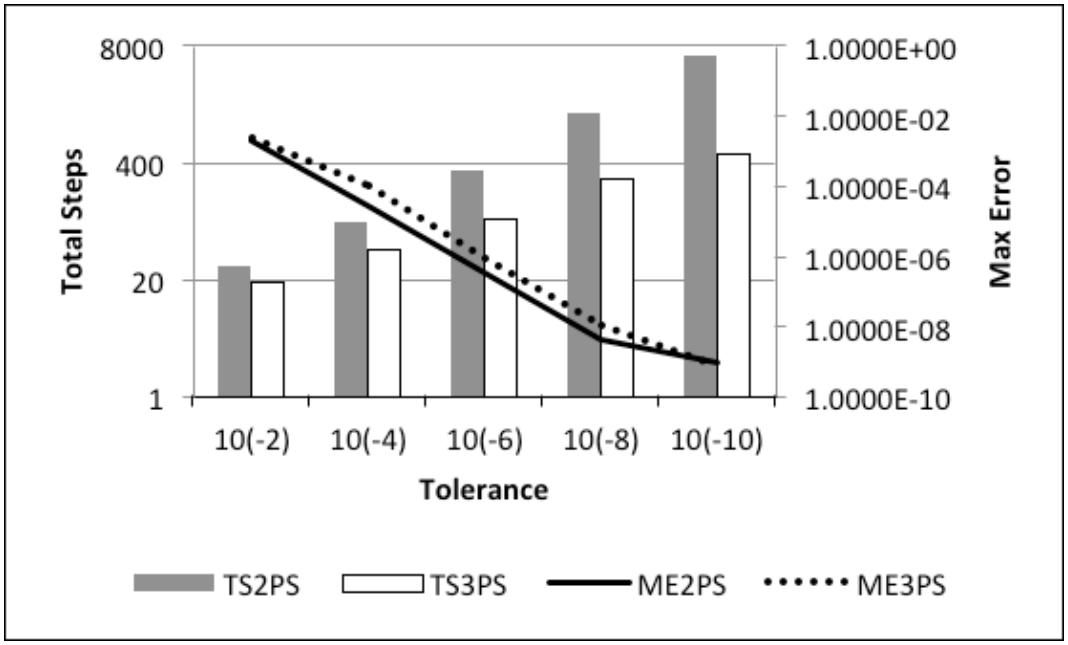


Figure 6: Comparison between the 2PS and 3PS methods for solving Problem 2

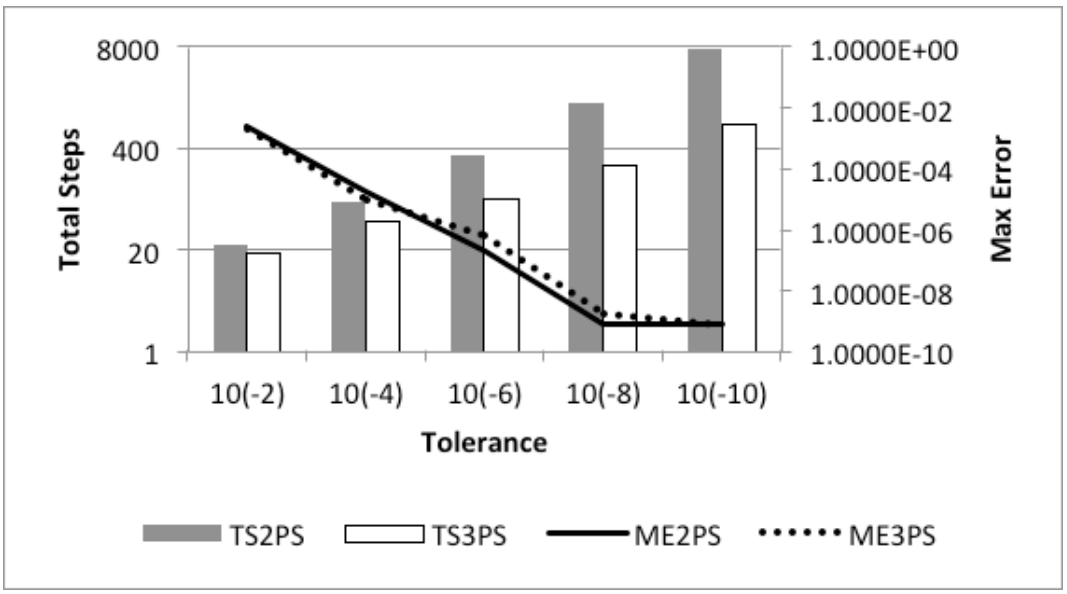


Figure 7: Comparison between the 2PS and 3PS methods for solving Problem 3

## 7. Conclusions

The numerical results in Table 1-3 clearly indicate that 3PS performs better in terms of total number of steps and less number of step failure compared to 2PS. The total function calls for 3PS is less than 2PS when solving all problems except for Problem 3 at tolerance $10^{-2}$. The computational cost at smaller tolerance obviously decreases when the codes are implemented in 3PS. Both of the methods achieved the desired accuracy in all tested problems. In Figures 5-7, the number of total steps is always less when the method is implemented in 3PS at all tolerances. However, the maximum error of 2PS seems to be smaller than 3PS at finer tolerances. The reason is because of the accumulation error occur since 3PS approximates three points simultaneously at each iteration. In general, it is shown that 3 point one-step block method is better than 2 point one-step block method when solving the tested problems.

## Acknowledgement

The authors gratefully acknowledge that this research was partially supported by Universiti Putra Malaysia under the Graduate Research Fellowship (GRF).

## References

Al-Mutib A.N.1977. Numerical Methods for Solving Delay Differential Equations. Ph.D Thesis. University of Manchester, United Kingdom.

Ibrahim M.A.-K., El-Safty A. & Abo-Hasha S.M. 1995. 2h-step spline method for the solution of delay differential equations. *Computers Math Applic* **29**(8): 1-6.

Ishak F., Suleiman M.B. & Omar Z. 2008. Two-point predictor-corrector block method for solving delay differential equations. *Matematika* **24** (2): 131-140.

Ismail F., Al-Khasawneh R.A, Lwin A.S. & Suleiman M.B. 2002. Numerical treatment of delay differential equations by Runge-Kutta method using Hermite interpolation. *Matematika* **18**: 79-90.

Majid Z.A., Suleiman M.B., Ismail F. & Othman M. 2003. 2-point implicit block one-step method half Gauss-Seidel for solving first order ordinary differential equations. *Matematika* **19**: 91-100.

Majid Z.A., Suleiman M.B. & Omar Z. 2006. 3-point implicit block method for solving ordinary differential equations. *Bulletin of the Malaysian Mathematical Sciences* **29**(1): 23–31

Mohamed A.R, Abd-El-Aziz E.S. & Mahmoud N.S. 2006. Numerical solution of system of first-order delay differential equations using polynomial spline functions. *International Journal of Computer Mathematics* **83**(12): 925–937

Oberle H.J. & Pesh H.J. 1981. Numerical treatment of delay differential equations by Hermite interpolation. *Numer. Math* **37**: 235–255.

Suleiman M.B. & Ismail.F. 2001. Solving delay differential equations using componentwise partitioning by Runge-Kutta method. *Applied Mathematics and Computation* **122**: 301-323.

Thompson S. 1990. Stepsize control for delay differential equations using continously imbedded Runge-Kutta methods of sarafyan. *Journal of Computational and Applied Mathematics* **31**: 267-275.

Zhang C. & Chen H. 2010. Block boundary value methods for delay differential equations. *Applied Numerical Mathematics* **60**: 915-923.

*Department of Mathematics*
*Faculty of Science*
*Universiti Putra Malaysia*
*43400 UPM Serdang*
*Selangor DE, MALAYSIA*
*E-mail: hazwani194@yahoo.com, zanariah@science.edu.upm.my*, mohameds@science.edu.upm.my*

*Institute for Mathematical Research*
*Universiti Putra Malaysia*
*43400 UPM Serdang*
*Selangor DE, MALAYSIA*
*E-mail: zanariah@science.upm.edu.my*, fudziah@science.upm.edu.my,*
*mohameds@science.upm.edu.my*

* Corresponding author