ACTA MATHEMATICA VIETNAMICA
Volume 33, Number 2, 2008, pp. 95-105

SOLVING DELAY DIFFERENTIAL EQUATIONS USING EMBEDDED SINGLY DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS

FUDZIAH ISMAIL AND RAED ALI AL-KHASSAWNEH

Abstract. In this paper we used three embedded diagonally implicit Runge-Kutta methods to solve a standard set of delay differential equations. Some of the methods share the FSAL (first stage as last) property and the same stage order. The Q-stability polynomials of the methods are obtained and their stability regions presented. Numerical results are tabulated and compared, from which we can conclude that the method with the highest order and the higher stage order is the most efficient.

1. Introduction

In recent years, there has been a growing interest in the numerical solutions of delay differential equations (DDEs). This is because of the appearance of such equations in various areas such as neural network theory, epidemiology and time lag control processes. DDEs also provide us with a realistic model of many phenomena arising in mathematics. For example, a DDE can be used for the modeling of population dynamics and the spread of infections diseases and two-body problems of electrodynamics. More examples can be found in Driver [1], Kuang [2] and Macdonald [3]. DDEs have the feature of having a delayed response to input conditions, so that the rate at which processes take place depends not only on the current state of the system but also the past states. Hence, the mathematical models commonly result in differential equations with a time delay. Most of the numerical methods for ordinary differential equations (ODEs) can be adapted to DDEs. Most of the codes developed for ODEs showed that the Runge-Kutta (RK) embedding technique is an efficient technique for the numerical integration of ODEs where each integration step is performed twice using the p-th and $(p+1)$-th order methods.

In this paper we are going to use the embedded singly diagonally implicit Runge-Kutta (SDIRK) methods to integrate DDEs and compare the numerical results. The word singly here means that all the eigenvalues of the coefficient matrix A are equal and for diagonally implicit RK method it means that all the diagonal elements are equal. We will use the term loosely also for method with

Received December 18, 2006; in revised form June 22, 2007.
Key words and phrases. Runge-Kutta method, delay differential equations, stage order.

the first diagonal element is zero. The stability of the method when applied to linear DDE will also be discussed.

2. Method

The family of embedded RK methods advances the integration from t_n to $t_{n+1} = t_n + h$, computing at each step two approximations y_{n+1} and $\bar{y}_{n+1}$to $y(t_{n+1})$ of order $(p+1)$ and (p) respectively, given by

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i, \quad \bar{y}_{n+1} = y_n + h\sum_{i=1}^{s-1} \bar{b}_i k_i,$$

where $k_i = t(t_n + c_i h, y_n + h\sum_{i=1}^{i} a_{ij}k_j\ )$, $i = 1, 2, .., s$. (Here s is the stage of the method.) The local truncation error is defined by

$$LTE = \left| y_{n+1} - \bar{y}_{n+1} \right|.$$

The value *LTE* gives the basis to find the next stepsize of the integration. Thus this technique is used for numerical integration using variable stepsize.

The coefficients of the embedded RK method can be written in table form as follows:

$$\begin{array}{c|c} C & A \\ \hline & b \\ & \bar{b} \end{array}$$

where $C = (c_1, c_2, .., c_s)$, $A = \{a_{ij}\}$, $b = (b_1, b_2, .., b_s)$ and $\bar{b} = (\bar{b}_1, \bar{b}_2, .., \bar{b}_{s-1})$. The method (p,q) means the method of order p and q-stage. Equations of order conditions associated with the order of the Runge-Kutta method which can be found in Butcher [4] are given in Table 1.

The first 2 equations need to be satisfied for the method of order two, the first 4 equations need to be satisfied for the third order method, for the fourth order method the first 8 equations have to be satisfied and for the fifth order method all 17 equations have to be satisfied. For the embedded (4,6) in (5,7) method we have to satisfy 8 equations for the fourth order and 17 equations for the fifth order method, thus a total of 25 equations needed to be fulfilled. To simplify the equations we used the simplifying assumptions and according to Butcher and Chen [5] if the simplifying assumptions

$$\sum_j a_{ij} c_j^2 = \frac{c_i^2}{2}, \quad i = 1, 2, .., s, \tag{2.1}$$

$$\sum_j a_{ij}c_j^3 = \frac{c_i^3}{3}, \quad i = 1, 2, .., s, \tag{2.2}$$

are satisfied then the stage order of the method is three; if only (2.1) is satisfied then the stage order is two.

TABLE 1

$\sum_i b_i = 1$	(2.1.1)	$\sum_{ij} b_i c_i^2 a_{ij} c_j = \frac{1}{10}$	(2.1.10)
$\sum_i b_i c_i = \frac{1}{2}$	(2.1.2)	$\sum_{ij} b_i c_i a_{ij} c_j^2 = \frac{1}{15}$	(2.1.11)
$\sum_i b_i c_i^2 = \frac{1}{3}$	(2.1.3)	$\sum_{ijk} b_i c_i a_{ij} a_{jk} c_k = \frac{1}{30}$	(2.1.12)
$\sum_{ij} b_i a_{ij} c_j = \frac{1}{6}$	(2.1.4)	$\sum_{ij} b_i a_{ij} c_j^3 = \frac{1}{20}$	(2.1.13)
$\sum_i b_i c_i^3 = \frac{1}{4}$	(2.1.5)	$\sum_{ijk} b_i a_{ij} c_j a_{jk} c_k = \frac{1}{20}$	(2.1.14)
$\sum_{ij} b_i c_i a_{ij} c_j = \frac{1}{3}$	(2.1.6)	$\sum_{ijk} b_i a_{ij} c_j a_{jk} c_k = \frac{1}{40}$	(2.1.15)
$\sum_{ij} b_i a_i c_j^2 = \frac{1}{12}$	(2.1.7)	$\sum_{ijk} b_i a_{ij} a_{jk} c_k^2 = \frac{1}{64}$	(2.1.16)
$\sum_{ijk} b_i a_{ij} a_{jk} c_k = \frac{1}{24}$	(2.1.8)	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{kl} c_l = \frac{1}{120}$	(2.1.17)
$\sum_i b_i c_i^4 = \frac{1}{5}$	(2.1.9)		

Using the two simplifying assumptions and the equations of order conditions for the fourth and fifth order method we derive the embedded diagonally implicit Runge-Kutta method of order (4,6) in (5,7). The method has a property whereby the first stage is zero and the last stage is equivalent to the vector output (b_i) or in the literature such methods are said to have FSAL (first stage as last) property. This property enables us to use the last k in the previous step as the first k in the current step. The method is given in Table 2

TABLE 2. DIRK Method (4,6) embedded in (5,7) with $\gamma = 0.28589$

0	0						
2γ	γ	γ					
$3\gamma + \gamma\sqrt{3}$	a_{31}	a_{32}	γ				
0.4	a_{41}	−0.04941651	−0.00450947	γ			
0.75	a_{51}	−0.11295160	−0.02779323	0.42253983	γ		
0.9	a_{61}	−0.42537807	−0.10703628	0.39570013	0.50326030	γ	
1	a_{71}	0	−0.01929017	0.53538626	0.23431316	−0.16631729	γ
	a_{71}	0	−0.01929017	0.53538626	0.23431316	−0.16631729	γ
	0.09438866	0	−0.03978261	0.74560855	−0.50512980	0.704915206	

Here the values of a_{i1} are given by $a_{i1} = c_i - \sum_{i=2}^{i} a_{ij} \quad (i = 1(1)7)$.

The method is then used to solve numerically DDEs given in Section 4 and compare the results with other methods such as the SDIRK (3,5) in (4,6) method which was derived by Butcher and Chen [5]. This method has the FSAL property with stage order two. Another method which we are going to use is (SDIRK) (4,5) in (5,6) method due to Ismail et al [6]. This method does not have the FSAL property and the stage order is two.

3. Implementation

In the general form, the first order delay differential equations (DDEs) can be written as

$$\begin{cases} y'(t) = f(t, y(t), y(t - \tau(t, y(t))) & \text{for } t \geqslant 0 \\ y(t) = \varphi(t): & \text{for } t \leqslant 0. \end{cases} . \tag{3.1}$$

The function $\tau(t, y(t))$ is called the delay, $t - \tau(t, y(t))$ is called the delay argument, the value of $y(\tau(t, y(t)))$ is the solution of the delay term. When the Runge-Kutta method is applied to DDE (3.1), the following equations are obtained:

$$k_i = f(t_n + c_i h, y(t_n + c_i h), y(t_n + c_i h - \tau))$$

or

$$k_i = f\left(t_n + c_i h, y\left(t_n + h\sum_{i=1}^{i} a_{ij}k_j\right), y(_n + c_i h - \tau)\right),$$

$$y_{n+1} = y_n + h\sum_{i=1}^{q} b_i k_i,$$

where $(t_n + c_i h - \tau)$ is the delay argument and interpolation is needed to approximate the value of the delay term $y(t_n + c_i h - \tau)$. Here the delay terms are approximated using divided difference interpolations.

According to Orbele and Pesch [7], the interpolation order and thereby the number of support points have to be adapted to the order of the method. If the numerical method is of order p then the interpolation order must be equal or greater than p. Since we are using divided difference interpolation the support points or the number of y_i used must be at least $p + 1$. Having that in mind for method of order five we used values of y at six points to approximate the delay term so that the divided difference interpolation is also of order five. Applying Butcher's method which is of order four we used five values of y for the divided difference interpolation so that the order of the interpolation is four which is equal to the order of the method itself.

The support points are assumed to be chosen such that they satisfy the following two conditions: they do not cross discontinuities and the delay argument under consideration should be close to the centre of the support points chosen. If $t - \tau$ falls in between y_i and $y_{i+1}$for any $i < n - 2$, then we use the values of $y_{i-2}, y_{i-1}, y_i, y_{i+1}, y_{i+2}$ and y_{i+3} for the interpolation. If $t - \tau$ is less than the initial point then we take the value of the initial function.

In this paper for all the examples which have discontinuities, the points of discontinuities are known and they can be treated by taking the point of discontinuity as the mesh point as proposed by Al-Mutib [8].

4. Q-stability for sdirk method

When Runge-Kutta method is applied to the test equation

$$\begin{cases} y\prime(t) = \mu y(t-\tau), & (t>0) \\ y(t) = \varphi(t), & (t \in [-\tau, 0]) \end{cases} \tag{4.1}$$

using the Lagrange interpolation to approximate the delay term, we have

$$k_{n+1}^{(i)} = \mu \sum L_l(c_i) y_{n-m+l}, \quad y_{n+1} = y_n + hb^T \underline{k}_{n+1}$$

or

$$h\underline{k}_{n+1} = h\mu \sum L_l(c) y_{n-m+l}, \quad y_{n+1} = y_n + h\mu b^T \sum L_l(c) y_{n-m+l}.$$

Let $Y_n = (y_n, h\underline{k}_n)^T$. The Q-stability polynomial of the method is

$$\det[IY_{n+1} - 1Y_n - VY_{n-m+l}],$$

where

$$\mathbf{1}= \begin{bmatrix} 1 & 0, & .., & 0 \\ 0 & 0, & .., & 0 \\ : & & & : \\ 0 & 0, & .., & 0 \end{bmatrix}, \qquad \mathbf{V}= \left[\begin{array}{c|c} h\mu b^T \sum L_l(c) & 0 \\ \hline h\mu \sum L_l(c) & \begin{matrix} 0 \\ : \\ 0 \end{matrix} \end{array}\right].$$

Taking $l = -3(1)2$, the Q-stability polynomial of the method can be written as

$$\begin{aligned} S(\beta,\zeta) &= \zeta^{m+4} - \zeta^{m+3} - \beta b^T (L_{-3}(c) + L_{-2}(c)\zeta + L_{-1}(c)\zeta^2 \\ &\quad + L_0(c)\zeta^3 + L_1(c)\zeta^4 + L_2(c)\zeta^5). \end{aligned} \tag{4.2}$$

The Q-stability polynomial of the SDIRK (3,4) in (4,5) method is

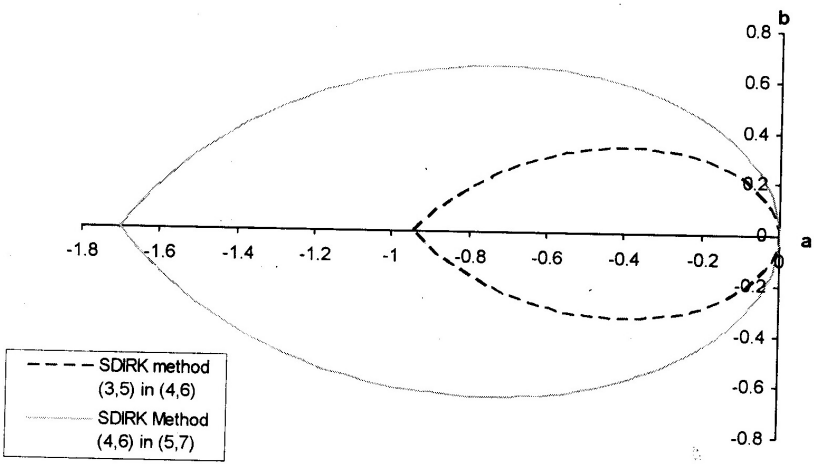
$$\begin{aligned} S(\beta,\zeta) = \; & \zeta^{m+3} - \zeta^{m+2} - \beta b^T (\quad L_{-2}(c) + L_{-1}(c)\zeta + L_0(c)\zeta^2 + L_1(c)\zeta^3 \\ & + L_2(c)\zeta^4) \times L_0(c)\zeta^3 + L_1(c)\zeta^4) + L_2(c)\zeta^5), \end{aligned} \tag{4.3}$$

where $\beta = h\mu$.

Locating the Boundary of the Q-Stability Region. Equating the stability polynomial to zero with $m = 1$ and taking $\zeta = \cos(\theta) + \sin(\theta)$ for $0 \leqslant \theta \leqslant 2\pi$, we solve for values of β to give the stability region of the method (here a is the real part and b is the imaginary of β), see Table 3.

The Q-stability region of SDIRK (4,5) in (5,6) method also have the same shape and cuts the negative real axis at -1.5 the region can be seen in Ismail *et al* [9]. Thus comparing the regions of Q-stability, SDIRK method (4,6) in (5,7) has the largest region of stability followed by SDIRK method (4,5) in (5,6) and

TABLE 3. The Q-stability region of embedded SDIRK method (3,5) in (4,6) and SDIRK method (4,6) in (5,7)



method (3,5) in (4,6) has the smallest region of Q-stability which are shown in Fig. 1.

5. TESTED PROBLEMS AND NUMERICAL RESULTS

In this section, some of the delay differential equations given in Al-Mutib [8] are tested upon. They are solved using the three methods and the delay terms are approximated using divided difference interpolation. The results are given in Tables 4.1-4.6.

Problem 5.1.

$y\prime(t) = -y(t - \tau(t))) + 1, \quad 1 \leqslant t \leqslant 10,$
$y(t) = \ln(t), \quad 0 < t \leqslant 1,$
$\tau(t) = t - \exp\left[1 - \frac{1}{t}\right],$
Solution: $y(t) = \ln(t), \quad t \geqslant 1.$

Problem 5.2.

$y\prime(t) = -y(t - 1 + \exp(-t) + \sin(t - 1 + \exp(-t)) + \cos(t)), \quad 0 \leqslant t \leqslant 10,$
$y(t) = \sin(t), \quad t \leqslant 0,$
$\tau(t) = t - \exp(-t),$
Solution: $y(t) = \sin(t).$

Problem 5.3.

$y\prime(t) = \cos(t)y\left(y(t) - 2\right), \quad 0 \leqslant t \leqslant 10,$
$y(t) = 1, \quad t \leqslant 0,$

$\tau(t) = t - y(t) + 2,$

Solution: $y(t) = \sin(t) + 1.$

Problem 5.4.

$y\prime(t) = \frac{1}{t}\exp(y(y(t) - \ln(2) + 1), \quad 1 \leqslant t \leqslant 3,$

$y(t) = 0, \quad t \leqslant 1,$

$\tau(t) = t - y(t) + \ln(2) - 1,$

Solution:

$$y(t) = \begin{cases} \ln(t), & 1 \leqslant t \leqslant 2, \\ \frac{t}{2} + \ln(2) - 1, & 2 \leqslant t \leqslant 3. \end{cases}$$

Problem 5.5.

$y\prime(t) = \frac{1}{t}y(t)y(\ln(y(t))), \quad 1 \leqslant t \leqslant 10,$

$y(t) = 1, \quad t \leqslant 1,$

$\tau(t) = t - \ln(y(t)),$

$$\mathrm{y(t)}= \begin{cases} t, & 1 \leqslant t \leqslant e, \\ \exp\left(\frac{t}{e}\right), & e \leqslant t \leqslant e^2. \end{cases}$$

Problem 5.6.

$$\left.\begin{array}{l} y\prime_1(t) = y_2(t), \\ y\prime_2(t) = -y_2(\exp(1 - y_2(t))(y_2(t))^2\exp(t - y_2(t)), \end{array}\right\}, 1 \leqslant t \leqslant 10,$$

$$\left.\begin{array}{l} y_1(t) = \ln(t), \\ y_2(t) = \frac{1}{t} \end{array}\right\}, 1 \leqslant t \leqslant 10,$$

$\tau(t) = t - \exp(1 - y_2(t)),$

Solution:

$$\left.\begin{array}{ll} y_1(t) = \ln(t), & 1 \leqslant t \leqslant 2, \\ y_2(t) = \frac{1}{t}, & 2 \leqslant t \leqslant 3. \end{array}\right\}$$

Tables 4 – 9 give the numerical results when the problems are solved using the three methods on personal computer Pentium 4. The notations used are as follows:

Max Error – max $|y(t_i) - y_{t_i}|$, (absolute value of the true solution minus the computed solution at the mesh point i).

The notation 6.455378E-5 means 6.455378×10^{-5}.

Tolerance $\sim$ the chosen tolerance,

Function $\sim$ the number of functions evaluations,

Steps $\sim$ the number of successful steps,

Failed steps $\sim$ the number of failed steps.

Methods used are:

S1: SDIRK method (3,5) embedded in (4,6) by Butcher and Chen [5].

S2: SDIRK method (4,5) embedded in (5,6) by Ismail and Suleiman [6].

S3: SDIRK method (4,6) embedded in (5,7) in Section 2.

6. Discussions and conclusions

We have used the embedded singly diagonally implicit Runge-Kutta method to solve delay differential equations. The Q-stability polynomial of the methods are obtained and the stability regions are traced, from which we can conclude that SDIRK (4,6) in (5,7) or S3 method has the largest stability region, thus is more stable compared to SDIRK (3,5) in (4,6) and SDIRK (4,5) in (5,6) methods. This is in line with the numerical results, whereby it was observed that solving the problems using method S3 generally required less number of steps, function evaluations and produced smaller error compared to S1 and S2. Furthermore, the stage order for method S3 is three and stage order for S2 is two whereas S1 is of order four and has stage order two. As a conclusion we can say that S3 is more efficient compared to S1 and S2 in solving DDEs, which confirmed that method with the highest order and highest stage order together with the largest stability region is the most efficient.

Table 4. Numerical results when Problem 4.1 is solved by S1, S2, S3 methods

Method	Functions	Steps	FailedSteps	MaxError	Tolerance
S1	192	8	1	$4.005268025E-01$	
S2	122	6	0	$3.575440973E-01$	10^{-2}
S3	119	5	0	$3.179314934E-01$	
S1	274	20	1	$5.883856656E-02$	
S2	242	16	0	$4.773087487E-02$	10^{-4}
S3	227	15	0	$4.080798302E-02$	
S1	556	51	0	$1.129204886E-04$	
S2	482	32	0	$5.555526510E-05$	10^{-6}
S3	464	31	0	$7.585214276E-05$	
S1	2817	161	0	$2.545746763E-07$	
S2	2371	157	0	$3.963515930E-07$	10^{-8}
S3	2371	149	0	$8.146055129E-08$	
S1	3291	253	0	$3.232046894E-10$	
S2	3137	209	0	$9.545175175E-11$	10^{-10}
S3	3019	201	0	$8.297344889E-11$	

TABLE 5. Numerical results when Problem 4.2 is solved by S1, S2, S3 methods

Method	Functions	Steps	Failed Steps	Max Error	Tolerance
S1	167	15	0	$4.773789105E-03$	10^{-2}
S2	132	10	0	$9.981939676E-04$	
S3	132	10	0	$8.808848767E-04$	
S1	513	52	0	$2.558796656E-05$	10^{-4}
S2	379	29	0	$1.117413289E-06$	
S3	444	34	0	$1.097476185E-06$	
S1	871	79	0	$3.485556410E-07$	10^{-6}
S2	717	55	0	$5.848431978E-08$	
S3	769	59	0	$4.334860184E-08$	
S1	2036	194	0	$1.053121906E-09$	10^{-8}
S2	1536	118	0	$4.969877758E-10$	
S3	1321	114	0	$2.342595129E-10$	
S1	4293	381	0	$1.435745782E-10$	10^{-10}
S2	3373	278	0	$3.742946072E-11$	
S3	3346	259	0	$3.645485419E-11$	

TABLE 6. Numerical results when Problem 4.3 is solved by S1, S2, S3 methods

Method	Functions	Steps	Failed Steps	Max Error	Tolerance
S1	155	13	0	$2.067649455E-03$	10^{-2}
S2	145	11	0	$2.247058358E-03$	
S3	132	10	0	$8.779047485E-04$	
S1	566	68	1	$6.637884738E-06$	10^{-4}
S2	535	41	0	$4.809784965E-07$	
S3	496	38	0	$4.595800416E-07$	
S1	1410	95	0	$9.549054721E-08$	10^{-6}
S2	1120	86	0	$9.366192725E-09$	
S3	1094	84	0	$8.340937942E-09$	
S1	1817	119	0	$1.106146053E-09$	10^{-8}
S2	1536	118	0	$4.969867593E-10$	
S3	1371	111	0	$3.764546763E-10$	
S1	3819	287	0	$8.297344889E-11$	10^{-10}
S2	3733	247	0	$2.035678808E-11$	
S3	3602	206	0	$9.633550011E-12$	

TABLE 7. Numerical results when Problem 5.4 is solved by S1, S2, S3 methods

Method	Functions	Steps	Failed Steps	Max Error	Tolerance
S1	96	10	0	$8.516790347E-03$	10^{-2}
S2	92	6	0	$3.237686445E-04$	
S3	91	6	0	$3.106886391E-04$	
S1	168	18	0	$9.478237678E-04$	10^{-4}
S2	158	13	0	$1.156477843E-04$	
S3	160	12	0	$9.561245979E-05$	
S1	227	21	1	$2.675345672E-05$	10^{-6}
S2	210	16	1	$8.672341237E-06$	
S3	206	16	1	$3.458930542E-06$	
S1	561	38	1	$3.568345163E-07$	10^{-8}
S2	431	33	1	$7.849233698E-08$	
S3	423	32	1	$4.548338954E-08$	
S1	898	71	3	$2.789304535E-09$	10^{-10}
S2	852	65	3	$5.385678965E-10$	
S3	846	63	3	$4.766785337E-10$	

TABLE 8. Numerical results when Problem 5.5 is solved by S1, S2, S3 methods

Method	Functions	Steps	Failed Steps	Max Error	Tolerance
S1	254	18	3	0.4690346576	10^{-2}
S2	140	10	2	0.2128879148	
S3	239	17	3	0.1805448397	
S1	352	25	3	$2.854656656E-02$	10^{-4}
S2	309	23	2	$2.709504361E-02$	
S3	331	17	2	$6.657089799E-03$	
S1	809	55	3	$9.557436410E-03$	10^{-6}
S2	599	45	3	$8.707365048E-03$	
S3	770	50	2	$6.549407802E-04$	
S1	2022	164	5	$1.765872191E-07$	10^{-8}
S2	1985	151	5	$1.092416455E-08$	
S3	1678	108	5	$4.427337539E-08$	
S1	3853	381	8	$1.067750282E-05$	10^{-10}
S2	3332	278	8	$2.801482230E-06$	
S3	3264	259	8	$2.542212673E-06$	

TABLE 9. Numerical results when Problem 4.6 is solved by S1, S2, S3 methods

Method	Functions	Steps	Failed Steps	Max Error	Tolerance
S1	197	8	1	$1.804453916E-01$	
S2	185	6	1	$6.457645453E-02$	10^{-2}
S3	182	5	1	$2.563707537E-02$	
S1	307	20	0	$4.622309856E-03$	
S2	296	16	0	$8.564436834E-04$	10^{-4}
S3	292	15	0	$1.564989832E-04$	
S1	731	51	0	$4.783834842E-04$	
S2	660	32	0	$3.645685634E-05$	10^{-6}
S3	669	31	0	$1.349783452E-05$	
S1	1836	161	1	$2.554809816E-06$	
S2	1534	157	1	$6.673397603E-07$	10^{-8}
S3	1429	149	1	$5.539973462E-07$	
S1	3791	253	0	$1.557355842E-07$	
S2	3573	209	0	$5.175869834E-08$	10^{-10}
S3	3346	201	0	$9.645485419E-09$	

REFERENCES

[1] R. D. Driver, *Ordinary and Delay Differential Equations*, Springer-Verlag, 1977.

[2] Y. Kuang, *Delay Differential Equations with Applications in Population Dynamics*, Academic Press, 1993.

[3] N. Macdonald, *Biological Delay System: Linear Stability Theory*, Cambridge University Press, 1989.

[4] J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, Runge-Kutta and General Linear Methods, John Wiley and Sons, 1987.

[5] J. C. Butcher and D. L. Chen, *A new type of singly diagonally implicit Runge-Kutta method*, Applied Numerical Mathematics **34** (2000), 174-188.

[6] F. Ismail and M. B. Suleiman, *Embedded singly diagonally implicit Runge-Kutta method (4,5) in (5,6) for the integration of stiff ODEs*, Intl. J. of Computer Math. **66** (1998), 325-341.

[7] H. J. Orbele and H. J. Pesh, *Numerical treatment of delay differential equations by Hermite interpolation*, Numerical Math. **37** (1981), 235-255.

[8] A. N. Al-Mutib, *Numerical Methods for Solving Delay Differential Equations*, Ph. D. Thesis, University of Manchester, 1977.

[9] F. Ismail and M. B. Suleiman, *The P-stability and Q-stability of singly diagonally implicit Runge-Kutta method for delay differential equations*, Intl. J. of Computer Math. **76** (2000), 267-277.

DEPARTMENT OF MATHEMATICS
UNIVERSITY PUTRA MALAYSIA 43400 SERDANG
SELANGOR, MALAYSIA

E-mail address: fudziah_i@yahoo.com.my

DEPARTMENT OF ECONOMICS
ZARQA PRIVATE UNIVERSITY
P. O. BOX 2000, ZARQA 1311, JORDAN.

E-mail address: r.alkhasawneh@zpu.edu.my