International Journal of Engineering, Science and Mathematics
Vol.6 Issue 7, Month 2017,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Singly Diagonally Implicit Runge-Kutta fourth and fifth order methods for Solving Delay Differential Equations

K.Ponnammal*
R.Sayeelakshmi**

Abstract

This paper presents Singly Diagonally Implicit Runge-Kutta method based on cubic-spline Interpolation for the numerical solution of delay differential equation.The advantage of implicit Runge-Kutta methods is in their superior stabilitycompared with the explicit methods, more so when applied to food limited equations.Our objective is to develop a scheme for solving Delay Differential equation using singly diagonally implicit Runge-Kutta method of 4^{th} order and 5^{th} order with the food limited population model equation and discussed about stability region. Errors of numerical results are compared with exact solution. This study is implemented with matlab programming.

Keywords:

Diagonally Implicit;
Runge-kutta;
cubic spline;
Delay Differential Equations;

Author correspondence:

*K.Ponnammal,
Assistant Professor, Department of Mathematics
Periyar E.V.R. College, Tiruchirappalli-620 023
Email:ponnammal_k@yahoo.co.in

**R.Sayeelakshmi
Assistant professor, Department of Mathematics
MookambigaiCollege of Engineering, Keeranur, Pudukkottai-622 502
Email: sayeelakshmi10@gmail.com

1. Introduction

A delay differential equation (DDE) is a differential equation in which the derivativeof the function at any time depends on the solution at elapsed time. Introduction of delay in the model improves its dynamics and allows a precise description of the real life phenomena. Delay differential equations (DDEs) widely used in ecology, physiology and many other areas of applied science. DDE's are large and important class of dynamical systems. The parameters in the DDE model are often unknown. Thus it is of great interest to estimate DDE parameters. The premier hindrance when solvingdelay differential equation is the way to treat the delay argument $y(t-\tau(t))$ which depends on the past values of the solution. Several attempts have made by Bellenand Zennaro (2003) to calculate the value of delayed argument using the DDE itself. However, with this approach the number of calculations increases drastically with time [4]. The other approach to calculate the delayed argument is interpolation which is used in this paper. Baker and Paul (1994) discussed the practical implementation of explicit and implicit RK-methods applied to linear delay differential equations [2]. Baker et al. (1995) provided references on various aspects of Delay Differential Equations [3]. Some of the application areas of delay differential equations are population dynamics, infectious disease, physiological and pharmaceutical kinetics, chemical kinetics, models of conveyor belts, urban traffic, heat exchangers, robotics, navigational controlof ships and aircrafts, and more general control problems. A delay is introduced in most applications of life sciences; to name one application population growth rate in population modeling is subject to delays. The first principle of population dynamicsis widely regarded as the exponential law of Malthus, as modeled by the Malthusiangrowth model Weisstein (2003)[20]. The early period was dominated by demographic studies such as the work of Benjamin Gompertz and Pierre Franois Verhulst in the early 19^{th} century, who refined and adjusted the Malthusian demographic model, introduced the Logistic model is a model of population growth. Baca¨er (2011) and Kuang (1993) described logistic differential equation of Pierre Verhulst [1, 16]. The continuous time of the logistic growth is described as the differential equation

$$\frac{dx}{dt} = rx(t)\left[1 - \frac{x(t)}{k}\right] \tag{1}$$

Where, r is the Malthusian parameter (rate of maximum population growth) and K is the so called carrying capacity of the population and x(t) is the population of the species. The logistic equation ignores other factors and focuses only on a species and its growth, logistic equation gives us the rate at which a population must decrease as it grows. The Logistic ordinary differential equation is simple and used to analyze and find the solution for different initial populations using the separable method. Researchers proposed various results in their literatures are inaccurate since the logistic equation making the assumption that the birth and death of members of the population in a species is immediately reflected in the birth and growth rate of population. In reality many species have a delay between the stages when they are sexually mature and can reproduce. The normal Logistic equation cannot reflect it, but it is possible to model this occurrence of lags or delays into the logistic equations. The logistic model assumes that the growth rate of a population at any time t depends on the relative number of individuals at a time. The process of reproduction is not instantaneous. Hutchinson (1948)[13] modified the logistic equation as:

$$\frac{dx}{dt} = rx(t)\left[1 - \frac{x(t-\tau)}{k}\right], \quad t \geq 0 \tag{2}$$

where, r and K are positive constants and make a delay. Several authors investigated Hutchinson’s equation. Gopalsamy et al. (1988) studied the logistic delay differential equation and gave sufficient condition for the oscillation and non-oscillation of equation [12]. Smith (1963) proposed an alternative to the logistic equation for a food-limited population as:

$$\frac{dx}{dt} = rx(t)\left[\frac{k - x(t)}{k + crx(t)}\right] \tag{3}$$

where x(t), r and k are the mass of population, the rate of increase with unlimited food and the value of x(t) at saturation[19]. The constant 1/c is the rate of replacement of the mass in the population at saturation. Gopalsamy (1992) founded the oscillation criteria for the autonomous delay food-limited equation [11].

$$\frac{dx}{dt} = rx(t)\left[\frac{k - x(t-\tau)}{k + crx(t-\tau)}\right], t \geq 0 \tag{4}$$

In recent years, research has carried out to solve DDEs using Explicit or Implicit Runge-Kutta (RK) method with Hermite Interpolation. Oberle and Pesch (1981) developed a class of numerical methods for the treatment of delay differential equations [17]. The study combined Runge-Kutta methods and Hermite interpolation of appropriate order for the numerical solution of retarded initial value problems with one constant delay. The delayed term is approximated by a three-point Hermite polynomial. Such work can be found in Bellen and Zennaro (2003) and Fudziah et al. (2002) [4, 9]. There are a number of techniques for obtaining the approximations which have been discussed. Shampine and Thompson (2001) developed the popular MATLAB-based dde23 for delay differential equations is well tested and user-friendly [18]. Bin

Suleiman and Ismail (2001) proposed Embedded Singly Diagonally Implicit Runge Kutta (SDIRK) method [6]. Bellen et al. (1988) and Fudziah et al. (2002) and Oberle and Pesch (1981), Goodman and Feldstein (1973) and Ismail and Alkhaswneh (2007) used hermite interpolation to approximate delay terms [5, 9, 17, 10, 15]. Fudziah et al. (2002) used divided difference interpolation and Hermite interpolation. In'tHout (1992) introduced a new interpolation procedure which leads to numerical processes that satisfy an important asymptotic stability condition [14]. Zennaro (1986) studied the asymptotic stability properties using test equations [21].

2. Methodology

In this paper, we find out the numerical solution of a non-autonomous food-limited equation with a constant delay. Most of the numerical methods for ordinary differential equations (ODEs) can be adapted to DDEs. The objective of this work is to derive a solution for the first order delay differential equation using Singly Diagonally implicit Runge-Kutta (SDIRK) method of order four and of order five with Cubic-Spline interpolation and discussed about stability regions. The value of delay term is approximated using Cubic-Spline Interpolation on the interval $[t_0-\tau, t_0],[t_0,t_0+\tau]$.

3. ODE and SDIRK methods

In general, a system of first-order ODE has the general form

$$\begin{aligned} &y' = f(x, y(x)) \\ &where,\ y(x) = (\mathrm{y}_1(x), \mathrm{y}_2(x).....\mathrm{y}_m(x))^T, \\ &f(x, y(x)) = (\mathrm{f}_1(x, y(x)), \mathrm{f}_2(x, y(x)).....\mathrm{f}_m(x, y(x)))^T, \end{aligned} \tag{5}$$

and m is the order of the system. The necessary condition for the equation(5) to have a unique solution is there be m associated conditions which specify the solution in some way for one or more values of x. In general, if all conditions mentioned at the initial point, say, y $(x_0) = y_0$. Jawias (2010) constructed the order equation for fourth order and fourth stage SDIRK method [22].

$$\begin{array}{c|cccc} c_1 & \gamma & & & \\ c_2 & a_{21} & \gamma & & \\ c_3 & a_{31} & a_{32} & \gamma & \\ c_4 & a_{41} & a_{42} & a_{43} & \gamma \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array}$$

Table 1: Butcher coefficients for a SDIRK method with 4 stages

It can be written in Butcher's Tableau [7, 8] as

0.20	0.20			
0.05	0.15	0.20		
0.40	0.78518518518519	0.98518518518519	0.20	
0.80	0.70671936788942	0.19819311123659	0.09147374364765	0.20
	0. 4074074074074	0.4074074074074	0.10714285714286	0.46851851851852

Table 2: Butcher tableau for a SDIRK method with 4th order 4th stage

Similarly, Now for the Runge Kutta fifth order fifth stage method the Butcher tableau described in Fudziah (2009) [23] as

0.07012572	0.07012572				
0.28506286	0.21493713	0.07012572			
0.48130895	0.14706690	0.26411633	0.07012572		
0.70483201	0.16565616	0.18423756	0.28481256	0.07012572	
0.92987427	0.17034709	0.26544595	0.09769835	0.32625715	0.07012572
	0.16938785	0.21992349	0.19374055	0.24674849	0.17019960

Table 3: Butcher tableau for a SDIRK method with 5th order 5th stage

When a q-stage Single Diagonally Implicit Runge-Kutta (SDIRK) methods are used to solve equation (5) at the point t_{n+1}, the following equations are obtained;

$$
\begin{aligned}
k_1 &= f(t_n + c_1 h, y_n + h a_{11} k_1) \\
k_2 &= f(t_n + c_2 h, y_n + h a_{21} k_1 + h a_{22} k_2) \\
&\vdots \\
k_i &= f(t_n + c_i h, y_n + h \sum_{j=1}^{i} a_{ij} k_j) \\
y_{n+1} &= y_n + \sum_{i=1}^{q} b_i k_i \quad (i = 1, 2, \ldots q)
\end{aligned}
\qquad (6)
$$

$y_n + h \sum_{j=1}^{q} a_{ij} k_j$ is called the internal stage of the method.

4. DDE and SDIRK methods

The system of DDE has only one delay term or multiple delays has the general form

$$\begin{aligned} y'(t) &= f(t, y, y(t-\tau)) \quad for\ t > t_0 \\ y(t) &= \phi(t) \quad for\ t \le t_0 \end{aligned} \tag{7}$$

$\phi(t)$ is the history initial function, the function τ (t, y(t)) is called the delay, (t- τ) called the delay argument, the value of y(t - τ (t,y(t))) is the solution of the delay term. The delay is classified as constant delay, time dependent and state dependent.

When the Runge-Kutta method is applied to DDE, the equation (6) becomes

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

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

$y(t_n+c_ih-\tau)$ is the delay term and interpolation is needed to approximate the value. Single diagonally implicit Runge-Kutta (SDIRK) schemes are a subclass of DIRK methods. Researchers have chosen the diagonally implicit Runge-Kutta method, because of the excessive cost in evaluating the stages in a fully implicit Runge-Kutta method. The diagonal in the coefficient matrix are identical in SDIRK methods, but not necessarily having equal diagonals with DIRK methods. The proposed work uses Singly Diagonally Implicit Runge-Kutta methods of 4th and 5th order to get the solution of delay differential equations.

The SDIRK method has constant values on the diagonal of the Butcher tableau, i.e, $a_{ii}=\gamma$ and is optimized for fast convergence of a non-linear solver. Thus when an iterative solver is used, the number of iterations needed to get the converged solution for each stage is lower. The continuous extension of an RK process is found as one of the standard techniques for obtaining dense output in the solution for ODE. It is based upon the continuous RK Tableau

$$\begin{array}{c|c} C & A \\ \hline & b^T(\theta) \end{array} = \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & a_{13} & a_{14} \\ c_2 & a_{21} & a_{22} & a_{23} & a_{24} \\ c_3 & a_{31} & a_{32} & a_{33} & a_{34} \\ c_4 & a_{41} & a_{42} & a_{43} & a_{44} \\ \hline & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) \end{array}$$

which defines as RK process for advancing θh, where $\theta \in (0,1]$.

$$a_{ij} = \int_0^{c_i} l_j(\xi)d\xi, i, j = 1,....v$$

$$b_i(\theta) = \int_0^{\theta} l_j(\xi)d\xi, i, j = 1,....v$$

$l_j(\xi)$ being the Lagrange polynomial coefficient $\prod_{k=1, k\neq i}^{s-stage} \frac{\xi - c_k}{c_i - c_k}$. The coefficients of continuous extension of above SDIRk method for 4th stage and 4th order, 5th stage and 5th order are given in Table 2 and 3.

The 4th order continuous extension is

$$b_1(\theta) = 13.8889\theta^4 - 23.1481\theta^3 + 10.5556\theta^2 - 0.8889\theta$$
$$b_2(\theta) = -6.3492\theta^4 + 11.8519\theta^3 - 7.1111\theta^2 + 1.6254\theta$$
$$b_3(\theta) = -8.9286\theta^4 + 12.5000\theta^3 - 3.7500\theta^2 + 0.2857\theta$$
$$b_4(\theta) = 1.388\theta^4 - 0.0120\theta^3 + 0.3056\theta^2 - 0.0222\theta$$

The 5th order continuous extension is

$$b_1(\theta) = 4.09\theta^5 - 12.28\theta^4 + 13.1395\theta^3 - 7.45\theta^2 + 0.19\theta$$
$$b_2(\theta) = -18.1818\,\theta^5 + 49.5455\theta^4 - 47.9121\theta^3 + 18.7682\theta^2 - 1.9909\theta$$
$$b_3(\theta) = 25.9740\theta^5 - 64.6104\theta^4 + 55.6320\theta^3 - 17.3701\theta^2 + 2.1111\theta$$
$$b_4(\theta) = -15.2672\,\theta^5 + 33.7786\theta^4 - 0.0383\theta^3 + 7.2252\theta^2 - 0.6947\theta$$
$$b_5(\theta) = 3.5112\theta^5 - 6.7544\theta^4 + 4.5327\theta^3 - 1.2526\theta^2 + 0.1175\theta$$

The classical RK4 are incorporated into delay term, the equation becomes

$$y_{n+1} = y_n + (b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4)$$

Where

$$k_1 = f(t_n + c_1h, y_n + ha_{11}k_1, y(t_n + c_1h - \tau))$$
$$k_2 = f(t_n + c_2h, y_n + ha_{21}k_1 + ha_{22}k_2, y(t_n + c_2\frac{h}{2} - \tau))$$
$$k_3 = f(t_n + c_3h, y_n + ha_{31}k_1 + ha_{32}k_2 + ha_{33}k_3, y(t_n + c_3\frac{h}{2} - \tau))$$
$$k_4 = f(t_n + c_4h, y_n + ha_{41}k_1 + ha_{42}k_2 + ha_{43}k_3 + ha_{44}k_4, y(t_n + c_4h - \tau))$$

In general

$$k_i = f(t_n + c_i h, y_n + h\sum_{j=1}^{i} a_{ij} k_j)$$

$$y_{n+1} = y_n + h\sum_{i=1}^{q} b_i k_1 \quad (i = 1,......q)$$

4.1. Procedure

Algorithm DDESDIRK

1. Find the initial value using history function
2. Divide the interval [t_0 ,t_f] into blocks [t_0- τ ,t_0], [t_0,t_0+ τ]…. [t_f - τ ,t_f]
3. The history function is used to find y(t) in [t_0- τ ,t_0]
4. At stage i, find k_i using SDIRK formula. In the interval [t_0, t_0- τ] estimates the delay term y(t_n+c_ih- τ) using cubic-spline interpolation or initial value.
5. For m≥ 2 estimate the value of delay term at interval [t_0+(m-1)*τ ; t_0+m*τ] using cubic-spline interpolation of previous block value.

5. Cubic Spline Interpolation Method

Given a set [(x_i, y_i), i =0,1….n] of n+1 point –value pairs, where x_0<x_1<.....<x_n. We wish to fit a piecewise-cubic spline f(x) to the point. That is, the curve f(x) is made up of n cubic polynomial $f_i(x) = a_i+b_ix+c_ix^2+d_ix^3$ for i = 0,1,….n-1, where if x falls in the range x_i≤ x ≤ x_{i+1}, then the value of the curve is given by $f(x) = f_i(x - x_i)$. The point x_i, at which the cubic polynomial are pasted together are called knots. For simplicity, we shall assume that x_i= i for i = 0,1…n. The continuity of f(x) ensures the condition that satisfy $f(x_i) = f_i(0) = y_0$ for i = 0,1,2….n-1, $f(x_{i+1})=f_i(1) = y_{i+1}$, for i = 0,1,2….n-1. The first derivative at each knot is continous $f'(x_{i+1}) = f_i'(1) = f_{i+1}'(0)$. Further, let k and n, k<n, are non- negative integers. Function S(x) is said to be a spline function of degree n with smoothness k if the following conditions are satisfied:

1. On each sub-interval [x_i, xi_{+1}] S(x) coincides with an algebraic polynomial of degree at most n.
2. S(x) and its derivatives up to order k all continous on the interval [a, b]. On the sub-interval [x_i, xi_{+1}] the cubic spine interpolation represents the polynomial as $S(x) = a(x-x_i)^3+b(x-x_i)^2+c(x-x_i)+d$, where a,b,c,d are the coefficients of the polynoimal.

6. Stability Regions

The P-stability properties of a numerical method for DDE's is the set S_P of pair of complex numbers $(\alpha,\beta), \alpha = h\lambda; \beta = h\mu,$ such that the discrete numerical solution {y_n}, n ≥ 0 of $y'(t) = \lambda y(t) + \mu\eta(t-\tau)$, t ≥$t_0$ y(t) = $\phi(t)$ t ≤ $t_0$where λ, μ ∈ c and τ is a constant delay obtained with constant step size h under the constraint $h = \frac{\tau}{m}, m \geq 1$, m integer satisfies $\lim_{n\to\infty} y_n = 0$ for all constant delay τ and all initial function $\phi(t)$.

Considering, the dealy is constant and applying R-Kmethod.

$$\eta(t_n + \theta h_{n+1}) = y_n + h_{n+1}\sum_{i+1}^{s} b_i(\theta) f(t_{n+1}^i, y_{n+1}^i, \bar{y}_{n+1}^i), 0 \leq \theta \leq 1 \quad (8)$$

$$y_{n+1}^i = y_n + h_{n+1}\sum_{j=1}^{s} a_{ij} f(t_{n+1}^i, y_{n+1}^i, \bar{y}_{n+1}^i), \quad i = 1, 2, ...s \quad (9)$$

If $t_{n+1}^{i} - \tau(t_{n+1}^{i}, y_{n+1}^{i}) > t_n$

$$\bar{y}_{n+1}^{i} = \eta(t_{n+1}^{i} - \tau(t_{n+1}^{i}, y_{n+1}^{i})) \qquad (10)$$

takes the simpler from

$$\eta(t_n + \theta h_{n+1}) = y_n + h_{n+1}\sum_{i+1}^{s} b_i(\theta) f(t_{n+1}^{i}, y_{n+1}^{i}, \eta(t_{n+1}^{i} - \tau), \quad 0 \le \theta \le 1 \qquad (11)$$

$$y_{n+1}^{i} = y_n + h_{n+1}\sum_{j=1}^{s} a_{ij} f(t_{n+1}^{j}, y_{n+1}^{i}, \eta(t_{n+1}^{j} - \tau)), \quad i = 1, 2, \ldots\ldots s \qquad (12)$$

where for $h_{n+1} < \tau$, $\eta(t_{n+1}^{j} - \tau)$ is known for any j applied to the test equation

$$y^{/}(t) = \lambda y(t) + \mu y(t-\tau), \; t \ge t_0$$
$$y(t) = \varphi(t) \qquad t \le t_0 \qquad (13)$$

with constant step size h satisfying the constraint $h = \frac{\tau}{m}$, takes the form

$$y_{n+1}^{i} = y_n + h\sum_{j=1}^{s} a_{ij}(\lambda y_{n+1}^{j} + \mu\eta(t_{n-m+1}^{j})), \quad i = 1, 2, \ldots\ldots s \qquad (14)$$

$$\eta(t_n + \theta h) = y_n + h\sum_{j=1}^{s} b_i(\theta)(\lambda y_{n+1}^{i} + \mu\eta(t_{n-m+1}^{i})) \qquad (15)$$

with

$b = (b_1, b_2 \ldots\ldots b_s)^T$, $b(\theta) = (b_1(\theta), b_2(\theta) \ldots\ldots b_s(\theta))^T$, $e = (1,1,\ldots.. 1)^T$

The unit vector I the s dimensional identity matrix, $A = (a_{ij})_i^s$, $j = 1$ $\alpha = h\lambda$ and $\beta = h\mu$ after elimination of the stage value y_{n+1}^{i} from equation(14) and computation of equation(15) for $\theta = c_1, c_2 \ldots.. c_s, 1,$ we get

$$\eta(t_{n+1}^{i}) = R_i(\alpha) y_n + \beta\sum_{j=1}^{s}(b(c_i)^T (I - \alpha A)^{-1})_j \eta(t_{n-m+1}^{j}), i = 1 \ldots\ldots\ldots s \qquad (16)$$

where

$$R_i(\alpha) = R^{(c_i)}(\alpha) = 1 + ab(c_i)^T (I - \alpha A)^{-1} e \qquad (17)$$

and $(x)_j$ is to be understood as the j[th] component of the row vector x. Moreover by equation (15) for $\theta = 1$, we get

$$y_{n+1} = R(\alpha)y_n + \beta\sum_{j=1}^{s}(b^T(I-\alpha A)^{-1})_j \eta(t_{n-m+1}^j) \tag{18}$$

where $R(\alpha)$ is the A-stability function.

$$R(\alpha) = \frac{\det[I-\alpha A+\alpha eb^T]}{\det[I-\alpha A]} \tag{19}$$

Therefore, A-stability region

$$s_A = \{\alpha \in C / |R(\alpha) < c|\} \tag{20}$$

This pair of equations reduces to the recurrence relation with constant coefficients

$$H_{n+1} = p(\alpha)H_n + \beta Q(\alpha)H_{n-m+1} \tag{21}$$

For the sequence of (s+1)-dimensional vectors

$$H_n = [\eta(t_n')......\eta(t_n^s), y_n]^T$$

Where

$$P(\alpha) = \begin{bmatrix} 0 & e+\alpha B(I-\alpha A)^{-1}e \\ 0^T & 1+\alpha b^T(I-\alpha A)^{-1}e \end{bmatrix}$$

$$Q(\alpha) = \begin{bmatrix} B(I-\alpha A)^{-1} & 0 \\ b^T(I-\alpha A)^{-1} & 0 \end{bmatrix}$$

and $B\left[b_j(c_i)\right]_{i,j}^{s} = 1$.The asymptotic behaviour of the solution of equation (21) is determined by the roots ξ of its characteristic equation

$$\det[\xi^m I - \xi^{m-1}p(\alpha) - \beta Q(\alpha)] = 0 \tag{22}$$

where this time, I is the (s+1) dimensional identity matrix. By a small direct calculation it is easy to check that for all $\xi \neq 0$ such that $\det[I-\alpha A-(\beta/\xi^m)B] \neq 0$ the left hand side of equation (22) can be factorized as follows

$$\det[\xi^m I - \xi^{m-1}p(\alpha) - \beta Q(\alpha)] = \xi^{ms+m-1}\det[I-\alpha A-(\beta/\xi^m)B](\xi - R^*(\alpha, \beta/\xi^m)) \tag{23}$$

Therefore, instead of the roots ξ of the characteristic equation (22), we can equivalently consider the solutions of the algebraic equation $\xi = R^*(\alpha, \beta/\xi^m)$, where the rational function

$R^*(\alpha, z) = 1 + (\alpha + z)b^T(I - \alpha A - zB)^{-1}e$ is called the P-stability function or $R^*(\alpha, z) = \dfrac{\det[I - \alpha A - zB + (\alpha + z)eb^T]}{\det[I - \alpha A - zB]}$.

P-stability region is defined as

$$\Gamma_\alpha = \left\{ z \in c \,/ \left| R^*(\alpha, z) \right| = 1 \right\}$$

The A-stability polynomial for 4th order coefficients is

$$R(\alpha) = \frac{-0.00206667\alpha^4 - 0.02533333\alpha^3 - 0.06000000\alpha^2 + 0.20000000\alpha + 1.00000000}{0.00160000\alpha^4 - 0.032000000\alpha^3 + 0.24000000\alpha^2 - 0.80000000\alpha + 1.00000000}$$

The A-stability polynomial for 5th order coefficients is

$$R(\alpha) = \frac{0.00018849955\alpha^5 + 0.0026321801\alpha^4 + 0.026511855\alpha^3 + 0.17373083\alpha^2 + 0.6493714\alpha + 1.00000000}{-0.000001695847\alpha^5 + 0.00012091477\alpha^4 - 0.0034485141\alpha^3 + 0.049176167\alpha^2 - 0.3506286\alpha + 1.00000000}$$

The P-stability polynomial for 4th order coefficient is $R^*(\alpha, \mathrm{z})$ where

$$\begin{aligned} Nr(\alpha, \mathrm{z}) = {} & -0.002066670\alpha^4 - 0.008266666\alpha^3 z - 0.025333330\alpha^3 - 0.01240000\alpha^2 z^2 \\ & - 0.075999999\alpha^2 z - 0.060000002\alpha^2 - 0.008266666\alpha z^3 - 0.075999999\alpha z^2 \\ & - 0.12\alpha z + 0.2\alpha - 0.00206667z^4 - 0.02533333z^3 - 0.060000002z^2 + 0.2 + 1.0 \end{aligned}$$

$$\begin{aligned} Dr(\alpha, \mathrm{z}) = {} & 0.0016\alpha^4 + 0.0064\alpha^3 z - 0.03200000\alpha^3 + 0.0096\alpha^2 z^2 + 0.48\alpha z - 0.8\alpha + 0.0016z^4 - 0.032z^3 \\ & + 0.24z^4 - 0.8z + 1.0 \end{aligned}$$

For fifth order coefficient

$$\begin{aligned} Nr(\alpha, \mathrm{z}) = {} & 0.00188499\alpha^5 + 0.000942497\alpha^4 z + 0.00263218\alpha^4 + 0.00188499\alpha^3 z^2 + 0.0105287\alpha^3 z \\ & + 0.02651185\alpha^3 + 0.001884995\alpha^2 z^3 + 0.0157931\alpha^2 z^2 + 0.07953556\alpha^2 z + 0.17373083\alpha^2 \\ & + 0.0009424973\alpha z^4 + 0.01052872\alpha z^3 + 0.079535564\alpha z^2 + 0.3474616\alpha z + 0.6493714\alpha \\ & + 0.000188499955z^5 + 0.0026321801z^4 + 0.026511855z^3 + 0.1737308z^2 + 0.6493714z + 1.0. \end{aligned}$$

$$\begin{aligned} Dr(\alpha, \mathrm{z}) = {} & -0.00000169\alpha^5 - 0.000008479\alpha^4 z + 0.00012091\alpha^4 - 0.000016958\alpha^3 z^2 + 0.00048365\alpha^3 z \\ & - 0.00344851\alpha^3 - 0.000016958\alpha^2 z^3 + 0.000725488\alpha^2 z^2 - 0.0103455\alpha^2 z + 0.04917616\alpha^2 \\ & - 0.0000084792\alpha z^4 + 0.000483659\alpha z^3 - 0.01034554\alpha z^2 + 0.098352333\alpha z - 0.3506286\alpha \\ & - 0.000001695847z^5 + 0.00012091477z^4 - 0.003448514z^3 + 0.049176167z^2 - 0.3506286z + 1.0. \end{aligned}$$

7. Numerical Results

The delay Food-Limited Equation is

$$\frac{dx}{dt} = rx(t)\left[\frac{k - x(t-\tau)}{k + rcx(t-\tau)}\right] \qquad (24)$$

Where r = 0.15, k=1.00, c = 0.5 and τ = 8 for $t \in [0,100]$ and the initial function $x(t) = \frac{1}{2}e^{-0.15t}$ for $t < 0$.

We have experimented program in MATLAB to solve equation (24) using the above mentioned procedure. Table 4 shows the result of delay differential equation and Figure 3 shows the error graph. The absoulute error (er1) is calculated for solution DDE using SDIRK method of order 4 incorporated with Cubic-Spline interpolation and error (er2) is calculated for solving DDE using SDIRK method of order 5 incorporated with Cubic-Spline interpolation. The A-stability region of SDIRK4 and SDIRK5 is given in Figure 1 and P-stability is in Figure2.

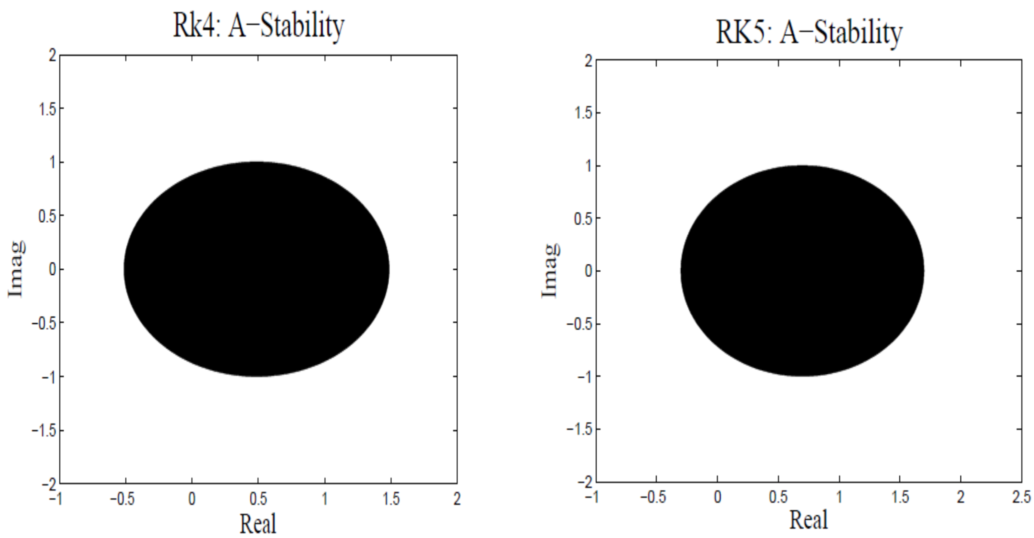

Fig. 1: A-Stability Region

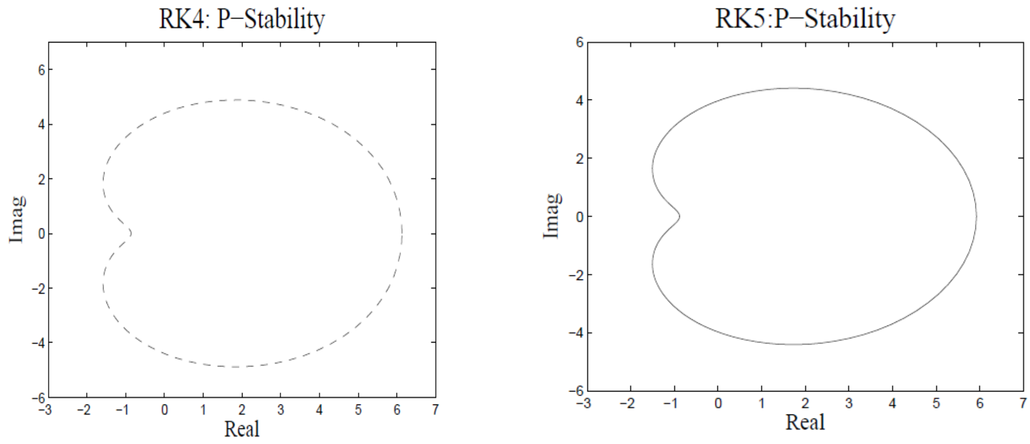

Fig. 2: P-Stability Region

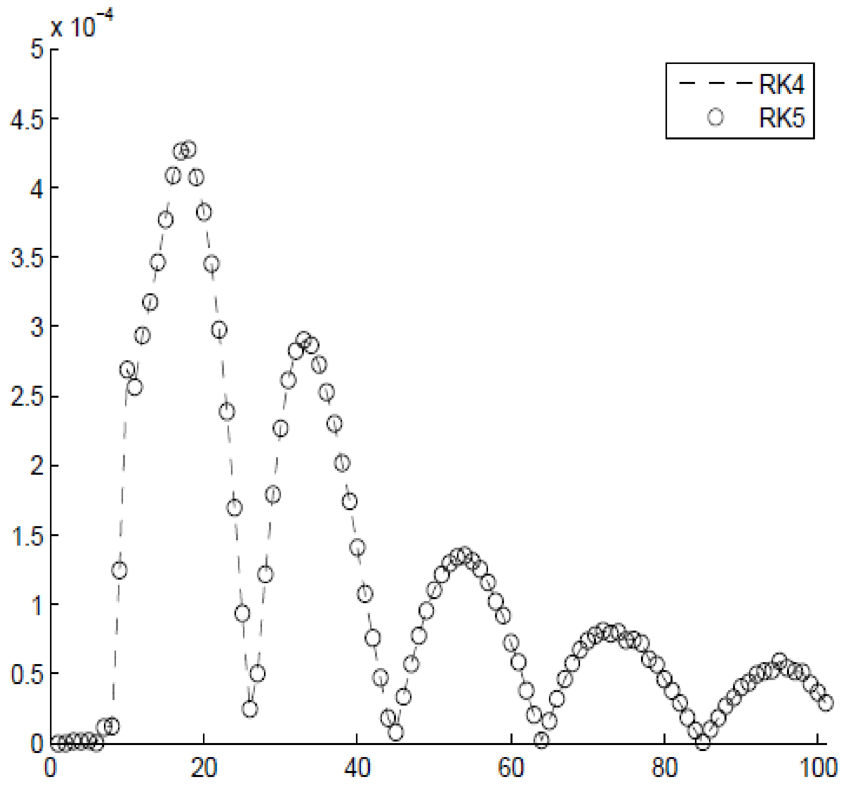

Fig.3: Error Graph

Table 4: Results of SDIRK4 AND SDIRK5 method for food-limited model equation

Time	DDE Y	SDIRK4 CSPLINE Y1	SDIRK5 CSPLINE Y2	Er1 (y-y1)	Er2 (y-y2)
0	0.5000000000000	0.5000000000000	0.5000000000000	0.0000000000000	0.0000000000000
1	0.4649039697365	0.4649040199425	0.4649039785772	0.0000000050206	0.0007850225698
2	0.4446426620529	0.4649040199425	0.4446442708397	0.0000016800755	0.0010393440971
3	0.4360135451664	0.4360149802399	0.4360148542850	0.0000014350735	0.0010808490531
4	0.4370618276929	0.4370639176004	0.4370637553019	0.0000020899074	0.0010823232142
5	0.4466636373766	0.4466640942929	0.4466638999807	0.0000004569163	0.0011567384878
6.	0.4642727108787	0.4642844729356	0.4642842446034	0.0000117620569	0.0013709267335
7	0.4897693634441	0.4897572394227	0.4897569979562	0.0000121240213	0.0018427039416
8	0.5233949241086	0.5235200200865	0.5235197029711	0.0001250959779	0.0024670374550
9	0.5642287458754	0.5644983091072	0.5644980211595	0.0002695632317	0.0033878531515
10	0.6107609893802	0.6110178415918	0.6110175375141	0.0002568522115	0.0046861413049
11	0.6625509347070	0.6628451625720	0.6628448627982	0.0002942278649	0.0061971055796
12	0.7191340208954	0.7194520830431	0.7194517929068	0.0003180621476	0.0079716296549
13	0.7799150970552	0.7802619219382	0.7802616517565	0.0003468248830	0.0099639742150
14	0.8440922332447	0.8444694612472	0.8444692224984	0.0003772280024	0.0121192191506
15	0.9105958965548	0.9110050988280	0.9110049037400	0.0004092022731	0.0143480260401
16	0.9779993058854	0.9784256622186	0.9784255269757	0.0004263563332	0.0165366319091
17	1.0445394403805	1.0449673788702	1.0449673091500	0.0004279384896	0.0185225506830
18	1.1083925048003	1.1088000577693	1.1088000550230	0.0004075529690	0.0201535457420
19	1.1676499692607	1.1680324862437	1.1680325561762	0.0003825169830	0.0212629263140
20	1.2203526142056	1.2206978072476	1.2206979495567	0.0003451930420	0.0217181405309
21	1.2646057295285	1.2649034541335	1.2649036654593	0.0002977246049	0.0214106515249
22	1.2987198751655	1.2989586157538	1.2989588880198	0.0002387405883	0.0202853743607
23	1.3213644805141	1.3215338089672	1.3215341293784	0.0001693284531	0.0183509447589
24	1.3317196182462	1.3318130007222	1.3318133515118	0.0000933824759	0.0156881688952
25	1.3295957370526	1.3296200820048	1.3296204428296	0.0000243449521	0.0124425190291
26	1.3155073936313	1.3154567274151	1.3154570781239	0.0000506662162	0.0088427592807
27	1.2905758033166	1.2904537182060	1.2904540394202	0.0001220851105	0.0051232206272
28	1.2564569476130	1.2562773692126	1.2562776439038	0.0001795784003	0.0015156158782
29	1.2152153321411	1.2149883043898	1.2149885198909	0.0002270277513	0.0017639604023
30	1.1691266685136	1.1688650528600	1.1688652019904	0.0002616156536	0.0045579616082
31	1.1205048932539	1.1202222744201	1.1202223559254	0.0002826188338	0.0067702235331
32	1.0715356280740	1.0712452009067	1.0712452189128	0.0002904271673	0.0083676026290

Time	DDE Y	SDIRK4 CSPLINE Y1	SDIRK5 CSPLINE Y2	Er1 (y-y1)	Er2 (y-y2)
33	1.0241458117198	1.0238591565482	1.0238591191102	0.0002866551715	0.0093711101468
34	0.9799189343311	0.9796462870730	0.9796462041440	0.0002726472581	0.0098424878735
35	0.9400665889073	0.9398137990999	0.9398136807378	0.0002527898074	0.0098635898464
36	0.9054345900879	0.9052043793886	0.9052042346327	0.0002302106992	0.0095248408831
37	0.8765358016251	0.8763340667720	0.8763339031314	0.0002017348531	0.0089183679856
38	0.8536189228692	0.8534448181854	0.8534446416357	0.0001741046837	0.0081144409657
39	0.8367017960291	0.8365609234467	0.8365607387399	0.0001408725824	0.0071813855439
40	0.8256488848052	0.8255414683386	0.8255412793995	0.0001074164666	0.0061605242745
41	0.8202003411954	0.8201246014005	0.8201244116703	0.0000757397948	0.0050833264138
42	0.8200091534354	0.8199619790905	0.8199617917697	0.0000471743448	0.0039716507861
43	0.8246614546491	0.8246433616431	0.8246431798436	0.0000180930060	0.0028452658648
44	0.8337041062652	0.8337123846758	0.8337122114948	0.0000082784105	0.0017131183078
45	0.8466417405900	0.8466754137219	0.8466752522542	0.0000336731318	0.0005878067686
46	0.8629484450255	0.8630056873914	0.8630055406983	0.0000572423658	0.0005198557333
47	0.8820671481422	0.8821448491451	0.8821447201863	0.0000777010028	0.0015972085829
48	0.9034079653767	0.9035037728867	0.9035036644202	0.0000958075099	0.0026264133597
49	0.9263538330229	0.9264643129756	0.9264642274339	0.0001104799527	0.0035881280045
50	0.9502615079328	0.9503832787218	0.9503832180761	0.0001217707890	0.0044589494193
51	0.9744698263478	0.9745995925430	0.9745995581661	0.0001297661952	0.0052135244667
52	0.9983110960466	0.9984452387305	0.9984452312693	0.0001341426839	0.0058265324628
53	1.0211250310776	1.0212601919924	1.0212602112612	0.0001351609147	0.0062736507336
54	1.0422798541148	1.0424110121734	1.0424110570824	0.0001311580585	0.0065362763296
55	1.0611867721709	1.0613122546018	1.0613123231298	0.0001254824309	0.0065968380298
56	1.0773334065087	1.0774493292231	1.0774494184450	0.0001159227143	0.0064513822024
57	1.0902989556597	1.0904010117917	1.0904011179671	0.0001020561320	0.0061037309650
58	1.0997675731219	1.0998595486126	1.0998596673347	0.0000919754906	0.0055595039010
59	1.1055739295980	1.1056462721874	1.1056463985872	0.0000723425893	0.0048558818223
60	1.1076624671863	1.1077208957415	1.1077210247339	0.0000584285551	0.0040103932364
61	1.1061450946114	1.1061831763671	1.1061833029219	0.0000380817556	0.0030748629051
62	1.1012459995550	1.1012663802243	1.1012664996373	0.0000203806693	0.0020839827844
63	1.0933207937628	1.0933228500895	1.0933229582292	0.0000020563266	0.0010857466427
64	1.0828190950335	1.0828028310391	1.0828029245454	0.0000162639943	0.0001220771379
65	1.0702605524614	1.0702284145365	1.0702284909516	0.0000321379248	0.0007718400440
66	1.0562112452452	1.0561649010026	1.0561649588240	0.0000463442426	0.0015645918965
67	1.0412495700255	1.0411919942084	1.0411920328647	0.0000575758170	0.0022350705303
68	1.0259446168269	1.0258770309556	1.0258770507163	0.0000675858713	0.0027676734984
69	1.0108259447881	1.0107519808955	1.0107519827340	0.0000739638925	0.0031590909202

Time	DDE Y	SDIRK4 CSPLINE Y1	SDIRK5 CSPLINE Y2	Er1 (y-y1)	Er2 (y-y2)
70	0.9963732295053	0.9962953290007	0.9962953144334	0.0000779005046	0.0034095076696
71	0.9830002383845	0.9829192949076	0.9829192658313	0.0000809434769	0.0035244514048
72	0.9710412492673	0.9709622534589	0.9709622120050	0.0000789958083	0.0035192675172
73	0.9607657789575	0.9606857716391	0.9606857200525	0.0000800073184	0.0033999647317
74	0.9523496029661	0.9522754015529	0.9522753421018	0.0000742014132	0.0031917274119
75	0.9459187816038	0.9458442636305	0.9458441985432	0.0000745179733	0.0028974886366
76	0.9415106404587	0.9414384891727	0.9414384205968	0.0000721512859	0.0025417040376
77	0.9391044099653	0.9390437235970	0.9390436535730	0.0000606863683	0.0021457993299
78	0.9386487553608	0.9385920762986	0.9385920067436	0.0000566790621	0.0017073954273
79	0.9400152454136	0.9399691011266	0.9399690338200	0.0000461442870	0.0012537833685
80	0.9430585363893	0.9430205744985	0.9430205110693	0.0000379618908	0.0007885748105
81	0.9475880178866	0.9475589884163	0.94755893032745	0.0000290294702	0.0003266401634
82	0.9533882979273	0.9533697845878	0.95336973311824	0.0000185133395	0.0001200645809
83	0.9602267851602	0.9602174218665	0.96021737809147	0.0000093632937	0.0005439827895
84	0.9678505083366	0.9678513950830	0.96785135985318	0.0000008867464	0.0009327972220
85	0.9760020827512	0.9760123144401	0.97601228836127	0.0000102316889	0.0012794336349
86	0.9844196913289	0.9844381178124	0.98443810122842	0.0000184264835	0.0015760188697
87	0.9928433282697	0.9928704312072	0.99287042418779	0.0000271029375	0.0018140106468
88	1.0010280970046	1.0010610235613	1.00106102589711	0.0000329265566	0.0019913200839
89	1.0087377569485	1.0087782296757	1.00877824088235	0.0000404727271	0.0020997164522
90	1.0157693090765	1.0158131483066	1.01581316763857	0.0000438392301	0.0021434625475
91	1.0219358399761	1.0219853698376	1.02198539631320	0.0000495298614	0.0021166115209
92	1.0270962241581	1.0271479571630	1.02714798959974	0.0000517330048	0.0020278112004
93	1.0311388663662	1.0311914001172	1.03119143717684	0.0000525337510	0.0018800553155
94	1.0339870503582	1.0340462909023	1.03404633114290	0.0000592405441	0.0016731899664
95	1.0356302658807	1.0356845248665	1.03568456679998	0.0000542589857	0.0014337453162
96	1.0360671828874	1.0361189132332	1.03611895538396	0.0000517303457	0.0011578911582
97	1.0353500963889	1.0354011939615	1.03540123492391	0.0000510975726	0.0008573818433
98	1.0335757533448	1.0336185332125	1.03361857170234	0.0000427798676	0.0005533466139
99	1.0219358399761	1.0308887110447	1.0308887459438	0.0000367484607	0.0002471741707
100	1.0270962241581	1.0273542696857	1.0273543000725	0.0000291146848	0.0000465788135

8. Conclusion

In this paper, we have described by solving Delay Differential Equation using SDIRK methods of order 4 and 5 by incorporating with Cubic-Spline interpolation method and obtained the result by implementing the study with MATLAB function DDE23. Finally it is found that SDIRK method of order 4 incorporated with Cubic-Spline interpolation for the assigned food limited population model equation and discussed about the stability regions and polynomial with the exact solution.

References

1. N.Bacaer, "A Short History of Mathematical Population Dynamics,"*Springer*, London. 2011.
2. Christopher T.H.Baker and Christopher A.H.Paul, "Computing stability regions rung- kutta method for delay differential equations,"*IMA Journal of Numerical Analysis*,14,3,347-362,1994.

3. Christopher T.H.Baker and Christopher A.H.Paul, and David R .Wille, "A bibliography on the numerical solutionof delay differential equations, " 1995 .
4. A.Bellen and M.Zennaro, "Numerical Methods for Delay Differential Equation," ISBN 978019850546, *ClarendonPress 2003*.
5. Alfredo Bellen, Zdislaw Jackiewicz, and Marino Zennaro, "Stability analysis of one-step methods for neutral delay-differential equations, "*Numerische Mathematik*,52, 6, 605-619,1988.
6. Mohamed bin Suleiman and Fudziah Ismail, "Solving delay differential equations using componentwise partitioningby runge-kutta method, "*Applied Mathematics and Computation*,122, 3, 301-323,2001.
7. J.C.Butcher,"Numerical methods for ordinary differential equations in the 20th century,"*Journal of Computationaland Applied Mathematics*, 125-129,2000.
8. J.C.Butcher,"Numerical methods for ordinary differential equations,"*Wiley and Sons*, England, 2008.
9. Ismail Fudziah Raed Ali, Al-Khasawneh, Mohamed Suleiman, " Numerical treatment of delay differential equations by rung-kutta method using hermite interpolation, *Matematika*,18, 79-90, 2002.
10. Richard Goodman and Alan Feldstein, "Round-off reeor for retarded ordinary differential equations a priori bounds and estimates,"*Numerische Mathematik*, 21, 5, 3555-372,1973.
11. K.Gopalsamy, " stability and oscillations in delay differential equations of population dynamics," Mathematics and its applications, *Kluwer Academic Publishers*, Dordrecht, Boston, 1992.
12. K. Gopalsamy, MRS Kulenovic, and G.Ladas, " Time lags in a foood-limited population, *Applicable Analysis*,"31, 3, 225-237, 1988.
13. G.Evelyn Hutcvhinson,"Circular causal systems in ecology,"*Annals of New York Academy of Sciences*, 50, 4, 221-246,1948.
14. K.J.In'tHout," A new interpolation procedure for adapting runge-kutta methods to delay differential equations,"*BIT Numerical Mathematics*, 32, 4, 634-649, 1992.
15. Fudziah Ismail and R.Alkaswneh, " Embedded diagonally implicit rung-kutta-nystrom4(3) pair for solving special second-order ivps,"*Applied Mathematics and Computation*, 190, 2, 1803-1814, 2007.
16. Yang Kuang, " Delay differential equations with applications in population dynamics,"*Academic Press, New York*1993.
17. H.J.Oberle and H.J.Pesch,"Numerical treatment of delay differential equarions by hermite interpolation,*"Numerische Mathematik*, 37, 2, 235-255, 1981.
18. L.F.Shampine and s.Thompson, "Solving ddes in matlab,*Applied Numerical Mathematics*," 37, 4, 441-458, 2001.
19. Frederick E.Smith, "population dynamics in daphnia magna and a new model for population growth,*Ecology*, 44, 4, 651-663, 1963.
20. Eric W.Weisstein, "Logistic Equation, From MathWorld-*A Wolfram Web Resource*," 2003.
21. Marino Zennaro, "P-stability properties of runge-kutta method for delay differential equarions,"*NumerischeMathematik*, 49, 2, 305-318, 1986.
22. N.I.C.Jawias Fudziah Ismail, Mohamed Suleiman and Azmi bib Jaafar, "Fourth order four-stage diagonally implicit Rung-Kutta method for linear ordinary differential equation,"*Malaysian Journal of Mathematical Sciences* 4, 1, 95-105, 2010.
23. Fudziah Ismail, Nur Izzati Che Jawias, Mohamed Suleiman and Azmi Jaafar, "Solving Linear Ordinary Differential Equations Using Singly Diagonally Implicit Runge-Kutta Fifth Order Five-Stage Method, "*WSEAS Transacationson Mathematics*, 8,8, 393-402,2009.