Numerical Solution of N^{th} Order Linear Delay Differential Equation Using Runge-Kutta Method

Suha Najeeb Al-Rawi * **Raghad Kadhim Salih****

Amaal Ali Mohammed***

Date of acceptance 9/11/2005

Abstract

In this paper, fourth order Runge Kutta method has been used to find the numerical solution for different types of first order linear delay differential equation. Moreover, this method has been modified in order to treat nth order linear delay differential equations.

Keywords: Delay differential equation (DDE), Retarded and Neutral delay differential equation, Runge-Kutta method, Numerical solution.

1. Introduction

The ordinary delay differential equation "DDE" is an equation in an unknown function y(t) and some of its derivatives are evaluated at arguments that differ in any of fixed number of values $\tau_1, \tau_2, \ldots, \tau_n$.

The general form of the nth order DDE is given by

$$F(t, y(t), y(t-\tau_1), \ldots, y(t-\tau_k), y'(t), y'(t-\tau_1), \ldots, y'(t-\tau_k), y^{n}(t), \ldots, y^{n}(t-\tau_k)) = 0 \quad \ldots (1)$$

where F is a given function and $\tau_1, \tau_2, \ldots, \tau_k$ are given fixed positive numbers called the "time delay" [1].

In some literature, equ. (1) is called a differential equation with deviating argument [1,2] or an equation with time lag [3] or a differential difference equation or a functional differential equation [4].

The general linear first order delay differential equation with constant coefficients can be presented in the form:

$$a_0 y'(t) + a_1 y'(t-\tau) + b_0 y(t) + b_1 y(t-\tau) = g(t) \quad \ldots (2)$$

where g(t) is a given continuous function and τ is a positive constant and a_0, a_1, b_0, b_1 are constants.

A delay differential equation of the first order is called homogenous if $g(t) \equiv 0$, otherwise it is not homogenous and the function g(t) is called the right-hand side vector of the equation.

We can distinguish between three types of delay differential equations, which are:

1. Retarded delay differential equation, which is obtained when: $(b_1 \neq 0, a_1 = 0)$ in equ.(2) the delay comes in y only and (2) takes the form : $a_0 y'(t) + b_0 y(t) + b_1 y(t-\tau) = g(t)$.

*Dr,University of Technology
**University of Technology
***University of Al-Mustansiriya

2. Neutral delay differential equation, which is obtained when:
   $(a_1 \neq 0, b_1 = 0)$ in equ.(2) the delay comes in y' only and (2) takes the form:
   $a_0 y'(t) + a_1 y'(t-\tau) + b_0 y(t) = g(t)$.
3. Mixed delay differential equation, (sometimes called advanced type), which is obtained where $(a_1 \neq 0, b_1 \neq 0)$, which means a combination of the previous two types.

The main difference between delay and ordinary differential equations is the kind of initial condition that should be used in delay differential equation differs from ordinary differential equation so that one should specify in delay differential equations an initial function on some interval of length τ, say $[t_0 - \tau, t_0]$ and then try to find the solution of equation (1) for all $t \geq t_0$.

Thus, we set $y(t) = \phi(t)$ for all $t_0 - \tau \leq t \leq t_0$, where $\phi(t)$ is some given continuous function. Therefore, the solution of delay differential equation consists of finding a continuous extension of $\phi(t)$ into a function y(t) which satisfies equ.(1) for all $t \geq t_0$, on successive time step interval with length equal to τ [5].

Another approach for solving DDE's which is developed by other researchers, are by using AL-Saady,A.S. [6] and AL-Marie, N.K. [5].

2. Runge-Kutta Method:

Runge Kutta methods, because of their self-starting property, have a unique place amongst the classical types of methods. They use only the information from the last step computed; therefore, they are called single-step method [7].

Runge Kutta method its utility is in finding numerical solution of initial value problems and this method is undertaken to find numerical solution both linear and non-linear for integral and integro-differential equation [8].

Consider the following first order differential equation

$y' = f(t, y(t))$ with $y(t_0) = y_0$ .... (3)

A q-order Runge Kutta method for equ.(3) is:

$$y(t_{i+1}) = y(t_i) + \sum_{j=1}^{q} w_j k_j \qquad 1 \leq i \leq N$$

where

$$k_j = k_j(t_i) = hf(t_i + c_j h, y(t_i) + \sum_{l=1}^{q} a_{jl} k_l) \qquad 1 \leq j \leq q$$

where w_j, c_j and a_{jl} are constants.

Runge-Kutta method of second and third order have been little used for ordinary differential equations, in general, and delay differential equations in particular.

However the following fourth order Runge-Kutta method which is most popular and more efficient for dealing with differential equations.

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad .. (4)$$

where

$$\begin{aligned} k_1 &= hf(t_n, y(t_n)) \\ k_2 &= hf(t_n + h/2, y(t_n) + k_1/2) \\ k_3 &= hf(t_n + h/2, y(t_n) + k_2/2) \\ k_4 &= hf(t_n + h, y(t_n) + k_3) \end{aligned} \quad .... (5)$$

2.1 A Single First Order Delay Differential Equation:

In this subsection a single step Runge Kutta method including fourth order is candidate to find the numerical solution for a different type of delay differential equation [7].

Let us reconsider the first order delay differential equation

$$y'(t) = f(t, y(t), y(t-\tau), y'(t-\tau)), \; t \in [t_0, \infty) \quad \ldots (6)$$

with initial function

$$y(t) = \phi(t) \quad \text{for} \quad t_0 - \tau \le t \le t_0 \quad \ldots (7)$$

equ.(6) may be solved if we use the initial function [9].

$$y'(t) = f(t, y(t), \phi(t-\tau), \phi'(t-\tau)) \quad \ldots (8)$$

with initial condition

$$y(t_0) = \phi(t_0) \quad \ldots (9)$$

For applying the Runge-Kutta method for equation (8) by using equ.(4) and equ.(5) we get the following formula.

$$y_{j+1} = y_j + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(t_j, \phi(t_j), \phi(t_j - \tau), \phi(t_j - \tau))$$
$$k_2 = hf(t_j + h/2, \phi(t_j) + k_1/2, \phi(t_j - \tau) + k_1/2, \phi(t_j - \tau) + k_1/2)$$
$$k_3 = hf(t_j + h/2, \phi(t_j) + k_2/2, \phi(t_j - \tau) + k_2/2, \phi(t_j - \tau) + k_2/2)$$
$$k_4 = hf(t_j + h, \phi(t_j) + k_3, \phi(t_j - \tau) + k_3, \phi(t_j - \tau) + k_3)$$

for $j = 0, 1, \ldots, n$

The following algorithm summarizes the steps for finding the numerical solution for the first order of delay differential equation.

Algorithm (RK4SI)

Step 1:

Set h = (τ) / N.

Step 2:

Compute $k_1 = hf(t_j, \phi(t_j), \phi(t_j - \tau), \phi'(t_j - \tau))$.

Step 3:

Compute $k_2 = hf(t_j + h/2, \phi(t_j) + k_1/2, \phi(t_j - \tau) + k_1/2, \phi'(t_j - \tau) + k_1/2)$

Step 4:

Compute $k_3 = hf(t_j + h/2, \phi(t_j) + k_2/2, \phi(t_j - \tau) + k_2/2, \phi'(t_j - \tau) + k_2/2)$

Step 5:

Compute $k_4 = hf(t_j + h, \phi(t_j) + k_3, \phi(t_j - \tau) + k_3, \phi'(t_j - \tau) + k_3)$

Step 6:

Compute $y_{j+1} = y_j + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

Step 7:

Set $t_j = jh$ for each $j = 0, 1, \ldots, n$.

2.2 A System of First Order Delay Differential Equations:

Consider a system of first order delay differential equation such as:

$$y'(t) = f_i(t, y_1(t), \ldots, y_n(t), y_1(t-\tau_1), \ldots, y_n(t-\tau_n), y_1'(t-\tau_1), \ldots, y_n'(t-\tau_n)),$$
$$t \in [-\tau, \infty) \quad \ldots (10)$$

with initial function

$$\begin{matrix} y_1(t) = \phi_1(t) & for & t_0 - \tau_1 \le t \le t_0 \\ \vdots & & \vdots \\ y_n(t) = \phi_n(t) & for & t_0 - \tau_n \le t \le t_0 \end{matrix} \quad \ldots (11)$$

Equation (10) may be solved by Runge-Kutta if

$$y'(t) = f(t, y_1(t), \ldots, y_n(t), \phi_1(t-\tau_1), \ldots, \phi_n(t-\tau_n), \phi_1'(t-\tau_1), \ldots, \phi_n'(t-\tau_n)) \quad \ldots (12)$$

with initial conditions

$$y_1(t_0) = \phi_1(t_0), \ldots, y_n(t_0) = \phi_n(t_0)$$

For applying Runge Kutta method for equations (10), (11) by using equations (4), (5) we get the following formula

$$y_{ij+1} = y_{ij} + \frac{1}{6}(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i})$$

where

$$k_{1i} = hf(t_j, \phi_1(t_j), \ldots \phi_m(t_j), \phi_1(t_j - \tau_1), \ldots \phi_m(t_j - \tau_m), \phi_1(t_j - \tau_1), \ldots \phi_m(t_j - \tau_m))$$

$$k_{2i} = hf(t_j + h/2, \phi_1(t_j) + k_{11}/2, \ldots \phi_m(t_j) + k_{1m}/2, \phi_1(t_j - \tau_1) + k_{11}/2, \ldots, \phi_m(t_j - \tau_m) + k_{m2} \phi_1(t_j - \tau) + k_{11}/2, \ldots \phi_m(t_j - \tau_m) + k_{1m}/2)$$

$$k_{3i} = hf(t_j + h/2, \phi_1(t_j) + k_{21}/2, \ldots \phi_m(t_j) + k_{2m}/2, \phi_1(t_j - \tau_1) + k_{21}/2, \ldots, \phi_m(t_j - \tau_m) + k_{2m}/2, \phi_1(t_j - \tau_1) + k_{21}/2, \ldots \phi_m(t_j - \tau_m) + k_{2m}/2)$$

$$k_{4i} = hf(t_j + h, \phi_1(t_j) + k_{3p} \ldots \phi_m(t_j); + k_{3m}, \phi_1(t_j - \tau_1) + k_{3p} \ldots \phi_m(t_j - \tau_m); + k_{3m}, \phi_1(t_j - \tau_1) + k_{3p} \ldots \phi_m(t_j - \tau_m) + k_{3m})$$

for each $j = 0,1,....,n, \quad i = 0,1,....,m$

The following algorithm summarizes the steps of finding the numerical solution for a system of linear delay differential equations.

Algorithm (RK4SY)

Step 1:

Set h = (τ) / N.

Step 2:

Compute $k_{1i} = hf_i(t_j, \phi_1(t_j), ... \phi_m(t_j), \phi_1(t_j - \tau_1), ..., \phi_m(t_j - \tau_m), \phi_1'(t_j - \tau_1), ... \phi_m'(t_j - \tau_m))$

Step 3:

Compute

$k_{2i} = hf_i(t_j + h/2, \phi_1(t_j) + k_{11}/2, ..., \phi_m(t_j) + k_{1m}/2, \phi_1(t_j - \tau_1) + k_{11}/2, ..., \phi_m(t_j - \tau_m) + k_{1m}/2, \phi_1'(t_j - \tau) + k_{11}/2, ... \phi_m'(t_j - \tau_m) + k_{1m}/2)$

Step 4:

Compute

$k_{3i} = hf_i(t_j + h/2, \phi(t_j) + k_{21}/2, ..., \phi_m(t_j) + k_{2m}/2, \phi_1(t_j - \tau_1) + k_{21}/2, ..., \phi_m(t_j - \tau_m) + k_{2m}/2, \phi_1'(t_j - \tau_1) + k_{21}/2, ..., \phi_m'(t_j - \tau_m) + k_{2m}/2)$

Step 5:

Compute

$k_{4i} = hf_i(t_j +, \phi_1(t_j) + k_{31}, ..., \phi_m(t_j), + k_{3m}, \phi_1(t_j - \tau_1) + k_{31}, ..., \phi_m(t_j - \tau_m), + k_{3m}, \phi_1'(t_j - \tau_1) + k_{31}, ..., \phi_m'(t_j - \tau_m) + k_{3m})$

Step 6:

Compute

$$y_{ij+1} = y_{ij} + \frac{1}{6}(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i})$$

Step 7: Set t_j = jh for each $j = 0,1,....,n$.

2.3 Nth - Order Delay Differential Equations:

The general form of nth–order delay differential equations

$$y^n(t) = f(t, y(t), y'(t), ..., y^{n-1}(t), y(t-\tau), y'(t-\tau), ... y^{n-1}(t-\tau)), \quad t \geq t_0$$

with initial function

$y(t) = \phi(t)$ for $t_0 - \tau \leq t \leq t_0$

where $\phi(t)$ and its first n-1 derivatives $\phi'(t), ..., \phi^{n-1}(t)$ are continuous on interval

$[t_0 - \tau, t_0]$

Obviously, the nth order equation with delay argument may be replaced for equations without a delay argument, by a system of nth-equation of first order delay differential equation as follows:

Let

$$x_1(t) = y(t)$$
$$x_2(t) = y'(t)$$
$$\vdots$$
$$x_{n-1}(t) = y^{(n-2)}(t)$$

therefore, we get the following system of the first order equation

$$x_1'(t) = x_2(t)$$
$$x_2'(t) = x_3(t)$$
$$\vdots$$
$$x_{n-1}'(t) = x_n(t)$$

$$x_n'(t) = f(t, x_1(t), ..., x_n(t), x(t-\tau), ..., x_n(t-\tau))$$

The above system can be treated using method prescribed in previous sections.

3. Numerical Examples:

Example (1):

Consider the following neutral delay differential equation of the first order:

$$y'(t) = 1 - y'(t - \frac{y(t)^2}{4}), \qquad t \geq 0$$

with initial function

$$y(t) = 1 + t \qquad 0 \leq t \leq 1$$

has exact solution

$$y(t) = 1 + \frac{t}{2} + \frac{t^2}{4} \qquad 0 \leq t \leq 1$$

When the algorithm (RK4SI) is applied, table (1) presents the comparison between the exact and numerical solutions depending on least square error (L.S.E.).

Table (1) The numerical solution of example(1) using (RK4SI)algorithm.

t	Exact	RK4SI
0	1	1
0.1	1.0525	1.0525
0.2	1.1100	1.1100
0.3	1.1725	1.1725
0.4	1.2400	1.2400
0.5	1.3125	1.3125
0.6	1.3900	1.3900
0.7	1.4725	1.4726
0.8	1.5600	1.5601
0.9	1.6525	1.6526
1	1.7500	1.7501
L.S.E.		3.13e-6

Example (2):

Consider the retarded delay differential equation of the first order:

$$y'(t)+y(t)-y(t-\frac{\pi}{2})=\sin t, \qquad t\geq 0$$

with initial function

$$y(t)=\sin t \qquad \frac{-\pi}{2}\leq t\leq 0$$

which has the exact solution

$$y(t)=\sin t \qquad 0\leq t\leq \frac{\pi}{2}$$

The problem is solved by applying the algorithm (RK4SI) to obtain the results of approximating function. Table (2) shows the numerical result with the exact solution depending on least square error (L.S.E.).

Table(2) The solution of example(2) using (RK4SI) algorithm

t	Exact	RK4SI
0	0	0
0.1	0.0998	0.0998
0.2	0.1987	0.1986
0.3	0.2955	0.2954
0.4	0.3894	0.3893
0.5	0.4794	0.4792
0.6	0.5646	0.5644
0.7	0.6442	0.6440
0.8	0.7174	0.7171
0.9	0.7833	0.7831
1	0.8415	0.8413
L.S.E.		2.6e-5

Example (3):

Consider the system delay differential equations of the first order:

$$y_1'(t)=y_5(t-1)+y_3(t-1), \qquad t\geq 0$$

$$y_2'(t)=y_1(t-1)+y_2(t-\frac{1}{2}), \qquad t\geq 0$$

$$y_3'(t)=y_3(t-1)+y_1(t-\frac{1}{2}), \qquad t\geq 0$$

$$y_4'(t)=y_5(t-1)+y_4(t-1), \qquad t\geq 0$$

$$y_5'(t)=y_1(t-1), \qquad t\geq 0$$

with initial functions

$$y_1(t)=e^{(t+1)}, \qquad t\leq 0$$

$$y_2(t)=e^{(t+\frac{1}{2})}, \qquad t\leq 0$$

$$y_3(t)=\sin(t+1), \qquad t\leq 0$$

$$y_4(t)=e^{(t+1)}, \qquad t\leq 0$$

$$y_5(t)=e^{(t+1)}, \qquad t\leq 0$$

with analytical solution

$$y_1(t)=e^t-\cos t+e, \qquad 0\leq t\leq\frac{1}{2}$$

$$y_2(t)=2e^t+e^{\frac{1}{2}}-2, \qquad 0\leq t\leq\frac{1}{2}$$

$$y_3(t)=e^{(t+\frac{1}{2})}-\cos t+1-e^{\frac{1}{2}}+\sin(1), \qquad 0\leq t\leq\frac{1}{2}$$

$$y_4(t)=\frac{1}{2}e^{2t}-\frac{1}{2}+e, \qquad 0\leq t\leq\frac{1}{2}$$

$$y_5(t)=e^t+e-1, \qquad 0\leq t\leq\frac{1}{2}$$

The example is solved by applying the algorithm (RK4SY). Table (3) gives a summary of the numerical solution and the least square errors (L.E.S.) for h=0.05.

Table (3) The exact and numerical solution of example (3) using RK4SY algorithm

T	$Exact_1$	$RK4SY_1$	$Exact_2$	$RK4SY_2$	$Exact_3$	$RK4SY_3$	$Exact_4$	$RK4S_4$	$Exact_5$	$RK4SY_5$
0	2.7183	2.7183	1.6487	1.6487	0.0175	0.0175	2.7183	2.7183	2.7183	2.7183
0.05	2.7708	2.7717	1.7513	1.7526	0.1034	0.1034	2.7709	2.7709	2.7696	2.7703
0.1	2.8284	2.8304	1.8591	1.8619	0.1958	0.1962	2.8290	2.8291	2.8235	2.8254
0.15	2.8913	2.8945	1.9724	1.9768	0.2955	0.2964	2.8932	2.8935	2.8801	2.8839
0.2	2.9596	2.9642	2.0915	2.0976	0.4024	0.4040	2.9642	2.9649	2.9397	2.9457
0.25	3.0334	3.0396	2.2168	2.2248	0.5168	0.5193	3.0426	3.0438	3.0023	3.0112
0.3	3.1128	3.1208	2.3484	2.3585	0.6389	0.6426	3.1293	3.1312	3.0681	3.0803
0.35	3.1980	3.2081	2.4869	2.4991	0.7690	0.7740	3.2252	3.2279	3.1373	3.1534
0.4	3.2890	3.3014	2.6324	2.6469	0.9073	0.9139	3.3311	3.3350	3.2101	3.2305
0.45	3.3861	3.4011	2.7853	2.8024	1.0540	1.0624	3.4481	3.4536	3.2866	3.3118
0.5	3.4894	3.5071	2.9462	2.9658	1.2094	1.2199	3.5774	3.5850	3.3670	3.3975
L.S.E		9.2E-4	L.S.E	0.0013	L.S.E	2.7E-4	L.S.E	1.16E-4	L.S.E	0.0025

Example (4):

Consider the delay differential equation of the second order:

$$y''(t)+\frac{1}{2}y(t)-\frac{1}{2}y(t-3\pi)=0, \qquad t\geq t_0$$

with initial function

$$y(t)=\cos t \qquad -3\pi\leq t\leq 0$$

The above delay differential equation can be replaced by a system of two first order equations.

$$x_1'(t)=x_2(t), \qquad t\geq 0$$
$$x_2'(t)=-\frac{1}{2}x_1(t)+\frac{1}{2}x_1(t-3\pi), \quad t\geq 0$$

with initial conditions

$$x(0)=1 \ , \ x'(0)=0 \quad -3\pi\leq t\leq 0$$

which has the exact solution

$$x_1(t)=\cos t \qquad 0\leq t\leq 3\pi$$
$$x_2(t)=-\sin t \qquad 0\leq t\leq 3\pi$$

The example is solved by applying (RK4SY) algorithm. Table (4) gives a summary of the numerical solution and the least square errors (L.E.S.) for h=0.1.

Table (4) The results of example(4) using RK4SY algorithm.

t	$Exact_1$	$RK4SY_1$	$Exact_2$	$RK4SY_2$
0	1	1	0	0
0.1	0.9950	0.9951	-0.0998	-0.09979
0.2	0.9801	0.9802	-0.1987	-0.1986
0.3	0.9553	0.9555	-0.2955	-0.2954
0.4	0.9211	0.9212	-0.3894	-0.3893
0.5	0.8776	0.8778	-0.4794	-0.4792
0.6	0.8253	0.8255	-0.5646	-0.5644
0.7	0.7648	0.7650	-0.6442	-0.6440
0.8	0.6967	0.6969	-0.7174	-0.7171
0.9	0.6216	0.6218	-0.7833	-0.7831
1	0.5403	0.5405	-0.8415	-0.8413
L.S.E.		2.6E-5	L.S.E.	2.6E-5

4. Conclusion :

In this work, fourth order Runge-Kutta method has been proved effectiveness in solving n^{th} order linear delay differential equations. From solving some numerical examples the following points have been identified:

1- The good approximation depends on the size of h, if h is decreased then the number of division points increases and L.S.E. approaches to zero.
2- Runge-Kutta method solved linear DDE's of any order by reducing the equation to a system of first order equations.

5. References :

1. Driver, R.D., (1977) "Ordinary and Delay Differential Equations" Springer verlag, New York, Inc.
2. Elśgol'ts, L.E., (1966) "Introduction to the theory of Differential Equations with Deviating Arguments" Holden. Day, Inc.
3. Oguztorelio, M.N., (1966) "Time-Lage Control Systems" Academic press, New York.
4. Bellman, R. and Cooke, K.L.,(1963) "Differential, Difference Equations" Academic press, Inc, New York.
5. AL-Marie, N.K.,(2001) "Variational Formulation of Delay Differential Equations" M.Sc. Thesis, Ibn AL-Haitham, Baghdad University.
6. AL-Saady, A.S.,(2000) "Cubic Spline Technique in Solving Delay Differential Equations" M.Sc. Thesis, College of Science, Saddam University.
7. Klaus Smchitt, (1972) "Delay and Functional Differential Equations and their Applications" Academic Press, New York.
8. Piskunov ON., (1974) "Differential and Integral Calculus, English translation" Mir Publishers, Moscow.
9. Baker, C.H. (2000) "Retarded Differential Equations", http:/ www.ma.man.ac.uk/nare ports.

الحلول العددية للمعادلات التفاضلية التباطؤية الخطية من الرتب العليا باستخدام طريقة رنكه- كته

سهى نجيب الراوي*   رغد كاظم صالح**   أمال علي محمد***

* دكتوراه، الجامعة التكنولوجية   ** الجامعة التكنولوجية
*** الجامعة المستنصرية

الخلاصة:

في هذا البحث استخدمت طريقة رنكه- كته من الدرجة الرابعة لإيجاد الحلول العددية للمعادلات التفاضلية التباطؤية الخطية من الرتبة الأولى، بالإضافة إلى ذلك طورت هذا الطريقة لمعالجة المعادلات التفاضلية التباطؤية الخطية من الرتب العليا .