A Third-Order Direct Integrators of Runge-Kutta Type for Special Third-Order Ordinary and Delay Differential Equations

Mohammed S. Mechee, F. Ismail, Z. Siri and N. Senu

1Department of Mathematics, Faculty of Mathematics and Computer Science, Kuffa University, Najaf, Iraq
2Institute of Mathematical Sciences, University of Malaya, 50603, Kuala Lumpur, Malaysia
3Department of Mathematics and Institute for Mathematical Research, Universiti Putra Malaysia, 43400, UPM, Serdang, Selangor, Malaysia

Corresponding Author: Mohammed S. Mechee, Department of Mathematics, Faculty of Mathematics and Computer Science, Kuffa University, Najaf, Iraq

ABSTRACT
A two sets of test problems are tested upon, the first set consists of problems on ordinary differential equations and the second set consists of problems on delay differential equations. The problems are solved using the new RKD method and numerical comparisons are made when the same problems are reduced to a first order system of ODEs and first order system of DDEs and solved using the existing Runge-Kutta methods of order three and four. The RKD method is adapted to solve delay differential equations (DDEs). Stability polynomial of the method for linear special third order DDEs is given. The numerical results have clearly shown the advantage and the efficiency of the new method in terms of accuracy and computational time.

Key words: Special third order, delay differential equations, direct Runge-Kutta method

INTRODUCTION
Generally speaking, a special third order differential equations (ODEs) of the form:

\[ Y'(t) = f(t, y(t)), \quad t \geq t_0 \]  

(1)

with initial conditions:

\[ y(t_0) = \alpha, \quad y'(t_0) = \beta, \quad y''(t_0) = \gamma \]

where, \( f: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \) which is not explicitly dependent on the first derivative \( y'(x) \) and the second derivative \( y''(x) \) of the solution. The ordinary differential Eq. 1 is frequently found in many physical problems such as thin film flow, gravity and electromagnetic waves. Most researchers, scientists and engineers used to solve Eq. 1 by converting the third order differential equations to a system of first order equations three times the dimension (Mechee et al., 2013a). However, it is more efficient if the problem can be directly solved using numerical methods. Such a type of work can be seen in Awoyemi and Idowu (2005), Waelah et al. (2011), Zainuddin (2011) and Jator (2010). All methods previously discussed are multistep methods; hence they need the starting values when used to solve ODEs Eq. 1. Most of the methods for solving ODEs can also be adapted for solving delay differential equations (DDEs).
In recent years there has been a growing interest in numerical solutions of DDEs, this is due
to the appearance of such equations in various areas such as neural network theory, epidemiology
and time lag control processes (Mechee et al., 2013b). DDEs also provide us with realistic model of
many phenomena arising in real world problems for example DDEs can be used in modeling of
population dynamics and spread of infectious diseases and two body problems of electrodynamics
(Bellen and Zennaro, 2003; Forde, 2005; Driver, 1977; Smith, 2011; Erneux, 2009; Kuang, 1993).
In this study, we are concerned with the one-step method particularly the Runge-Kutta method of
order three for directly solving third order ordinary differential equations. Accordingly, we have
developed a direct Runge-Kutta (RKD) method which can be directly used to solve Eq. 1. The
advantage of the new method over multistep methods is that it is self-starting. The method is then
adapted for solving special third order DDEs with multiple delays. The third order DDE can be
written in the following form:

\[ y''(t) = f(t, y(t), y(t-\tau_1), y(t-\tau_2), \ldots, y(t-\tau_n)), \quad t > t_0 \]  

(2)

with initial conditions:

\[ y(t) = \varphi(t), \quad y'(t) = \varphi'(t), \quad y''(t) = \varphi''(t), \quad t \leq t_0 \]

where, \( \varphi \) is a continuous function and \( \tau_1, \tau_2, \ldots, \tau_n \) are time delays.

Numerical results on two sets of problems consisting of ordinary and delay differential equations
are given and compared with the numerical results when the problems are reduced to a system of
first order ODEs and DDEs, respectively and solve using Runge-Kutta methods. Stability
polynomial of the method when applied to linear third order DDE is also presented. Numerical
results on two sets of problems consisting of ordinary and delay differential equations are given and
compared with the numerical results when the problems are reduced to a system of first order ODEs
and DDEs respectively and solve using Runge-Kutta methods.

**Derivation of RKD method**

The general form of RKD method with s-stage for solving initial value problem Eq. 1 can be
written as:

\[ y_{n+1} = y_n + hy_n' + \frac{h^2}{2} y_n'' + h \sum_{i=1}^{s} b_i k_i \]  

(3)

\[ y_{n+1}' = y_n' + hy_n'' + h \sum_{i=1}^{s} b_i k_i \]  

(4)

\[ y_{n+1}'' = y_n'' + h \sum_{i=1}^{s} b_i k_i \]  

(5)

Where:

\[ k_i = f(t_n + \tau_i, y_n) \]  

103
\[ k_i = f \left( t_n + c_i h, y_n + b_i y_n' + \frac{(c_i h)^2}{2} y_n'' + h^3 \sum_{j=1}^{s} a_{ij} y_{n-j} \right) \]  \hspace{1cm} (7)

for \( i = 2, 3, \ldots, s \). The parameters of RKD method are \( c_i, a_{ij}, b_i, b'_i \) for \( i = 1, 2, \ldots, s \) and \( j = 1, 2, \ldots, s \) are assumed to be real. If \( a_{ij} = 0 \) for \( i < j \), it is an explicit method and otherwise implicit method. The RKD method can be expressed in Butcher notation using the table of coefficients as follows:

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( \Lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b' )</td>
<td>( b'' )</td>
</tr>
<tr>
<td>( b''' )</td>
<td></td>
</tr>
</tbody>
</table>

To determine the coefficients of the RKD method, the expressions given in Eq. 3-7 are expanded using Taylor's series expansion. After some algebraic manipulations this expansion a equated to the true solution which are given by Taylor's series expansion. General order conditions for the RKD method can be found from the direct expansion of the local truncation error. The order conditions can found in Mechee et al. (2013a) which introduced direct Runge-Kutta (RKD) method of order five with three stage for solving thin film problem.

**ORDER CONDITIONS OF THE METHOD**

Mechee et al. (2013b) derived the order condition of RKD method up to order six. In this study using the same technique to get the order conditions up to order four. The order conditions for two-stage third order RKD method can be written as follows.

Order conditions for \( y \):

- **Order 3**
  \[ \sum b_i = \frac{1}{6} \]  \hspace{1cm} (8)

- **Order 4**
  \[ \sum b_i c_i = \frac{1}{24} \]  \hspace{1cm} (9)

Order conditions for \( y' \):

- **Order 2**
  \[ \sum b'_i = \frac{1}{2} \]  \hspace{1cm} (10)

- **Order 3**

104
\[
\sum b_i c_i = \frac{1}{6}
\]  \quad (11)

- Order 4

\[
\sum b_i c_i^2 = \frac{1}{12}
\]  \quad (12)

Order conditions for \( y'' \):

- Order 1

\[
\sum b_i = 1
\]  \quad (13)

- Order 2

\[
\sum b_i c_i = \frac{1}{2}
\]  \quad (14)

- Order 3

\[
\sum b_i c_i^2 = \frac{1}{3}
\]  \quad (15)

- Order 4

\[
\sum b_i c_i^3 = \frac{1}{4}, \sum b_i c_i^4 = \frac{1}{24}
\]  \quad (16)

All indices are from 1 to \( s \). To get third-order RKD method, the following simplifying assumption is used in order to reduce the number of equations to be solved:

\[
b_i = b_i (1 - c_i), i = 1, 2
\]  \quad (17)

**ZERO STABILITY OF THE METHODS**

Next, we will discuss the zero-stability of the methods it is one of the criteria for the method to be convergent. Zero-stability is an important tool for proving the stability and convergence of linear multistep methods. The interested reader is referred to the textbooks by Lambert (1991) and Butcher (2008) in which zero-stability is discussed. Zero-stability has been discussed in Hairer et al. (2010), where it is used to determine an upper bound on the order of convergence of linear multistep methods.

In studying the zero stability of RKD method, we can write method, Eq. 2-4 as follows:
\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
y_{n+1} \\
y_{n} \\
h^2y_{n+1}
\end{pmatrix} =
\begin{pmatrix}
1 & 1 & \frac{1}{2} \\
0 & 1 & 1 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
y_{n} \\
y_{n} \\
h^2y_{n}
\end{pmatrix}
\] (18)

\[
\rho(\varepsilon) = |I - A| =
\begin{vmatrix}
-1 & -1 & \frac{1}{2} \\
0 & -1 & -1 \\
0 & 0 & -1
\end{vmatrix}
\] (19)

Thus the characteristic polynomial is:

\[
\rho(\varepsilon) = (\varepsilon - 1)^3
\] (20)

Hence, the method is zero-stable since the roots are \( \varepsilon = 1, 1, 1 \), are less or equal to one.

**DERIVATION OF THIRD ORDER RKD METHODS**

The RKD method of s-stage and p-order can be derived by solving the order conditions of the method. The system of nonlinear equations (order conditions) of the method depend on p, however, the existence of the solutions of this system depends on the number of coefficients of the method which depends on the s-stage of the method in addition to the number of independent order conditions of the method.

**DERIVATION OF TWO-_STAGE THIRD-ORDER RKD METHODS**

To derive the two-stage and third-order RKD method, we use the algebraic conditions up to order three in the equations of order conditions in \( y, y' \) and \( y'' \) in Eq. 8, 10, 11 and 13-15. The resulting system of equations consists of six nonlinear equations with 8 unknowns variables to be solved. Consequently, there is a solution with two free parameters \( b_1 \):

\[
b_2 = \frac{1}{6} b_1
\]

and \( a_{31} \) however we chose one of them arbitrary;

\[
a_{31} = \frac{11}{200}
\]

but the second free coefficients can be chosen using minimization of the truncation error. Accordingly Dormand (1986) the free parameters can be chosen by minimizing the global error of the fourth order conditions minimize the global error. The technique is as follows:

- First: We find the error coefficients of \( y, y' \) and \( y'' \) respectively as the following:
Table 1: Butcher tableau

<table>
<thead>
<tr>
<th>0</th>
<th>11/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3</td>
<td>1/15</td>
</tr>
<tr>
<td>1/4</td>
<td>1/2</td>
</tr>
<tr>
<td>1/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>

\[
\| \tau^{(4)} \|_2 = \sqrt{\frac{-5}{72} + \left( \frac{2}{3} b_1 \right)^2}
\]

\[
\| \tau^{(4)} \|_2 = \sqrt{\left( \frac{1}{36} \right)^2 + \left( \frac{1}{72} \right)^2} = \frac{5}{5184}
\]

\[
\| \tau^{(4)} \|_2 = \frac{1}{216} \sqrt{91 + 26244a_{21}^2 - 291a_{31}}
\]

- Second: We find the global error norm as the following:

\[
\| \tau^{(4)}_{g} \|_2 = \frac{1}{216} \sqrt{361 - 4320b_i + 20736b_i^2 + 181a_{31}}
\]

Finally, we minimize the four truncation errors in Eq. 21-24 with respect to the free parameter \( b_i \), in this case we get the free parameter as \( b_i = 1/10 \). The error norms for \( y_n, y'_n \) and \( y''_n \) are given by:

\[
\| \tau^{(4)} \|_2 = 2.777 \times 10^{-3}, \quad \| \tau^{(4)} \|_2 = 3.105 \times 10^{-3}, \quad \| \tau^{(4)} \|_2 = 1.46 \times 10^{-3}
\]

respectively, where \( \tau^{(4)}, \tau^{(4)}_g \) and \( \tau^{(4)}_s \) are error terms of the fourth-order conditions for \( y, y' \) and \( y'' \) respectively. The RKD method of order three and two-stage is denoted by RKD3 which can be expressed in the following Butcher tableau in the Table 1.

**Adapting RK methods for directly solving third order DDEs**: Consider problem Eq. 2 for initial value problem for delay differential equations. To find the solution \( y_{n+1} \) at the point \( t_{n+1} \) for \( i = 0, 1, \ldots, n-1 \), we need the values of solutions at point \( t_i \) and all time delays points \( t_i \tau_i, t_i \tau_2, \ldots, t_i \tau_n \) for \( i = 0, 1, \ldots, n \) (Meeche et al., 2013a). The general form of RKD methods with s-stage for solving the initial value problem DDEs Eq. 2 can be written as the following:

\[
y_{n+1} = y_{n} + hy_{n} + \frac{h^2}{2} y_{n} + h^2 \sum_{i=-b}^{s} b_i k_i
\]

\[
y'_{n+1} = y'_{n} + hy'_{n} + h^2 \sum_{i=-b}^{s} b_i k_i
\]
\[ y_{n+1} = h' + h \sum_{i=1}^{n} b_i k_i \] (27)

\[ k_i = f(t_n, y_n, y(t_n - \tau_1), y(t_n - \tau_2), \ldots, y(t_n - \tau_s)) \] (28)

\[ k_i = f \left( t_n + c_i h, y_n + h c_i y_n + h^2 c_i y_n + h^3 \sum_{j=1}^{i-1} a_{ij} k_j, y(t_n + h c_i - \tau_i), y(t_n + h c_i - \tau_j), \ldots, y(t_n + h c_i - \tau_s) \right) \] (29)

The parameters of the methods Eq. 25-29 are \( c_i, a_{ij}, b_i, b_i^- \) and \( a_{ij} \) for \( i = 1, 2, 3, \ldots, s \) and \( j = 1, 2, 3, \ldots, s \) are given in Table 1.

**TIME DELAY INTERPOLATION**

Let the interval of the differential Eq. 2 be \( I = [a, b] \) and \( t_i = a + ih \) for \( i = 0, 1, 2, \ldots, n \) and:

\[ h = \frac{b-a}{n} \]

where \( n \) the number of points in the interval \( I \).

The numerical method approximates the solution \( y_{n+1} \) at the point \( t_{n+1} \) for \( i = 0, 1, 2, \ldots, n-1 \),

Hence:

\[ y_{i+1} = (y_{i+1}(t_i), y(t_i, \tau_1), y(t_i, \tau_2), \ldots, y(t_i, \tau_s)) \] (30)

To evaluate the delay term we used cubic interpolation and the details of it can be referred to Mechee et al. (2013b).

**STABILITY OF THE METHODS WHEN APPLIED TO THIRD ORDER DDE**

To study the stability of numerical method (Eq. 3-5), consider the linear test equation:

\[ Y''(x) = \alpha^2 y(x) + \mu^2 y(x - \tau) \] (31)

when, the method is applied to the linear test Eq. 31, (Jiaoxun and Yuhao, 2005; Hongjong and Jiaoxun, 1995; Al-Mutib, 1984; Liu and Spijker, 1990), we have:

\[ y_{n+1} = y_n + h y_n' + h^2 \sum_{i=1}^{n} b_i k_i \left( x_n + c_i h, y_n + h c_i y_n + h^2 y_n''(t_n + c_i h - \tau_i) \right) \]

\[ = y_n + h y_n' + h^2 y_n'' + h^3 \sum_{i=1}^{n} b_i k_i \left( \alpha^2 y_n + \mu^2 y(t_n + c_i h - \tau_i) \right) \]

\[ y_{n+1} = y_n' + h y_n'' + h^2 \sum_{i=1}^{n} b_i k_i \left( x_n + c_i h, y_n + h c_i y_n + h^2 y_n''(t_n + c_i h - \tau_i) \right) \]

\[ = y_n' + h y_n'' + h^2 \sum_{i=1}^{n} b_i k_i \left( \alpha^2 y_n + \mu^2 y(t_n + c_i h - \tau_i) \right) \]
\[ y_{n+1} = y_n + h \sum_{i=1}^{5} b_i k_i \left( x_n + c_i h, y_n + h c_i, Y_i, y(x_n + c_i h - \tau) \right) \]
\[ = y_n + h \sum_{i=1}^{5} b_i k_i \left( \alpha_i Y_i + \mu_i y(x_n + c_i h - \tau) \right) \]

Where:
\[ Y_i = y_n + c_i h y_n' + \frac{h^2}{2} y_n'' + h^3 \sum_{i=1}^{i} a_i f \left( x_n + c_i h, Y_i, y(x_n + c_i h - \tau) \right) \]
\[ = y_n + h y_n' + \frac{h^2}{2} y_n'' + h^3 \sum_{i=1}^{i} a_i f \left( \alpha_i Y_i + \mu_i y(x_n + c_i h - \tau) \right) \]

and:
\[ zn + 1 = T zn + (\alpha h) BY + (\mu h) B Z_n(\tau) \]

such that:
\[ Z_n = \begin{pmatrix} y_n \\ h y_n' \\ h^2 y_n'' \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} b_1 & b_2 & b_3 \\ b_1' & b_2' & b_3' \\ b_1'' & b_2'' & b_3'' \end{pmatrix} \]

So:
\[ \frac{Y_1}{Y_2} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{pmatrix} + \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{pmatrix} h y_n' + \frac{h^2}{2} \begin{pmatrix} c_1^2 \\ c_2^2 \\ \vdots \\ c_t^2 \end{pmatrix} y_n'' + \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1s} \\ \vdots & \ddots & \vdots \\ \alpha_{t1} & \cdots & \alpha_{ts} \end{pmatrix} \begin{pmatrix} \alpha_i Y_i + \mu_i y(x_n + c_i h - \tau) \\ \alpha_i Y_i + \mu_i y(x_n + c_i h - \tau) \\ \vdots \end{pmatrix} \]

and:
\[ \frac{Y_1}{Y_2} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{pmatrix} + \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{pmatrix} h y_n' + \frac{h^2}{2} \begin{pmatrix} c_1^2 \\ c_2^2 \\ \vdots \\ c_t^2 \end{pmatrix} y_n'' + \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1s} \\ \vdots & \ddots & \vdots \\ \alpha_{t1} & \cdots & \alpha_{ts} \end{pmatrix} \begin{pmatrix} \alpha_i Y_i + \mu_i y(x_n + c_i h - \tau) \\ \alpha_i Y_i + \mu_i y(x_n + c_i h - \tau) \\ \vdots \end{pmatrix} + \begin{pmatrix} \mu y(x_n + c_i h - \tau) \\ \mu y(x_n + c_i h - \tau) \\ \vdots \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{pmatrix} + \begin{pmatrix} \mu y(x_n + c_i h - \tau) \\ \mu y(x_n + c_i h - \tau) \\ \vdots \end{pmatrix} \begin{pmatrix} c_1^2 \\ c_2^2 \\ \vdots \\ c_t^2 \end{pmatrix} + \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} h y_n' + H c \end{pmatrix} \]
where, $H_u = (\alpha h)^3$, $H_v = (\mu h)^3$.

So:

\[
\begin{pmatrix}
Y_t \\
Y_{n+1}
\end{pmatrix} = (1 - H\alpha)^{-1}(CZ_n + H\alpha A)
\begin{pmatrix}
\mu y(x_n + c_1 h - \tau) \\
\mu y(x_n + c_2 h - \tau) \\
\vdots \\
\mu y(x_n + c_n h - \tau)
\end{pmatrix}
\]

where:

\[
A = \begin{pmatrix}
\alpha_1 & \cdots & \alpha_n \\
\vdots & \ddots & \vdots \\
\alpha_1 & \cdots & \alpha_n
\end{pmatrix},
C = \begin{pmatrix}
1 & c_1 & \frac{c_1^2}{2} \\
1 & c_2 & \frac{c_2^2}{2} \\
\vdots & \vdots & \vdots \\
1 & c_n & \frac{c_n^2}{2}
\end{pmatrix}
\]

Hence, the stability polynomial of the method is:

\[
Z_{n+1} = T Z_n + H_u B(1 - H\alpha A)^{-1}(CZ_n + H\alpha Z_n(\tau))
\]

\[
+ H_v B Z_n(\tau) = T Z_n + T_2 Z_n(\tau)
\]

where:

\[
Z_n(\tau) = \begin{pmatrix}
y(x_n + c_1 h - \tau) \\
y(x_n + c_2 h - \tau) \\
\vdots \\
y(x_n + c_n h - \tau)
\end{pmatrix},
T_1 = T + H_u B(1 - H\alpha A)^{-1} C \text{ and } T_2 = H_v B \left[ H\alpha (1 - H\alpha A)^{-1} A + I \right]
\]

**NUMERICAL RESULTS**

In this section a set of third order ordinary and delay differential equations are solved using RKD3 method of order three and numerical results are compared with the existing RK methods of the orders three and four.

The following notations are used in Fig. 1-8:

- **h**: Stepsize used
- **RKD3**: The Direct Rung-Kutta method of order three derived in section 5
- **RK3**: Existing Runge-Kutta method order three as given in Butcher (2008)
- **RK4**: Existing Runge-Kutta method order fourth as in Dormand (1996)
- **Total time**: The total time in second to solve the problems
- **MAX ERROR**: $\max |y(x_f) - y_{ref}|$ Absolute value of the true solution minus the computed solution
Fig. 1: A log max of errors versus computational time for problem 1

Fig. 2: A log max of errors versus computational time for problem 2

Fig. 3: A log max of errors versus computational time for problem 3
Fig. 4: A log max of errors versus computational time for problem 4

Fig. 5: A log max of errors versus computational time for problem 5

Fig. 6: A log max of errors versus computational time for problem 6
Fig. 7: A log max of errors versus computational time for problem 7

Fig. 8: A log max of errors versus computational time for problem 8

PROBLEMS OF ODES

Problem 1 (Homogenous linear):

\[ y^{(3)}(t) = -y(t), \quad 0 < t < b \]

- Initial conditions:

\[ Y(0) = 1, \quad y'(0) = -1, \quad y''(0) = 1, \]

- Exact solution:

\[ y(t) = e^t, \quad b = 1 \]

Problem 2 (Non homogenous linear):

\[ y^{(3)}(t) = -e^{-t}, \quad 0 < t < b \]

113
Initial conditions:

\[ Y(0) = 1, \ y'(0) = -1, \ y''(0) = 1 \]

Exact solution:

\[ y(t) = e^{-t}, \ b = 1 \]

**Problem 3 (Homogenous non linear):**

\[ y'''(t) = \frac{3}{8y^2(t)}, \ 0 < t < b \]

Initial conditions:

\[ t(0) = 1, y'(0) = \frac{1}{2}, \ y''(0) = -\frac{1}{4} \]

Exact solution:

\[ y(t) = \sqrt{1 + t}, \ b = \pi \]

**Problem 4 (Non homogenous linear):**

\[ Y'''(t) = -\delta Y^4, \ 0 < t < b \]

Initial conditions:

\[ Y(0) = 1, \ y'(0) = -1, \ y''(0) = 2 \]

Exact solution:

\[ y(t) = \frac{1}{1 + t}, \ b = \pi \]

**PROBLEMS OF DDEs**

**Problem 5 (Homogenous non linear):**

\[ Y'''(t) = -e^{-t} \text{te}^t y^2(t-1), \ t > 0 \]

Initial conditions:

\[ Y(0) = 0, \ y'(0) = -1, \ y''(0) = 1 \]
• Exact solution:

\[ y(t) = e^{-t}, \ b = 1 \]

Problem 6 (Homogenous linear):

\[ y''(t) - e^{-t}y(t-\tau), \ t > 0 \]

• Initial conditions:

\[ Y(0) = 1, \ y'(0) = -1, \ y'(0) = 1 \]

• Exact solution:

\[ y(t) = e^{-t}, \ b = 1 \]

Problem 7 (Non homogenous non linear)

\[ y'''(t) = y''(t-\tau) + y'(t-\tau) + \frac{3}{8y'(x)} - 2(1 + t) + \tau, \ t > 0 \]

• Initial conditions:

\[ y(0) = 1, \ y'(0) = 1 \]

• Exact solution:

\[ y(t) = \sqrt{1+t}, \ b = \pi \]

Problem 8 (Non homogenous linear)

\[ y'''(t) = y'(t-\tau) + y(t-\tau) - \ln(1 + t - \tau) - \ln(1 + t - \tau_x) + \frac{2}{(1 + t)^3}, \ t > 0 \]

• Initial conditions:

\[ Y(0) = 0, \ y'(0) = 1, \ y'(0) = -1 \]

• Exact solution:

\[ y(t) = \ln(1+t), \ b = \pi \]

DISCUSSION AND CONCLUSION

In this study, we have derived the RKD method of two-stage, third order. This method has been adapted with delay differential equations. The zero-stability of the method is proven. The stability
polynomial of the method when applied to linear third order DDE is also given. We used the method for solving special third order ODEs as well as DDEs. For DDEs cubic interpolation is used to evaluate the delay terms. Numerical results show that the RKD method is more accurate and requires less computational time compared to the existing RK methods.

REFERENCES