General Maple Code for Solving Scalar Linear Neutral Delay Differential Equations

Mohamed S. M. Bahgat

Mathematics Department, Faculty of Science, Minia University, Minia 61519, Egypt
E-mail: mohamed.bahagat@mu.edu.eg

Abstract: The objective is to demonstrate how to create straightforward Maple programs for numerical computations and programs for condensing or changing mathematical formulas, polynomials, or symbolic expressions. It is assumed that readers are accustomed to using interactive Maple. The programming language used in Maple is interpreted and interactive. Due to the overhead of the interpreter, Maple is not appropriate for running programs that require a lot of numbers. Although it can be used to create numerical codes and for high-precision numerical calculations. This paper uses Maple's general code to compute the method of steps (MoS) solution of linear neutral and delay differential equations (DDEs). The paper relies on entering simple inputs to get a quick result explained by theoretical solutions and their graphics. A Maple symbolic computation was more efficient than a programming language computation for linear non-neutral DDEs. Maple was faster at solving linear neutral DDEs, which are usually more challenging. Using the MoS methodology, we list and talk about various examples of non-neutral DDEs and NDDEs reported in the literature.
Keywords: DDEs; NDDEs; Maple procedure; algorithm
2000 MSC. 65H10; 65H05; 65K05. 41A25.

1. Introduction

For their applications or modeling goals, scientists and engineers frequently need to use specialized software. Several elements, including complexity/speed, architecture, library usability, tolerances, etc., may have an impact on a model's accuracy. We look at how well Maple uses the approach of steps to solve linear delay differential equations (DDEs). With the aid of Maple, we may compute analytical solutions to linear neutral and non-neutral DDEs using symbolic mathematical computations and the method of steps. An analytical solution to DDEs can be difficult, even for linear DDEs. The derivative of the state variable appears to have a time delay, which makes neutral delay differential equations (NDDEs) generally more challenging to solve. Delay differential equations (DDEs) and ordinary differential equations (ODEs) are two different types of differential equations, and the latter is typically more challenging to solve because the derivatives of the state variables depend on the values of the state variables from the past. MoS is a traditional method for finding DDE solutions; however, it is impossible to move forward in time and find the solution in the far future. The imitative method for solving DDEs is a primary method called the method of steps [1-10]. When performed manually, it is generally known to be a laborious operation; however, using symbolic toolboxes in computer systems, the complexity can be greatly reduced [11]. Consider the following neutral delay differential equation with a fixed delay β:

$$
\begin{aligned}
y'(t) &= f\big(t, y(t), y(t-\beta), y'(t-\beta)\big), \quad t > t_0 \\
y(t) &= \varphi_0(t), \quad t \in [-\beta + t_0, 0] \\
y(t) &= y_0, \quad t = t_0.
\end{aligned}
\tag{1}
$$

Where β is the discrete or constant delay and t_0 is the initial time value. Solutions of neutral delay equations of type (1) are computed with the Maple procedure in the next sections. This paper has the following structure. In Section 2, the formulation of MoS for solving linear NDDEs is presented. In Section 3, a maple procedure for implementing MoS. In subsections 3.1 and 3.2, we provide various examples of DDEs, NDDEs and their associated MoS solutions. Conclusions on the symbolic calculations necessary to generate a DDE solution using the MoS are given in the last Section 4.

2. Formulation of the method of steps

To solve DDE initial value problems (IVPs), we need a historical function, $\varphi_0(t): [-\beta, t_0] \longrightarrow R$ and $y(t_0) = y_0$ is an initial value. Since we establish an unlimited set of initial conditions between $t \in [-\beta, t_0]$, this condition means that DDEs are in fact infinite-dimensional problems. The following NDDE is solved by $y_1(t)$, which is a solution, by using integration properties on the interval $[t_0, t_0 + \beta]$,

$$
\begin{aligned}
y_1'(t) &= f\big(t, y_1(t), \varphi_0(t-\beta), \varphi'_0(t-\beta)\big), \\
& t \in [t_0, t_0 + \beta] \\
y_1(t) &= \varphi_0(t), \quad t = t_0.
\end{aligned}
$$

We discover the general derivative formula shown below for the mth partition of the IVP $m \in N$ by carrying out this procedure repeatedly for new intervals and applying the solution for the previous interval:

$$y'_m(t) = f\big(t, y_m(t), y_{m-1}(t-\beta), y'_{m-1}(t-\beta)\big),$$
$$t \in [(m-1)\beta, m\beta],$$
$$y_m(t) = y_{m-1}(t), \quad t = (m-1)\beta.$$

On the interval $t \in [0, m\beta)$, the solutions of $\{y'_m(t)\}_{m=1}^{\infty}$ constitute a piecewise solution of the initial NDDE (1). The MoS offers an analytical solution for DDEs, but it has significant limitations, such as time consumption and the difficulty of the integrals that must be computed across each interval. Before using this methodology, a complete understanding of the past is also necessary. It can be challenging to examine the behavior of the solution or to carry out a stability analysis of $y(t)$ as $t \to \infty$. If we have a linear NDDE of first-order differential equation with a single constant delay and constant/variable coefficients

$$y'(t) = \alpha_1(t)y(t) + \alpha_2(t)y(t-\beta) + \alpha_3(t)y'(t-\beta) + f(t), \quad t \in [0, T],$$

$$y(t) = \varphi(t), \quad t \in [-\beta, 0], \tag{2}$$

where $\beta > 0$, $\alpha_1(t) \neq 0$ and $\alpha_2(t) \neq 0$. We will employ the procedure and its conditions. The history function is known on the interval $[-\beta, 0]$, because it is the function $\varphi(t)$. As a result, we may state that the equation is solved for the interval $[-\beta, 0]$. Since $t \in [0, \beta]$ and $t - \beta \in [-\beta, 0]$, then $y(t-\beta)$ becomes $y_0(t-\beta)$ on $[0, \beta]$. Therefore, the equation (2) in $[0, \beta]$ becomes

$$y'_1(t) = \alpha_1(t)y_1(t) + \alpha_2(t)y_0(t-\beta) + \alpha_3(t)y'_0(t-\beta) + f(t), \quad t \in [0, T] \tag{3}$$

$$y_0(t) = \varphi(0).$$

Since $y_0(t-\beta)$ and $y'_0(t-\beta)$ are known, it is $\varphi(t-\beta)$ and $\varphi'(t-\beta)$, hence equation (3) is an ordinary differential equation rather than a delay differential equation. So, given the initial condition, $y(0) = \varphi(0)$. we solve it on the interval $[0, \beta]$,

$$y'(t) - \alpha_1(t)y(t)$$
$$= \alpha_2(t)\varphi(t-\beta) + \alpha_3(t)\varphi'(t-\beta) + f(t), \quad t \in [0, \beta]$$

$$y(t) = \varphi(t), \quad t \in [-\beta, 0], \tag{4}$$

equation (4) has this general solution:

$$y_1(t) = \frac{1}{e^{\int -\alpha_1(t)dt}} \int e^{\int -\alpha_1(t)dt}\big(\alpha_2(t)\varphi(t-\beta) + \alpha_3(t)\varphi'(t-\beta) + f(t)\big)dt. \quad \text{on} \quad [0, \beta]. \tag{5}$$

Once more focusing on the interval $[\beta, 2\beta]$, the equation becomes

$$y'_2(t) - \alpha_1(t)y_2(t)$$
$$= \alpha_2(t)y_1(t-\beta) + \alpha_3(t)y'_1(t-\beta) + f(t), \quad t \in [\beta, 2\beta]$$

$$y(\beta) = y_1(\beta). \tag{6}$$

The solution can theoretically be found by continuing this process on the interval $[0, m\beta]$, $m = 1, 2, \cdots, k$.

3. A Maple procedure for implementing a method of steps

Here, we show distinct examples of non-neutral delay differential equations (DDEs) and neutral delay differential equations (NDDEs) described in [7, 8] and apply the MoS approach to each example to find the solution using the general code of Maple software. The examples that follow in this part serve to clearly show each command in the MoS procedure that is described below and that we refer to as ALGMETSTEPS. By specifying first the initial function $\varphi(t)$ and the coefficient functions $\alpha_1(t)$ and $\alpha_2(t)$, the solution of (3) on the interval $[0, m\beta]$ satisfying the initial condition can be found using ALGMETSTEPS, and entering the name and parameters of the procedure ALGMETSTEPS, and then calling it by typing its name. Use the key "Enter" twice to execute ALGMETSTEPS by executing this body of commands. In Section 3.1, ALGMETSTEPS is invoked to recalculate the delay equation. We can create our own functions in Maple using the Maple **proc** keyword.

The syntax is:

proc (< **argumentsSequence** >) [local < **NamesLocalVariables** >; global < **NamesGlobalVariables** >;]
  < **statement sequence** >
  end;

[ ] indicates optional parameters.
 • A list of variable names, perhaps null, separated by commas is known as a <**argumentsSequence** > The procedure's arguments are these variables.
 • The <**NamesLocalVariables**> variable name sequence is separated by commas.
 • The list of variables in local <**NamesLocalVariables**> is specific to the procedure. Any assignments made to these variables will only affect the procedure's scope. Regardless of the value of the variable outside the process, the values of these variables will be empty when the operation begins.
• Global <**NamesGlobalVariables**> is a list of variables that the method uses on a global scale. All assignments made throughout the operation will be of a global nature, and the variables' initial values will be those of the active Maple session.
• The procedure's "statement sequence" is its main body and can include any legal arrangement of Maple statements.
The following rules are used to determine whether variables are local or global if they are not explicitly declared in the preamble:
• Local variables are those that are assigned values in the procedure.
• Local variables are those that appear as the controlling

variables in a for...do loop.
• In addition to these variables, all others are global variables.

Declaring variables as local or global directly in the preamble is good programming practice since it prevents erroneous errors and makes the process simpler to understand. Now we describe the maple code of the proposed method.

where:
ALGMETSTEPS: is the name of procedure

> $ALGMETSTEPS := \textbf{proc}\left(\text{Phi, a, b, } c\text{, T, β, f, k} :: posint\right)$

Phi: is the initial function
$a = \alpha1(t), b = \alpha2(t)$ and $c = \alpha3(t)$: are the coefficients of $y(t), y(t-\beta)$ and $y'(t-\beta)$ respectively.
T: the starting point of the interval.
β: is the constant delay.
k: is the end point of the interval.
f: is the function of t.

```
> restart;

> #DE := d/dt y(t) = α1(t)·y(t) + α2(t)·y(t−β) + α3(t)·y'(t−β) +f(t);

# where ss[i] = y(t−β), sd[i] = y'(t−β)

> ALGMETSTEPS := proc( Phi, a, b, c, T, β, f, k :: posint)
   local i, y, DE, IC, Soln, s, sd, ss, Ysol, u, α1, α2, α3;
   global ysol;
   α1 := t→a;
   α2 := t→b;
   α3 := t→c;
   ss[0] := eval( Phi, t=t−β);
   s[0] := Phi;
   sd[0] := eval( d/dt (Phi), t=t−β);
   for i from 0 to k−1 do
      DE := d/dt y(t) = α1(t)·y(t) + α2(t)·ss[i] + α3(t)·sd[i] +f;
      IC := y(T+i·β) = eval( s[i], t=T+i·β);
      Soln := dsolve( {DE, IC}, y(t));
      s[i+1] := rhs( Soln);
      if T−β < T+i·β−β < T then
         sd[i+1] := sd[0];
         ss[i+1] := ss[0];
      else
         ss[i+1] := eval( s[i+1], t=t−β);
         sd[i+1] := eval( d/dt s[i+1], t=t−β);
      end if;
      Ysol := t→s[0] + Σ_{j=0}^{k−1} (s[j+1] − s[j]).Heaviside(t−T−j·β);
   end do;
   u := simplify( convert( Ysol(t), piecewise, t));
   ysol := unapply(u, t);
   ysol(t); plot( ysol(t), t=0 ..T+k·β, discont=true, labels=[t, y], labelfont
      = [TIMES, BOLD, 16], thickness = 7, axis = [gridlines = [10, color = blue]],
      axesfont=[TIMES, 16])
end proc;
```

The non-neutral DDEs and NDDEs established in [12] are presented and discussed in the following two sections, then we employ Maple software to identify solutions based on the MoS technique.

3.1 Linear DDE examples

In order to solve this type of equation, we need to implement the general Maple program code ALGMETSTEPS, where we need to enter β, the historical function $\varphi(t)$, and the three coefficients $y(t), y(t-\beta)$, and $y'(t-\beta)$, where $y'(t-\beta)$ equals zero for linear non-neutral DDE and finally, we enter the value of $f(t)$. Below are a few examples of how the general code can be applied to each problem and found solutions and graphs. These examples illustrate the individual commands of our algorithm of steps procedure, known as ALGMETSTEPS, described below. With ALGMETSTEPS, the solution of

$$y'(t) = \alpha_1(t)y(t) + \alpha_2(t)y(t-\beta) + f(t), \quad t \in [0, m\beta]$$
$$y(t) = \varphi(t), \quad t \in [-\beta, 0].$$

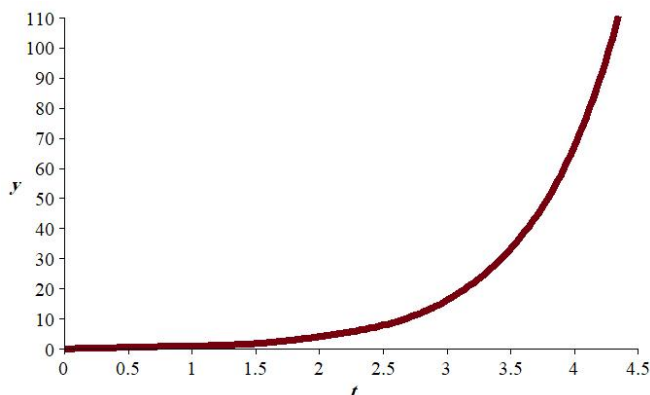**Example 1:** In this example, we will use the classic work by [1].
$$y'(t) = 6y(t-1),$$
on the interval $[1,4]$ satisfying the initial condition
$$y(t) = \varphi_0(t) = t, \quad 0 \le t \le 1.$$
Since the initial interval is $[0,1]$, Invoking *ALGMETSTEPS* with $Phi = \varphi(t) = t$, $\alpha1 = 0$, $\alpha2 = 6$, $\alpha3 = 0$, $T = 1$, $\beta = 1$ and $k = 4$. The values of the solution is calculated at $t = 1.5, 2, 2.5, 3$. The solution and its graph are shown below:

> $ALGMETSTEPS(t, 0, 6, 0, 1, 1, 0, 4);$

> $'y(t)' = simplify(ysol(t));$

$$y(t) = \begin{cases} t & t \le 1 \\ 3t^2 - 6t + 4 & t \le 2 \\ 6t^3 - 36t^2 + 78t - 56 & t \le 3 \\ 9t^4 - 108t^3 + 504t^2 - 1056t + 835 & t \le 4 \\ \frac{54}{5}t^5 - 216t^4 + 1764t^3 - 7272t^2 + 15072t - \frac{62641}{5} & 4 < t \end{cases}$$

> $'y(1.5) = ysol(1.5);' 'y(2) = ysol(2);' 'y(2.5) = ysol(2.5);' 'y(3) = ysol(3);$

$$y(1.5) = 1.75$$
$$y(2) = 4$$
$$y(2.5) = 7.750$$
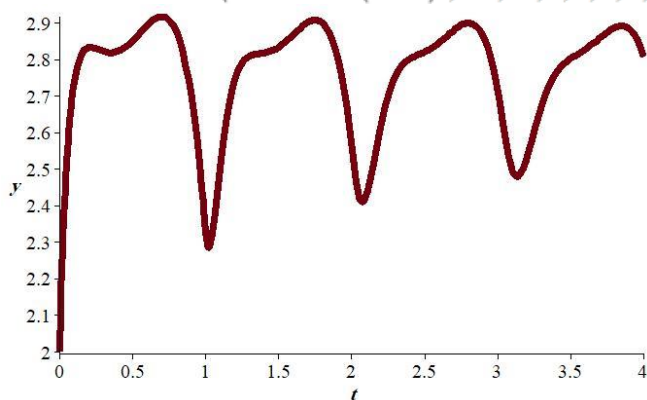$$y(3) = 16$$

Example 2: Find the solution for DDE
$$y'(t) = -15y(t) + 15y(t-1),$$
on the interval $[0,3]$ satisfying the initial condition
$$y(t) = \varphi_0(t) = -6t(t+1)^3 - t + 2, \quad -1 \le t \le 0$$
Since the initial interval is $[-1,0]$, Invoking *ALGMETSTEPS*
with $Phi = \varphi(t) = -6t(t+1)^3 - t + 2$, $\alpha1 = -15$, $\alpha2 = 15$, $\alpha3 = 0$, $T = 0$, $\beta = 1$ and $k = 4$. The solution and its graph are shown below:

> $ALGMETSTEPS\left(2 - t - 6 \cdot t \cdot (t+1)^3, -15, 15, 0, 0, 1, 0, 4\right);$

> `'y(t)'= simplify( ysol(t));`

Example 3: Consider the DDE is shown below:
$$y'(t) = -\frac{1}{5}y(t) - \frac{5}{8}y(t-3), \qquad t \ge 0$$
on the interval $[0,3]$ satisfying the initial condition
$$y(t) = \varphi_0(t) = (t+\tfrac{3}{2})^2, \qquad -3 \le t \le 0$$
Since the initial interval is $[-3,0]$, Invoking *ALGMETSTEPS*
with $Phi = \varphi(t) = (t + \tfrac{3}{2})^2$, $\alpha1 = -\tfrac{1}{5}$, $\alpha2 = -\tfrac{5}{8}$, $\alpha3 = 0$, $T = 0$, $\beta = 3$ and $k = 2$. The solution and its graph are shown below::

> $ALGMETSTEPS\left(\left(t+\tfrac{3}{2}\right)^2, -\tfrac{1}{5}, -\tfrac{5}{8}, 0, 0, 3, 0, 2\right);$
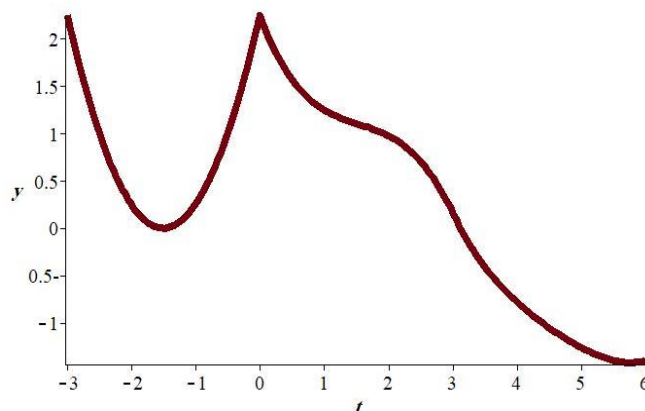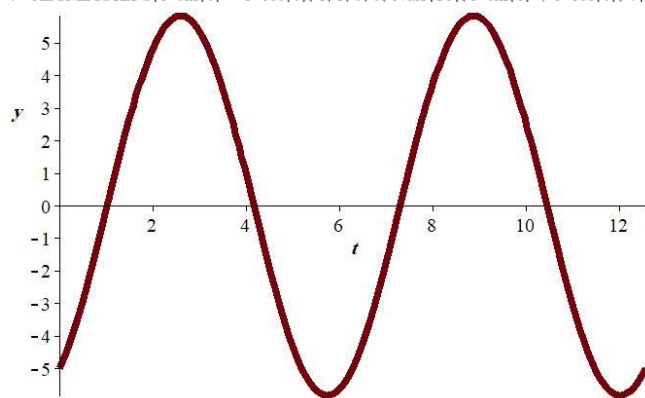

> `'y(t)'= simplify( ysol(t));`

Example 4: Find the solution for the following DDE
$$y'(t) = y(t) + y(t - \pi) + 3\cos(t) + 5\sin(t), \qquad t \ge 0$$
$$y(t) = \varphi_0(t) = 3\sin(t) - 5\cos(t), \qquad t \le 0$$
Since the initial interval is $[-\pi, 0]$, Invoking *ALGMETSTEPS*
with $Phi = \varphi_0(t) = 3\sin(t) - 5\cos(t)$, $\alpha1 = -1$, $\alpha2 = 1$, $\alpha3 = 0$, $T = 0$, $\beta = \pi$ and $k = 4$, $f(t) = 3\cos(t) + 5\sin(t)$. The graph and the solution shown below:

> $ALGMETSTEPS(3 \cdot \sin(t) - 5 \cdot \cos(t), 1, 1, 0, 0, \text{evalf}(Pi), 5 \cdot \sin(t) + 3 \cdot \cos(t), 4);$
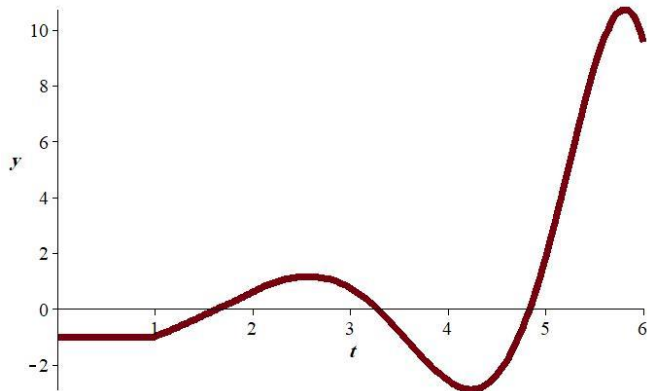

Example 5: Find the solution for the following DDE with variable coefficients
$$y'(t) = -\frac{1}{t+1}y(t) - ty(t-1), \qquad t \ge 1$$
$$y(t) = \varphi_0(t) = -1, \qquad t \le 1$$
Since the initial interval is $[0,1]$, Invoking *ALGMETSTEPS*
with $Phi = \varphi(t) = -1$, $\alpha1 = --\frac{1}{t+1}$, $\alpha2 = -t$, $\alpha3 = 0$, $T = 0$, $\beta = 1$ and $k = 5$. The solution and its graph are shown below:

> $ALGMETSTEPS\left(-1, -\dfrac{1}{1+t}, -t, 0, 1, 1, 0, 5\right);$

> $'y(t)' = simplify(ysol(t));$

$$y(t) = \begin{cases} -1 & t \le 1 \\[2mm] \dfrac{1}{6}\dfrac{2t^3 + 3t^2 - 17}{1+t} & t \le 2 \\[3mm] -\dfrac{1}{120}\dfrac{8t^5 - 5t^4 - 20t^3 - 160t^2 - 320t + 1044}{1+t} & t \le 3 \\[3mm] \dfrac{1}{5040}\dfrac{48t^7 - 259t^6 + 294t^5 - 1365t^4 - 2940t^3 + 25431t^2 + 50862t - 164133}{1+t} & 3 < t \end{cases}$$

3.2 Linear neutral DDE examples

ALGMETSTEPS is a general Maple program code we can use it to solve this type of equation., where we need to enter β, the historical function $\varphi(t)$, and the three coefficients of $y(t)$, $y(t - \beta)$, and $y'(t - \beta)$. Below are a few examples of how the general code can be applied to each problem and found solutions and graphs. These examples illustrate the individual commands of our algorithm of steps procedure, known as ALGMETSTEPS, described below. With ALGMETSTEPS, the solution of
$y'(t) = \alpha_1(t)y(t) + \alpha_2(t)y(t - \beta) + \alpha_3(t)y'(t - \beta) + f(t), \qquad t \in [0, m\beta],$
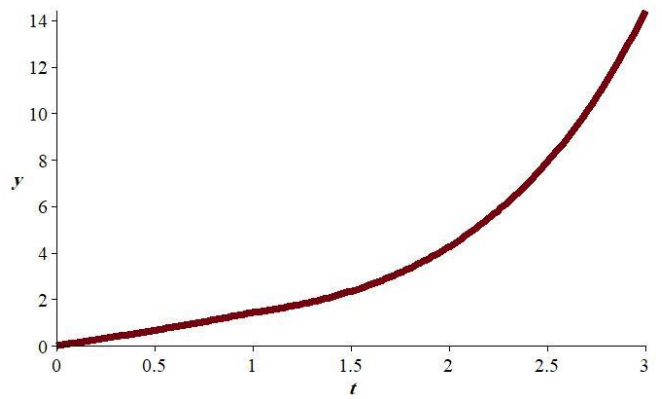$\quad y(t) = \varphi(t), \ t \in [-\beta, 0].$

Example 6: Consider the NDDE is shown below:
$$y'(t) = y(t) + y(t-1) - \frac{1}{4}y'(t-1), \qquad t \ge 0$$
$$y(t) = \varphi_0(t) = -t, \qquad -1 \le t \le 0$$
Since the initial interval is $[-1,0]$, Invoking *ALGMETSTEPS* with $Phi = \varphi(t) = -t, \alpha1 = 1, \ \alpha2 = 1, \ \alpha3 = -\frac{1}{4}, \ T = 0,$ $\beta = 1$ and $k = 5$. Below we find the solution and its graph:

> $ALGMETSTEPS\left(-t, 1, 1, -\dfrac{1}{4}, 0, 1, 0, 3\right);$

> $'y(t)' = simplify(ysol(t));$

$$y(t) = \begin{cases} -t & t \le 0 \\[2mm] t - \dfrac{1}{4} + \dfrac{1}{4}e^t & t \le 1 \\[3mm] -t + \dfrac{1}{2} + \dfrac{3}{16}te^{t-1} + \dfrac{17}{16}e^{t-1} + \dfrac{1}{4}e^t & t < 2 \\[3mm] t - \dfrac{3}{4} + \dfrac{9}{128}t^2 e^{t-2} + \dfrac{39}{64}e^{t-2}t + \dfrac{3}{16}te^{t-1} + \dfrac{1}{4}e^t + \dfrac{17}{16}e^{t-1} - \dfrac{17}{4}e^{t-2} & 2 \le t \end{cases}$$
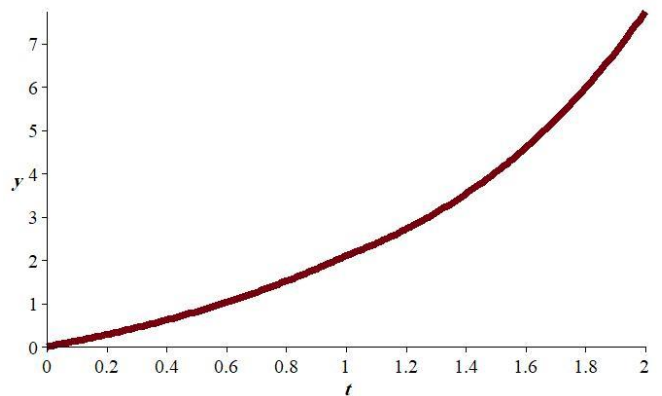
Example 7: Let us consider the following NDDE:
$$y'(t) = y(t) + y(t-1) - \frac{1}{4}y'(t-1) + \sin(t), \qquad t \ge 0$$
$$y(t) = \varphi_0(t) = -t, \qquad -1 \le t \le 0$$
Since the initial interval is $[-1,0]$, Invoking *ALGMETSTEPS* with $Phi = \varphi(t) = -t, \alpha1 = 1, \ \alpha2 = 1, \ \alpha3 = -\frac{1}{4}, \ T = 0,$ $\beta = 1, f(t) = \sin(t)$ and $k = 5$. The solution and its graph are shown below:

> $ALGMETSTEPS\left(-t, 1, 1, -\dfrac{1}{4}, 0, 1, \sin(t), 2\right);$

$$y(t) = \begin{cases} -t & t \le 0 \\[2mm] -\dfrac{1}{2}\cos(t) - \dfrac{1}{2}\sin(t) + t - \dfrac{1}{4} + \dfrac{3}{4}e^t & t \le 1 \\[3mm] \dfrac{1}{16}\left(9te^t + 8\cos(t-1)e + 2\sin(t-1)e - 8\cos(t)e + 12e^{t+1} - 8\sin(t)e - 16te + 8e + 3e^t\right)e^{-1} & 1 < t \end{cases}$$
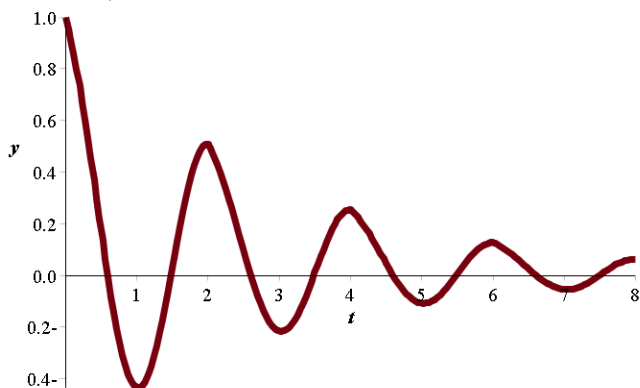
Example 8: Let us consider the following NDDE:
$$y'(t) = -2y(t) + y(t-2) - \frac{1}{2}y'(t-2), \qquad t \ge 0$$
$$y(t) = \varphi_0(t) = \sin(\pi t), \qquad -2 \le t \le 0$$
Since the initial interval is $[-2,0]$, Invoking *ALGMETSTEPS*

with $Phi = \varphi(t) = \sin(\pi t)$, $\alpha 1 = -2$, $\alpha 2 = 1$, $\alpha 3 = \frac{1}{2}$, $T = 0$, $\beta = 2$ and $k = 6$. The solution and its graph are shown below:

> $ALGMETSTEPS\left(\cos(\pi \cdot t), -2, 1, \frac{1}{2}, 0, 2, 0, 4 \right);$

> $'y(t)' = simplify(ysol(t));$

$$
y(t) = \begin{cases}
\cos(\pi t) & t \leq 0 \\
\frac{1}{2}e^{-2t} + \frac{1}{2}\cos(\pi t) & t \leq 2 \\
\frac{1}{2}e^{-2t} + \frac{1}{4}\cos(\pi t) + \frac{1}{4}e^{-2t+4} & t < 4 \\
\frac{1}{2}e^{-2t} + \frac{1}{8}\cos(\pi t) + \frac{1}{4}e^{-2t+4} + \frac{1}{8}e^{-2t+8} & t < 6 \\
\frac{1}{2}e^{-2t} + \frac{1}{16}\cos(\pi t) + \frac{1}{4}e^{-2t+4} + \frac{1}{8}e^{-2t+8} + \frac{1}{16}e^{-2t+12} & 6 \leq t
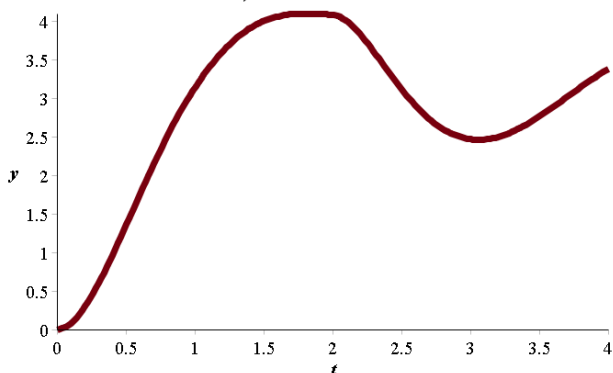\end{cases}
$$

Example 9: Let us consider the following NDDE:
$$y'(t) = -\frac{3}{2}y(t) - \frac{9}{10}y'(t-2), \qquad t \geq 0$$
$$y(t) = \varphi(t) = -\frac{5}{6}(t+2)t^3, \qquad -2 \leq t \leq 0$$

This example does not contain a state-dependent delay term. Since the initial interval is $[-2,0]$, Invoking ALGMETSTEPS with $Phi = \varphi(t) = -\frac{5}{6}(t+2)t^3$, $\alpha 1 = -\frac{3}{2}$, $\alpha 2 = 0$, $\alpha 3 = -\frac{9}{10}$, $T = 0$, $\beta = 2$ and $k = 3$. The solution and its graph are shown below:

> $ALGMETSTEPS\left(-\frac{5}{6} \cdot (t+2) \cdot t^3, -\frac{3}{2}, 0, -\frac{9}{10}, 0, 2, 6, 2 \right);$

$$
y(t) = \begin{cases}
-\frac{5}{6}(t+2)t^3 & t \leq 0 \\
-13t^2 + 2t^3 + \frac{88}{3}t - \frac{176}{9} + \frac{176}{9}e^{-\frac{3}{2}t} & t \leq 2 \\
-\frac{412}{5} + \frac{174}{5}t - \frac{18}{5}t^2 + \frac{132}{5}te^{-\frac{3}{2}t+3} - \frac{1012}{45}e^{-\frac{3}{2}t+3} + \frac{176}{9}e^{-\frac{3}{2}t} & 2 < t
\end{cases}
$$

4. Conclusion

This study provides valuable insights into the performance of Maple's general code in solving neutral and retarded linear DDEs using MoS. The results obtained demonstrate the effect of various factors on computation time, including the number of intervals employed, complexity of the history function, and DDE parameters. Maple symbolic computations were quicker than computations from programming languages for linear non-neutral DDEs. Maple was quicker at solving linear neutral DDEs, which are frequently more challenging to solve. We used Maple's general code to solve homogeneous and non-homogeneous DDEs and NDDES in this study. Overall, the study contributes to the development of a Maple code that can solve delayed and neutral systems of linear differential equations.

Abbreviations:

MoS: the method of steps
DDEs: Delay differential equation.
NDDEs: Neutral differential equations.
ODEs: Ordinary differential equations.
IVPs: Initial value problems

Acknowledgements

References:

[1] L.E. El'sgol'ts, Introduction to the Theory of Differential Equations with Deviating Arguments. 1st Ed. Holden-Day, Inc., San Francisco, 1973
[2] H.L. Smith, An Introduction to Delay Differential Equations with Applications to the Life Sciences. Springer Science & Business Media, New York, NY, USA, 2011
[3] K.L. Cooke, Differential-Difference equations. In International Symposium on Nonlinear Differential Equations and Nonlinear Mechanics. 1st Ed. Elsevier: Amsterdam, The Netherlands, 1963
[4] J.M. Heffernan, R.M. Corless, *Mathematical* Sciences, 31 (2006) 21–34.
[5] R.J. Bauer, G. Mo, W. Krzyzanski, *Computer Methods* and *Programs* in Biomedicine, 111 (2013) 715–734.
[6] T. Kalmár-Nagy, *Differential Equations and Dynamical Systems,* 17 (2009) 185–200.
[7] E. Kaslik, S. Sivasundaram, *Journal of Computational and Applied Mathematics*, 236 (2012) 4027–4041.

[8] E. Cimen, Y. Ekinci, International Journal of Mathematics and Computer Science, 1 (2017) 1–11.

[9] Q. Hongyu, Z. Qifeng, W. Shaohua, *Applied Mathematics and Computation,* 346 (2019) 76–85.

[10] G. González-Parra, S. Sultana, A.J. Arenas, *Mathematics,* 10 (2022) 354-375.

[11] G. Kerr, G. González-Parra, M. Sherman, *Applied Mathematics and Computation,* 420 (2022) 126914.

[12] G. Kerr, G. González-Parra, *Mathematics and Computers in Simulation,* 197 (2022) 308–326.