Malaysian Journal of Mathematical Sciences 17(3): 413–424 (2023)
https://doi.org/10.47836/mjms.17.3.10

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

Direct Multistep Method for Solving Retarded and Neutral Delay Differential Equation with Boundary and Initial Value Problems

Jaaffar, N. T.*[1], Ismail, N. I. N.[2], Majid, Z. A.[2, 3], and Senu, N.[2, 3]

[1]*Faculty of Business & Communication, Universiti Malaysia Perlis, 01000 Kangar, Perlis, Malaysia*
[2]*Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
[3]*Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

E-mail: tasnemjaaffar@gmail.com
**Corresponding author*

Received: 8 May 2023
Accepted: 3 August 2023

Abstract

The boundary and initial conditions that are related to the retarded and neutral delay differential equations, respectively, will be resolved in this work by using the previous direct multistep method. This method solves retarded and neutral delay differential equations directly by implementing the proposed method without converting it to a first-order system. For boundary value problems, the shooting strategy incorporated with the Newton method is utilized to predict the guessing value. The initial value problem for neutral delay differential equations on the other hand is resolved directly with special attention to the differential part of the problem. Several numerical examples are investigated to observe the capability of the developed strategies and methods for solving retarded delay differential equations with boundary value problems and neutral delay differential equations with initial value problems.

Keywords: boundary value problem; constant delay; retarded delay differential equation; direct method; initial value problem; neutral delay differential equation; shooting method.

1 Introduction

A delay differential equation (DDE) is under the ordinary differential equation (ODE) category with time lags where the function is dependent on its past values and current value. DDEs are applied as mathematical models in various real-life phenomena, including chemical reactions, population dynamics, and neural networks. The general form for second-order retarded delay differential equation (RDDE) is as shown below:

$$\begin{aligned} y''(t) &= f\Big(t, y(t), y'(t), y(t-\tau), y'(t-\tau)\Big), \\ y(t) &= \eta(t), \qquad y'(t) = \eta'(t), \qquad [-\tau, a], \end{aligned} \tag{1}$$

with the boundary conditions given as shown below,

$$y(a) = \alpha, \qquad y(b) = \beta. \tag{2}$$

Neutral delay differential equations (NDDEs) are a special type of DDEs that include not only past values of the function but also past values of its derivative. NDDEs arise in many applications, including control theory and engineering systems with time delays. They are more complex than standard RDDEs and require specialized techniques for their analysis and solution. The equation for second-order NDDE is defined by,

$$\begin{aligned} y''(t) &= f\Big(t, y(t), y'(t), y(t-\tau), y'(t-\tau), y''(t-\tau)\Big), \\ y(t) &= \phi(t), \qquad y'(t) = \phi'(t), \qquad [-\tau, a], \end{aligned} \tag{3}$$

with the initial value problem,

$$y(a) = \gamma, \qquad y'(a) = \lambda, \tag{4}$$

τ is the time lag which is a positive constant. Both the functions, f and inital functions, in equations (1) and (3), need to be continuous along the interval $t \in [a, b]$, where $a, b, \lambda \in \mathbb{R}$. The initial function is given as $\eta(t)$, the solution for the delay term is $y(t-\tau)$ while $y'(t-\tau)$ and $y''(t-\tau)$ are the delay derivatives which need to be considered continuous in solving NDDE.

The existence and uniqueness of the solutions to the mathematical problems under consideration are being assumed. The second-order RDDEs and NDDEs in (1) and (3) respectively can also be defined from the general types of DDEs and NDDEs [9]. Consider,

$$\eta \in C^{r-2}[a,b], \qquad r > 2, \qquad f : [a,b] \times C^1[a,b] \times C^1[a,b] \times C[a,b] \times C[a,b] \to \Re.$$

For RDDE,
H_1: For any $y \in C^1[-\tau, b]$ the mapping $t \to f(t, y, z, u, w)$ is a continuous on $[a, b]$.
H_2: The following Lipchitz condition holds:

$$\begin{aligned} &\Big\| f(t, y_1, z_1, u_1, w_1) - f(t, y_2, z_2, u_2, w_2) \Big\| \leq \\ &\qquad L\Big(\|y_1 - y_2\|_{[-\tau,t]} + \|z_1 - z_2\|_{[-\tau,t-\tau]} + \|u_1 - u_2\|_{[-\tau,t-\tau]} + \|w_1 - w_2\|_{[-\tau,t]} \Big), \end{aligned}$$

with: $L \geq 0$, $\tau > 0$ for any $t \in [a, b]$ where $y_1, y_2, z_1, z_2 \in C^1[a, b]$ and $u_1, u_2, w_1, w_2 \in C[-\tau, b]$.

As for NDDE,
H_1: For any $y \in C^1[-\tau, b]$ the mapping $t \to f\big(t, y, z, u, v, w\big)$ is a continuous on $[a, b]$.
H_2: The following Lipchitz condition holds:

$$\Big\|f\big(t, y_1, z_1, u_1, v_1, w_1\big) - f\big(t, y_2, z_2, u_2, v_2, w_2\big)\Big\| \leq$$
$$L\Big(\|y_1 - y_2\|_{[-\tau,t]} + \|z_1 - z_2\|_{[-\tau,t-\tau]} + \|u_1 - u_2\|_{[-\tau,t-\tau]} + \|v_1 - v_2\|_{[-\tau,t-\tau]} + \|w_1 - w_2\|_{[-\tau,t]}\Big),$$

with: $L \geq 0$, $\tau > 0$ for any $t \in [a, b]$ where $y_1, y_2, z_1, z_2 \in C^1[a, b]$ and $u_1, u_2, v_1, v_2, w_1, w_2 \in C[-\tau, b]$.

Under the condition of H_1 and H_2, equations (1) and (3) have a unique solution of $y \in C^2[a, b] \cap C^1[-\tau, b]$ [10].

In both RDDEs and NDDEs, the presence of delays introduces nonlocal effects that can lead to complex dynamics such as oscillations, stability switches, and even chaos. Therefore, the study of these types of differential equations has important implications for understanding real-world phenomena and developing effective control strategies. The RDDE with BVP and NDDE with IVP has been studied extensively since the 1970s but not so many studies in the last few years.

Solving RDDE with boundary conditions by numerical method had proposed by several researchers such as [16] where they solved using the shooting method of bisection method while Reddien and Travis [18] provided two projection schemes of the collocation and Galerkin methods by approximating the functions using the cubic spline polynomials. Chocholaty and Slahor [7] used the Runge-Kutta method after linearising the nonlinear RDDE and changed into a Cauchy problem while Agarwal and Chow [1] solved using the finite difference method. Bakke and Jackiewicz [3] solved BVP for RDDE by combining the central difference method, Lagrange interpolation, and Richardson extrapolation. Other than that, Qu and Agarwal [17] solved the two-point BVP of RDDE by using a subdivision approach in the approximation of the basis function for the collocation method. Research is made by Bartoszewski [4] by applying the forward finite difference method and composite Simpson method after reducing the second-order BVP to the system of the first order and changing into fixed-point problems. Recent research by Li and Zhang [13] had solved linear and nonlinear RDDE problems using the extended generalized Stormer-Cowell method.

Other than RDDE, a series of articles considering NDDE have been published by [11, 12] for both one-step and multi-step methods in $PECE$ mode. Semi-analytical techniques including the reproducing kernel Hilbert space method (RKHSM), variational iteration method (VIM), homotopy analysis method (HAM), and homotopy perturbation method (HPM) have all been proposed by a few authors, [5, 6] and [19]. The two-step Runge-Kutta [23] approach and the one-leg θ-method [24] method were compared with the four semi-analytical methods. The findings confirmed the effectiveness of analytical approaches for resolving NDDE. Analytical techniques such as Lie group analysis to find the exact solutions of second order NDDE has been studied by [15] where they completed the classification of second order non-linear NDDE to solvable Lie algebra. Finally, the convergence of solutions for NDDE is proven by [22] where an example is discussed to illustrate the efficiency of the result.

In this study, we are interested in solving second-order RDDE and NDDE with BVP and IVP, respectively, by using the multistep method derived in [14].

2 Methodology

In this section, the direct Adams Moulton order 4 method (DAMM4) in [14] will be derived. The point y_{i+1} at t_{i+1} is obtained by integrating once and twice for both equations (1) and (3) over the interval $[t_i, t_{i+1}]$.

Integrate once:

$$\int_{t_i}^{t_{i+1}} y''(t)\,dt = \int_{t_i}^{t_{i+1}} f\Big(t, y(t), y'(t)\Big)\,dt,$$
$$y'(t_{i+1}) = y'(t_i) + \int_{t_i}^{t_{i+1}} f\Big(t, y(t), y'(t)\Big).$$

Integrate twice:

$$\int_{t_i}^{t_{i+1}}\int_{t_i}^{t} y''(t)\,dt\,dt = \int_{t_i}^{t_{i+1}}\int_{t_i}^{t} f\Big(t, y(t), y'(t)\Big)\,dt\,dt,$$
$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \int_{t_i}^{t_{i+1}} (t_{i+1}-t) f\Big(t, y(t), y'(t)\Big)\,dt.$$

By considering $t = t_{i+1}+sh$, replacing $dt = hds$ and approximating the function $f(t, y, y')$ by using Lagrange interpolating polynomial, the predictor and corrector formula for order 4 are obtained. The interpolation points used in the corrector formula are as follows:

$$\Big\{(t_{i-2}, y_{i-2}), (t_{i-1}, y_{i-1}), (t_i, y_i), (t_{i+1}, y_{i+1})\Big\}.$$

Hence, the predictors of order 3 and correctors of the order 4 formula are obtained as below:

Predictor:

$$y'_{i+1} = y'_i + \frac{h}{12}\Big(23f_i - 16f_{i-1} + 5f_{i-2}\Big), \tag{5}$$
$$y_{i+1} = y_i + hy'_i + \frac{h^2}{24}\Big(19f_i - 10f_{i-1} + 3f_{i-2}\Big). \tag{6}$$

Corrector:

$$y'_{i+1} = y'_i + \frac{h}{24}\Big(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}\Big), \tag{7}$$
$$y_{i+1} = y_i + hy'_i + \frac{h^2}{360}\Big(38f_{i+1} + 171f_i - 36f_{i-1} + 7f_{i-2}\Big). \tag{8}$$

Thus, the formulas above are used to solve second-order RDDE and NDDE in (1) and (3) respectively. This method is implemented using C programming to generate the approximate solution of y_{i+1}.

To improve the accuracy of the approximations, the results of y'_{i+1} as well as y_{i+1} are evaluated via the predictor-corrector technique, which is PECE mode. P is designated as the predictor formula, followed by E to evaluate the function f, C to be the corrector formula, and finally, E again to evaluate the function f to be utilized for the subsequent iteration.

The BVP will be converted into two initial value problems (IVPs) in the manners listed below in order to be handled by the shooting strategy, where the boundary conditions (2) will be changed to the initial condition by guessing the value of $y'(a)$,

$$y(a) = \alpha, \qquad y'(a) = p. \tag{9}$$

Assume that (1) is written as having a solution that depends upon t and p, as illustrated below:

$$y''(t,p) = f\Big(t, y(t,p), y(t-\tau, p-\tau), y'(t-\tau, p-\tau)\Big), \tag{10}$$

and initial values expressed as,

$$y(a,p) = \alpha, \qquad y'(a,p) = p_1.$$

The variables $p = p_k$ are chosen to fulfil the following requirement:

$$\lim_{k\to\infty} y(b, p_k) - \beta = 0.$$

The initial guessing value, p_1 can be generated by using,

$$p_1 = \frac{\beta - \alpha}{b - a}.$$

In order to differentiate (10), the partial derivative towards the variable p is applied. Then, let $z(t,p) = \dfrac{\delta y}{\delta p}(t,p)$, hence will get as the following,

$$z''(t,p) = z(t,p).\frac{\delta f}{\delta y}(t,y,y') + z'(t,p).\frac{\delta f}{\delta y'}(t,y,y'), \tag{11}$$

and initial values,

$$z(a,p) = 0 \quad \text{and} \quad z'(a,p) = 1.$$

Each IVP, (10) and (11) are then calculated for each iteration to be able to obtain the subsequent guessing value, p_k by applying Newton's method. Newton's method formula is,

$$p_k = p_{k-1} - \frac{y(b, p_{k-1}) - \beta}{z(b, p_{k-1})}.$$

The IVPs (10) and (11) were solved simultaneously by using DAMM4. This technique is continued until the condition of $|y(b, p_{k-1}) - \beta| \leq$ tolerance is met. The proposed method's methodology was designed via C programming.

Considering the NDDE with IVP, both delay conditions, $y(t-\tau)$ and $y''(t-\tau)$ will be approximated by applying different approaches directly. Since the proposed method is a multistep method, therefore a one-step method is required to find the initial solutions. A direct Euler method as shown below,

$$y'_{i+1} = y'_i + h\Big[f\Big(t, y(t), y(t-\tau), y''(t-\tau)\Big)\Big], \tag{12}$$

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2}\Big[f\Big(t, y(t), y(t-\tau), y''(t-\tau)\Big)\Big], \tag{13}$$

is applied to estimate the first two solutions before applying the proposed method. As the delay term and its derivative also need to be considered, the first and second-order divided difference formulas are utilized in estimating both $y'(t-\tau)$ and $y''(t-\tau)$ respectively. Approximation values are then generated using the programming language C with a constant step size technique.

2.1 Algorithm for RDDE with BVP

The algorithm for the DAMM4 to solve second-order RDDE with boundary conditions is as follows:

Step 1: Set $h = \dfrac{b-a}{N}$, $p_1 = \dfrac{(\beta-\alpha)}{(b-a)}$, $y_0 = \alpha$, $y_0' = p_1$, $z_0 = 0$, $z_0' = 1$. p_1 is the guessing value for y_0'. The tolerence, TOL is set to be $TOL = 10^{-5}$.

Step 2: Set the initial values $t_0, (t_0-\tau), (y_0-\tau), (y_0'-\tau), (z_0-\tau), (z_0'-\tau)$, $f\Big(t_0, y_0, y(t_0-\tau), y'(t_0-\tau)\Big)$ and $f\Big(t_0, z_0, z(t_0-\tau), z'(t_0-\tau)\Big)$.

Step 3: Locate the position of $(t_i-\tau)$. If $(t_i-\tau) \leq t_0$ then use the initial function to approximate the delay term $y(t_i-\tau)$ or if $(t_i-\tau)$ falls in the previous points then take back the previous approximate solution. $z(t_i-\tau)$ and $z'(t_i-\tau)$ are always equal to zero because $z = \dfrac{\delta y}{\delta p} = 0$ and $z' = \dfrac{\delta y'}{\delta p} = 0$.

Step 4: All of the initial solutions are evaluated by using the direct modified Euler's method and the direct Euler's method that act as the corrector and predictor respectively.

Step 5: For $n = 3, 4, \ldots$ the DAMM4 is applied.

Step 6: Calculate the predictor and corrector values of the next iteration of y_{n+1} and z_{n+1} using the same procedures from **Step 3** to **Step 5** by using (5) and (7).

Step 7: Check whether $(y_N - \beta) \leq TOL$, if so, calculate the maximum absolute errors else set the new p_k by using Newton's method.

Step 8: Procedure is complete.

2.2 Algorithm for NDDE with IVP

The algorithm for the DAMM to solve second-order NDDE with initial condition (DAMM4) is as follows:

Step 1: Set all values for $h = \dfrac{b-a}{N}$, y_0, $y_t = \eta(t)$, $y_t' = \eta'(t)$.

Step 2: Set the initial values $t_0, (t_0-\tau), (y_0-\tau), (y_0''-\tau)$, $f\Big(t_0, y_0, y(t_0-\tau), y''(t_0-\tau)\Big)$.

Step 3: Locate the position of $(t_i-\tau)$. If $(t_i-\tau) \leq t_0$ then use the initial function to approximate the delay terms or if $(t_i-\tau) \geq t_0$ then the additional method derived is applied.

Step 4: Direct Euler's method is applied to compute all the initial solutions.

Step 5: For $n = 3, 4, \ldots$ the DAMM4 is applied.

Step 6: The divided difference formulae derived are applied to find the delay derivative.

Step 7: The maximum absolute errors, average errors, overall steps taken, and total function calls evaluated are calculated computationally.

Step 8: Procedure is complete.

3 Results

There are five test problems that are used in this study as shown below. The approximate solutions obtained are compared with the solutions from the method of previous papers to examine the efficiency of our method.

Example 3.1. *RDDE:*

$$\begin{aligned} &y''(t) = -\frac{1}{2} + \frac{1}{2}y(t-\pi), && 0 \le t \le \pi, \\ &\eta(t) = 1 - \sin(t), && -\pi \le t \le 0, \\ &y(0) = 1, \quad y(\pi) = 1. && \end{aligned}$$

Exact solution: $y(t) = 1 - \sin(t)$.

Example 3.2. *RDDE:*

$$\begin{aligned} &y''(t) = y(t-\pi), && 0 \le t \le \pi, \\ &\eta(t) = \sin(t), && -\pi \le t \le 0, \\ &y(0) = 0, \quad y(1) = 0. && \end{aligned}$$

Exact solution: $y(t) = \sin(t)$.

Example 3.3. *RDDE:*

$$\begin{aligned} &y''(t) = -\frac{1}{16}\sin\big(y(t)\big) - (t+1)y(t-1) + t, && 0 \le t \le 2, \\ &y(t) = t - \frac{1}{2}, && t \le 0, \\ &y(t) = -\frac{1}{2}, && t \ge 2. \end{aligned}$$

No exact solution.

Example 3.4. *NDDE:*

$$\begin{aligned} &y''(t) = -y'(t) + y'(t-1) - 2y''(t-1), && 0 \le t \le 1, \\ &y(t) = -t, && t \le 0, \\ &y(0) = 0, && t \le 0, \\ &y'(0) = -1, && t \le 0. \end{aligned}$$

Exact solution: $y(t) = -2 + t + 2e^t$.

Example 3.5. *NDDE:*

$$\begin{aligned} &y''(t) = y'(t) + y''(t-1), && 0 \le t \le 1, \\ &y(t) = 1, && t \le 0, \\ &y(0) = 1, && t \le 0, \\ &y'(0) = 0, && t \le 0. \end{aligned}$$

Exact solution: $y(t) = e^t$.

Tables 1-6 use the following abbreviations:

h	:	Step size chosen.
MAXE	:	Maximum errors obtained.
AVE	:	Average errors obtained.
FCN	:	Function calls evaluated.
TS	:	Total steps taken.
ITN	:	Total iteration of guess.
p_{last}	:	Last guessing p_k at last iteration.
DAMM4	:	Direct Adams Moulton method for solving second-order RDDE and NDDE.
CR	:	The numerical integration technique with Taylor series in [21].
AR	:	The parameter fitted scheme in [2].
NS	:	The shooting technique using Euler's method in [16].
CRY	:	The finite differences method in [8].
2PFBM4	:	Two point fully multistep block method in [20].
2PDBM4	:	Two point diagonally multistep block method in [10].

Table 1: The results for DAMM4 when solving Example 3.1.

h	$\frac{\pi}{30}$	$\frac{\pi}{300}$	$\frac{\pi}{3000}$
MAXE	4.69E-06	6.18E-10	5.92E-14
AVE	2.77E-06	3.91E-10	3.72E-14
ITN	2	2	2
p_{last}	-0.999995	-1.000000	-1.000000

Table 2: The results for DAMM4 when solving Example 3.2.

h	$\frac{\pi}{30}$	$\frac{\pi}{300}$	$\frac{\pi}{3000}$
MAXE	1.97E-06	3.04E-10	1.14E-13
AVE	1.21E-06	1.89E-10	7.69E-14
ITN	2	2	2
p_{last}	0.999984	1.000000	1.000000

Table 3: The comparison MAXE for DAMM4, NS and CRY when solving Example 3.3.

		h		
$y_{approximate}$	Method	0.25	0.125	$\frac{1}{128}$=0.0078125
y(0.5)	DAMM4	1.51E-02	4.41E-03	4.31E-04
	NS	2.22E-01	1.24E-01	8.76E-03
	CRY	1.87E-02	4.73E-03	7.43E-05
y(1.0)	DAMM4	2.88E-02	8.74E-03	8.62E-04
	NS	3.98E-01	2.29E-01	1.65E-02
	CRY	4.03E-02	1.02E-02	1.60E-04
y(1.5)	DAMM4	2.17E-02	6.09E-03	3.10E-04
	NS	4.38E-01	2.44E-01	1.72E-02
	CRY	2.49E-02	6.29E-03	9.87E-05

Table 4: The results for DAMM4 when solving Example 3.3.

	h		
$y_{approximate}$	0.25	0.125	$\frac{1}{128}$=0.0078125
y(0.5)	1.513E-02	4.41E-03	4.31E-04
y(1.0)	2.88E-02	8.74E-03	8.62E-04
y(1.5)	2.17E-02	6.09E-03	3.10E-04
ITN	5	5	5
p_{last}	-2.492707	-2.516615	-2.525147

Table 5: The results for DAMM4, ABM4, and RK4 when solving Example 3.4 at $h = 0.1, 0.01$ and 0.001.

h	MTD	FCN	TS	MAXE	AVERE
0.1	DAMM4	10	10	9.7099e-02	5.0546e-02
	2PFBM4	10	6	1.8960e-01	1.0706e-01
	2PDBM4	6	6	1.8981e-01	1.0710e-01
0.01	DAMM4	100	100	9.9138e-03	1.2107e-03
	2PFBM4	100	51	1.9894e-02	3.1406e-03
	2PDBM4	51	51	1.9896e-02	3.2659e-03
0.001	DAMM4	1000	1000	9.9331e-04	1.9095e-05
	2PFBM4	1000	501	1.9989e-03	5.1136e-05
	2PDBM4	501	501	1.9990e-03	6.1222e-05

Table 6: The results for DAMM4, ABM4, and RK4 when solving Example 3.5 at $h = 0.1, 0.01$ and 0.001.

h	MTD	FCN	TS	MAXE	AVERE
0.1	DAMM4	10	10	7.0630e-02	1.3243e-02
	2PFBM4	10	6	7.0630e-02	1.4417e-02
	2PDBM4	6	6	7.0630e-02	1.6841e-02
0.01	DAMM4	100	100	9.6783e-03	3.4873e-04
	2PFBM4	100	51	9.6783e-03	1.6976e-04
	2PDBM4	51	51	9.6783e-03	2.9857e-04
0.001	DAMM4	1000	1000	9.9675e-04	2.3733e-05
	2PFBM4	1000	501	9.9675e-04	2.2302e-04
	2PDBM4	501	501	9.9675e-04	2.2300e-04

4 Discussion

The abilities of the DAMM4 to solve second-order linear RDDE with boundary conditions are demonstrated in Table 1 and Table 2 for Example 3.1 and Example 3.2 respectively. The maximum errors, MAXE are reduced as the step size, h becomes smaller to portray the accuracy of DAMM4. The number of iteration needed to guess the initial condition, ITN are only two for linear RDDE in Example 3.1 and Example 3.2, hence this will help in reducing the total computational cost. The most accurate guessing of the initial value indicated by p_{last} shows that this value converge to a single value as the step size decreases for both Example 3.1 and Example 3.2. This conveys that the strategy of the shooting method along with Newton's method able to give a converge solution for the missing initial condition.

The result of maximum error in Table 3 shows that DAMM4 is better than the previous study, NS and comparable with CRY when solving Example 3.3. Hence, this proves that DAMM4 is able to solve not just second-order linear but also nonlinear RDDE with boundary conditions problems. Based on Table 4, our approach needs five iterations, ITN to guess the initial value of p_k because Example 3.3 is nonlinear type RDDE compared to only two iterations needed for Example 3.1 and Example 3.2 which are linear type RDDE, eventually the computational cost to solve Example 3.3 will be higher than Example 3.1 and Example 3.2. The value of p_{last} that converge to a single value when the step size decreases has prove the ability and convergence of our mentioned staretgy to find the best initial condition.

Referring to Table 5 and Table 6 when solving NDDE, the proposed method has been seen to produce more accurate results than 2PFBM4 and 2PDBM4 even though both comparison methods are block methods. The function calls evaluated are as same as 2PFBM4 since 2PFBM4 is a fully multistep block method with an extra point calculated in its computation. The accuracy obtained managed to overpass 2PDBM4 even with slightly larger total steps taken. Overall, the proposed method is applicable in solving both RDDE and NDDE with constant delay.

5 Conclusions

In this article, DAMM4 has been implemented in solving second-order RDDE with BVP and NDDE with IVP directly without transforming the second order to first order system, hence this could reduce the computational cost. The strategy of shooting technique along with Newton's method has been applied to solve the boundary problems involved. This strategy has shown that the guessing initial value converge to a single value and the iterations needed are not more than five iterations, hence it could save the computational cost as well. The accuracy of DAMM4 to solve both RDDE and NDDE has been observed when the maximum errors are compared with the previous methods. Overall, the numerical results obtained have shown that DAMM4 is suitable and applicable to solve both RDDE and NDDE with BVP and IVP, respectively.

Acknowledgement The authors would like to express their gratitude to all anonymous reviewers for their valuable feedback and suggestions, which greatly contributed to the improvement of this paper.

Conflicts of Interest The authors declare no conflict of interest.

References

[1] R. P. Agarwal & Y. M. Chow (1986). Finite difference methods for boundary value problems of differential equations with deviating arguments. *Computational & Mathematics with Applications*, *12*(11, Part A), 1143–1153. https://doi.org/10.1016/0898-1221(86)90018-0.

[2] A. Andargie & Y. N. Reddy (2013). Parameter fitted scheme for singularly perturbed delay differential equations. *International Journal of Applied Science and Engineering*, *11*(4), 361–373. https://doi.org/10.6703/IJASE.2013.11(4).361.

[3] V. L. Bakke & Z. Jackiewicz (1989). The numerical solution of boundary value problems for differential equations with state dependent deviating arguments. *Aplikace Matematiky*, *34*(1), 1–17. http://eudml.org/doc/15560.

[4] Z. Bartoszewski (2010). A new approach to numerical solution of fixed-point problems and its application to delay differential equations. *Applied Mathematics and Computation*, *215*(12), 4320–4331. https://doi.org/10.1016/j.amc.2009.12.058.

[5] J. Biazar & B. Ghanbari (2012). The homotopy perturbation method for solving neutral functional–differential equations with proportional delays. *Journal of King Saud University-Science*, *24*(1), 33–37. https://doi.org/10.1016/j.jksus.2010.07.026.

[6] X. Chen & L. Wang (2010). The variational iteration method for solving a neutral functional-differential equation with proportional delays. *Computers & Mathematics with Applications*, *59*(8), 2696–2702. https://doi.org/10.1016/j.camwa.2010.01.037.

[7] P. Chocholaty & L. Slahor (1979). A numerical method to boundary value problems for second order delay differential equations. *Numerische Mathematik*, *33*, 69–75. https://doi.org/10.1007/BF01396496.

[8] C. W. Cryer (1973). The numerical solution of boundary value problems for second order functional differential equations by finite differences. *Numerische Mathematik*, *20*, 288–299. https://doi.org/10.1007/BF01407371.

[9] R. D. Driver (1977). *Ordinary and delay differential equations* volume 20. Springer-Verlag New York Inc, New York.

[10] N. Jaaffar, Z. Majid & N. Senu (2021). Numerical computation of third order delay differential equations by using direct multistep method. *Malaysian Journal of Mathematical Sciences*, *15*(3), 369–385.

[11] Z. Jackiewicz (1982). Adams methods for neutral functional differential equations. *Numerische Mathematik*, *39*, 221–230. https://doi.org/10.1007/BF01408695.

[12] Z. Jackiewicz & E. Lo (2006). Numerical solution of neutral functional differential equations by Adams methods in divided difference form. *Journal of Computational and Applied Mathematics*, *189*(1-2), 592–605. https://doi.org/10.1016/j.cam.2005.02.016.

[13] C. Li & C. Zhang (2017). The extended generalized Störmer–Cowell methods for second-order delay boundary value problems. *Applied Mathematics and Computation*, *294*, 87–95. http://dx.doi.org/10.1016/j.amc.2016.09.006.

[14] Z. A. Majid, P. S. Phang & M. Suleiman (2011). Solving directly two point non linear boundary value problems using direct Adams Moulton method. *Journal of Mathematics and Statistics*, *7*(2), 124–128. https://doi.org/10.3844/jmssp.2011.124.128.

[15] L. Muhsen & N. Maan (2016). Lie group analysis of second-order non-linear neutral delay differential equations. *Malaysian Journal of Mathematical Sciences*, *10*(S), 117–129.

[16] K. D. Nevers & K. Schmitt (1971). An application of the shooting method to boundary value problems for second order delay equations. *Journal of Mathematical Analysis and Applications*, *36*, 588–597.

[17] R. Qu & R. P. Agarwal (1998). A subdivision approach to the contruction of approximate solutions of boundary value problems with deviating arguments. *Computational & Mathematics with Applications*, *35*(11), 121–135. https://doi.org/10.1016/S0898-1221(98)00089-3.

[18] G. W. Reddien & C. C. Travis (1974). Approximation methods for boundary value problems of differential equations with functional arguments. *Journal of Mathematical Analysis and Applications*, *46*(1), 62–74. https://doi.org/10.1016/0022-247X(74)90281-9.

[19] M. G. Sakar (2017). Numerical solution of neutral functional-differential equations with proportional delays. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, *7*(2), 186–194. https://doi.org/10.11121/ijocta.01.2017.00360.

[20] H. Y. Seong (2016). *Multistep Blocks Methods For Solving Higher Order Delay Differential Equations*. PhD thesis, Universiti Putra Malaysia, Malaysia,.

[21] C. L. Sirisha & Y. N. Reddy (2017). Numerical integration of singularly perturbed delay differential equations using exponential integrating factor. *Mathematical Communications*, *22*(2), 251–264.

[22] C. Tunç (2015). Convergence of solutions of nonlinear neutral differential equations with multiple delays. *Boletín de la Sociedad Matemática Mexicana*, *21*(2), 219–231. https://doi.org/10.1007/s40590-014-0050-6.

[23] W. Wang, Y. Zhang & S. Li (2009). Stability of continuous Runge–Kutta-type methods for nonlinear neutral delay-differential equations. *Applied Mathematical Modelling*, *33*(8), 3319–3329. https://doi.org/10.1016/j.apm.2008.10.038.

[24] W. S. Wang & S. F. Li (2007). On the one-leg θ-methods for solving nonlinear neutral functional differential equations. *Applied Mathematics and Computation*, *193*(1), 285–301. https://doi.org/10.1016/j.amc.2007.03.064.