FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 8 No. 6, December, 2024, pp 313 - 320
DOI: https://doi.org/10.33003/fjs-2024-0806-2865

LINEAR-THETA METHOD FOR THE DISCRETIZATION AND NUMERICAL SOLUTION OF FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS WITH MULTIPLE RETARDATIONS

***Dawodu, K. A. and Obarhua, F. O.**

Department of Mathematical Sciences, The Federal University of Technology, P. M. B. 704, Akure, Nigeria.

*Corresponding authors' email: kadawodu@futa.edu.ng

ABSTRACT

This study presents a special case of *proximal point algorithm* for solving linear programming problem (LPP). This method, also known as the *Alternating Direction Method of multipliers* (ADMM), was deployed because of its strong convergence properties of the method of multipliers, the decomposability property of dual ascent and the potential to solve large- scale structured optimization problems. The update formulas for the LPP were derived from the associated augmented Lagrangian with the primal and dual residuals also derived for the convergence of the algorithm. The Game theory was re-structured into a LPP amenable to the ADMM. Prisoner's Dilemma in Game theory was tested with the ADMM provided the matrix operator is invertible to guarantee its convergence. Other Numerical examples were also tested and it was discovered that the developed algorithm performs faster than the conventional simplex method.

Keywords: Multiple delay, θ-Method, Discretization

INTRODUCTION

Many scientific problems are described by the mathematical modeling of the form

$$\begin{cases} \frac{dy}{dt}(t) = & f(t, y(t)) \qquad t \in [t_0, T], \\ y(t_0) = & y_0. \end{cases} \tag{1}$$

The solutions are known under the global Lipschitz condition below

$$|f(t, s_1) - f(t, s_2)| < L|s_1 - s_2| \tag{2}$$

where $(t, s_1), (t, s_2) \in dom(f)$ such that with the Lipschitz constant $L > 0$, the problem (1) has a unique solution on the domain $dom(f)$. However, we cannot guarantee the analytical solutions of majority of such problems especially for Non-smooth differential equations of the delay type. It is then necessary to find suitable numerical algorithms to ascertain accurate approximated solutions to the problem. The numerical integration of (1) under condition (2) is the most applicable method in the numerical solution of the mathematical modeling of real-life problems.

The aim of this paper is to deploy the θ - method for the numerical integration of the variant of problem (1) with multiple delays(lags); represented in the general form below:

$$\begin{cases} y'(t) & = f(t, y(t), y^d(t)), \quad t_0 - \tau \le t \le T, \\ y(t) & = \phi(t), \qquad t_0 - \tau \le t \le t_0 \\ y(t_0) & = y_0 \end{cases} \tag{3}$$

where $y(t) \in R^n$, $\phi(t) \in R^n$ is a known and piece-wise continuous function, $y^d(t) = \big(y(t - \tau_1), y(t - \tau_2), \ldots, y(t - \tau_d)\big) \in R^{nd}$, $f: [t_0, T] \times R^{n(1+d)} \to R^n$ and the delays $\tau_1, \tau_2, \ldots, \tau_d$ are given positive constants with $\tau = \max\{\tau_j\}_{j=1}^{d} \in R^n$, hence a system of n first order DDE. However, this research paper is limited to general in-homogeneous linear ODE with multiple delays such that the functional f in eqn. (3) is expressed in the form

$$f(t, y(t), y^d(t)) = p(t)y(t) + \sum_{j=1}^{d} \alpha^{(j)}(t)y(t - \tau_j) + g(t) \tag{4}$$

where the coefficients of the ODE are $n \times n$ dimensional variable coefficient matrices with $p(t) \in R^{n \times n}, g(t) = [g_1(t), g_2(t), \ldots, g_n(t)]^T \in R^n$ is a known continuous function representing the external excitation and $\alpha^{(j)}(t) \in R^{n \times n}$ for all $j = 1,2, \ldots, d$ and are continuous everywhere in the interval $[t_0, T]$. Also, the function f and the initial (pre-shaped) function $\phi(t) = [\phi_1(t), \phi_2(t), \ldots, \phi_n(t)]^T \in R^n$ satisfy the following conditions: For any constant L, the Lipschitz condition holds:

$$||f(t, y_1) - f(t, y_2)|| \le ||y_1 - y_2|| \tag{5}$$

For any $y \in C^1 \in [t_0 - \tau, T]$, the mapping below is continuous:

$$f: t \in [t_0 - \tau, T] \to y(t) \tag{6}$$

Under the conditions (5) and (6) above, the statement of problem, in eqns. (3) -(4), has a unique solution. This method can be applied to the work of (Adamu *et al.*, 2023) if redesigned into an optimal control model by introducing delay terms.

MATERIALS AND METHODS

A commonly used numerical scheme for the discretization of (3) is the forward backward Euler numerical scheme called the θ - method with $\theta \in [0,1]$, the splitting parameter, expressed in the form

$$y_{k+1} = y_k + \delta[(1 - \theta)f_k + \theta f_{k+1}], \tag{7}$$

where $y(t) \approx y(t_k) = y_k, f(t, y(t)), y^d(t) \approx f(t_k, y(t_k), y^d(t_k - m_j\delta)) = f_k$ for all $j = 1,2, \ldots, d$.

Analysis of the θ - Method

Let the linear difference operator defined on the method in eqn (7) be

$$[y(t); \delta] = C_0\delta y(t) + C_1\delta y'(t) + C_2\delta^2 y''(t) + \ldots + C_{p+1}\delta^{p+1}y^{p+1}(t) + T_{n+k} \tag{8}$$

$$C_0 = C_1 = C_2 = \ldots = C_{p+1} \qquad C_{p+2} \ne 0 \tag{9}$$

Definition 1*:* The coefficient C_{p+2} is called the error constant, the term $T_{n+k} = C_{p+2}y^{p+2}(t) + O(\delta^{p+2})$ the local truncation error and $p \ge 1$ the order of the method.

Equating eqn (8) to the Linear difference operator below yields;

$$[y(t); \delta] = \sum_{k=0}^{n} (\alpha_j y_{k+j} - \delta\beta_j f_{k+j}); \ \ for\ y_{k+j} = (y_k + j\delta) \tag{10}$$

$$0 = y_{k+1} - y_k - \delta[(1 - \delta)f_k + \theta f_{k+1}] \tag{11}$$

Taking the Taylor's series expansion of the derivatives and combining coeficients of like terms in δ^k gives $C_0 = C_1 = C_2 = 0\ for\ \theta = \frac{1}{2}$ such that the error constant is $\frac{-1}{12}$ and the order of the method is $p = 2$.

Definition 2: A Linear Multistep method of order $p \geq 1$ is said to be consistent if the sum of the coefficient of the first characteristics polynomial is equal to zero ($\sum_{k=0}^{n} a_j = 0$) The sum of the coefficients of the first characteristics equation of eqn (10) is

$$\sum_{k=0}^{n} \alpha_j = -1 + 1 = 0 \quad (12)$$

The first characteristics polynomial is $\rho(r) = -1 + r$ while $\rho'(1) = 1$. The second characteristics equation is $\sigma(r) = (1-\theta)r^0 + \theta r$ at $r = 1$ yields $\sigma(1) = 1$. Since the conditions are all satisfied, we then conclude that the method in eqn (10) is consistent for all $\theta \in [0,1]$.

Definition 3: A Linear multistep method of is Zero stable if the roots of first characteristics polynomial $\rho(r)$ has a modulus greater than 1 i.e $|r| \geq 1$ and absolutely stable within the region defined by $h(r) = \frac{\rho(r)}{\sigma(r)}$ defined by boundary locus method.

The first characteristics polynomial $\rho(r) = -1 + r$ has a root at $r = 1$ which satisfies the condition $|r| \geq 1$. Therefore, the method is zero stable. Furthermore, the piecewise linear interpolation of the functional in eqn. (3) over the entire interval $[t_0 - \tau, T]$ can be defined as

$$y^{(h)}(t) = \begin{cases} \phi(t) & \text{for } t \leq t_0 \\ \frac{(t_{k+1}-t)}{h} y_k + \frac{(t-t_k)}{h} y_{k+1} & \text{for } t > t_0 \end{cases} \quad (13)$$

where $t \in [t_k, t_{k+1}]$ for $k = 1,2,\ldots$. The degree of precision (order) of θ - method is one while of second order if θ is set to 0:5. However, if the Delay Differential Equation (DDE) in (14) below is considered,

$$\begin{cases} y'(t) &= \lambda y(t) + \mu y(t-\tau), \quad t_0 \leq t \leq T, \\ y(t) &= \phi(t), \quad t_0 - \tau \leq t \leq t_0, \end{cases} \quad (14)$$

where $\lambda, \mu \in \mathrm{C}$ are complex numbers and $\tau > 0$, then the solution y(t) of the linear DDE (14) tends to zero as t tends to infinity (that is, $y(t) \to 0$ as $t \to \infty$); provided $\phi(t)$ is continuous and P - Stable, that is $|\lambda| < Re(\mu)$ according to (Al-Mutib, 1984). Though the adaptation of the θ - method to DDE had been considered in many literature such as in (Calvo and Grande, 1988) and (Liu and Spijker, 1990), however this research intends to adapt the θ - method to DDE with multiple delay terms with the aim of deploying a Direct Method or any of the numerical Algorithms (such as Conjugate Gradient Method or preconditioned Conjugate Gradient Method or perhaps any of the Newton's methods) with high rate of convergence.

Discretization

For the purpose of illustration, the one-dimensional and generalized cases of the DDE in eqn. (3) were separately considered for $n = 1; m = 1$ and $n > 1; m > 1$ respectively. Recurrence relations were derived to help generate the matrix operators for each of the cases. The discretized matrix operators were well-posed sufficiently enough to ensure that the derived linear system were amenable to the direct method or perhaps any algorithms such as the CGM or Newton methods. To support the discretization procedure, we will be introducing some theorems as stated below:

Theorem 1: (Rationality theorem): *Given the real numbers $\tau_j, \tau_{j+1} > 0$ for $\tau_j < \tau_{j+1}$, then there exist a unique real number $\delta < 1 \in R$ such that the ratios of the numbers is a rational number Q.*

Theorem 2: *Given any interval $[a, b]$, there exist a steplength $\delta \in R^+$ such that each sub-interval is a constant multiple of the steplength.*. See proof of theorems 1 and 2 in (Dawodu, 2021).

Theorem 3*: A numerical method for DDEs is called* ***P-stable*** *if, for all efficient λ, μ satisfying the condition $|\lambda| < Re(\mu)$, the numerical solution y_k, of (14) at the mesh points $t_k - kh$, $k >$,0, satisfies $y_k \to 0$ as $\to \infty$ for every stepsize h such that $h = \tau/m$, where m is a positive integer (Z^k).* See proof of theorem in (Lu, 1991).

Theorem 4*: Given that $0 \leq \theta \leq 1$ then the numerical stability of the linear θ - method in (7) above is* ***GP-stable*** *if and only if $\frac{1}{2} \leq \theta \leq 1$.* See proof of theorem in (Barwell, 1975).

By consequence of theorems 1 and 2 above, the entire interval $[T - t_0 + \tau]$ is divided into $N + m \in \mathrm{Z}^+$ number of grid points with a steplength of δ such that the ratio $m_j = \frac{\tau_j}{\delta} \in \mathrm{R}^+$ are on the grid points for all $j = 1,2,\ldots,d$. The choice of the steplength is picked to ensure that non of the ratios m_j is off the grid. The steplength is made as finite as possible for better refinement or accuracy. It is imperative to note that if h_{max} is the maximum steplength for which $\tau_j = m_j h_{max}$ such that all the values $m_j \in \mathrm{Z}^+$ are all positive integers, then $10^{-k} h_{max}$ for $k = 1,2,\ldots$ are better steplengths for improved refinements. In the discretization of the delays terms, the positive delay constants are assumed to be monotone increasing where $\tau_j < \tau_{j+1} \in \mathrm{R}$ for all $j's$. The discrete representation of the continuous-time interval $[t_0, T]$ is given as $I_k = [t_k, t_{k+1}]$ by letting $t_k = t_0 + k\delta$ for $k = 0,1,\ldots,N$ with equal steplength $\delta = \frac{(T-t_0)}{N}$. The discretization of the delay terms $y(t - \tau_j)$ for all $j = 1,2,\ldots,d$ is then expressed as:

$$y^{(k-m_j)} = \begin{cases} \phi(t_k - m_j\delta) & : k - m_j < 0; k = 0,1,\ldots,(m_j - 1), t \in [t_0 - \tau, t_0] \quad \text{(Known)} \\ y(t_k - m_j\delta) & : k - m_j \geq 0; \quad \text{for} \quad k = m_j, (m_j - 1), \ldots, N, t \in [t_0, T] \text{ (Unknown)} \end{cases} \quad (15)$$

Case 1: One-Dimensional case with $n = 1$ and $d > 1$

In this case, the DDE is posed as;

$$\begin{cases} y'(t) = & p(t)y(t) + \sum_{j=1}^{d} \alpha_j(t) y(t - \tau_j) + g(t), \quad t_0 \leq t \leq T \\ y(t) = & \phi(t), \quad t_0 - \tau \leq t \leq t_0 \\ y(t_0) = & y_0, \end{cases} \quad (16)$$

where

$y(t) \in \mathrm{R}, \phi(t) \in \mathrm{R}, y^d(t) = \big(y(t-\tau_1), y(t-\tau_2), \ldots y(t-\tau_d)\big) \in \mathrm{R^d}, f: [t_0, T] \times \mathrm{R^{(1+d)}} \to \mathrm{R}$ and $\tau = \max\{\tau_j\}_{j=1}^{d} \in \mathrm{R}$.

Using the concept in eqn.(19), $f^{(k)}$ and $f^{(k+1)}$ are respectively given as

$$f_k = p(t_k)y_k + \sum_{j=1}^{d} \alpha_j(t_k) y_{k-m_j} + g(t_k) \quad (17)$$

$$f_{k+1} = p(t_{k+1})y_{k+1} +$$

$\sum_{j=1}^{d} \alpha_j(t_{k+1})y_{k+1-m_j} + g(t_{k+1})$ (18)

Substituting eqns. (20) and (21) into the θ-splitting scheme in eqn. (7) for the discretization of the continuous-time DDE yields the recurrence relation.

$$\beta(k,\theta)y_k + \gamma(k+1,\theta)y_{k+1} = a(\theta)\sum_{j=1}^{d} \alpha_j(t_k)y_{k-m_j} + g(t_k) + b(\theta) + \sum_{j=1}^{d} \alpha_j(t_{k+1})y_{k+1-m_j} + a(\theta)g(t_k) + b(\theta)g(t_{k+1}), k = 0,1,\dots N-1 \quad (19)$$

where $\beta(k,\theta) = \delta(\theta-1)p(t_k) - 1, \gamma(k+1,\theta) = 1 - \delta\theta(t_{k+1}), a(\theta) = \delta(1-\theta), b(\theta) = \delta\theta$ and $y_{k-m_j} = \phi(t_k - m_j\delta)$ fo all $j = 1,2,\dots,d$.

For $k = 1$

$$\beta(0,\theta)y_0 + \gamma(1,\theta)y_1 = a(\theta)\sum_{j=1}^{d} \alpha_j(t_0)y_{-m_j} + g(t_0) + b(\theta) + \sum_{j=1}^{d} \alpha_j(t_1)y_{1-m_j} + a(\theta)g(t_0) + b(\theta)g(t_1), \quad m_1 > 1$$

For $k = 2$

$$\beta(1,\theta)y_0 + \gamma(2,\theta)y_2 = a(\theta)\sum_{j=1}^{d} \alpha_j(t_1)y_{1-m_j} + b(\theta) + \sum_{j=1}^{d} \alpha_j(t_2)y_{2-m_j} + a(\theta)g(t_1) + b(\theta)g(t_2), \quad m_1 > 2$$

For $k = m_1$

$$-b(\theta)\alpha_1(t_{m_1+1})y_1 + \beta(m_1,\theta)y_{m_1} + \gamma(m_1+1,\theta)y_{m_1+1} = \alpha(\theta)\sum_{j=1}^{d} \alpha_j(t_{m_1})y_{m_1-m_j} + b(\theta)\sum_{j=1}^{d} \alpha_j(t_{m_1})y_{m_1-m_j} + a(\theta)g(t_{m_1}) + b(\theta)g(t_{m_1})$$

For $k = m_1 + 1$

$$-a(\theta)\alpha(t_{=m_1+1})y_1 - b(\theta)\alpha_1(t_{m_1+2})y_1 + \beta(m_1+1,\theta)y_{m_1+1} + \gamma(m_1+2,\theta)y_{m_1+2} = \alpha(\theta)\sum_{j=1}^{d} \alpha_j\big(t_{m_1+1}\big)y_{m_1+1-m_j} + b(\theta)\sum_{j=1}^{d} \alpha_j\big(t_{m_1+2}\big)y_{m_1+2-m_j} + a(\theta)g\big(t_{m_1+1}\big) + b(\theta)g(t_{m_1+2})$$

or $k = m_d$;

$$-a(\theta)\alpha_{d-1}(t_{m_d})ym_d - m_{d-1}\dots - a(\theta)\alpha_2(t_{m_d})ym_{m_d-m_2} - \alpha(\theta)\alpha_1(t_{m_d})y_{m_d-m_1} - b(\theta)\alpha_d(t_{m_d+1})y_1\dots - b(\theta)\alpha_2(t_{m_d+1})y_{m_d+1-m_2} - b(\theta)\alpha_1(t_{m_d+1})y_{m_d+1-m_1} + \beta(m_d,\theta)y_{m_d} + \gamma(m_d+1,\theta)y_{m_d+1} = \alpha(\theta)\alpha_d(t_{m_d})y_0 + \alpha(\theta)g(t_{m_d}) + b(\theta)g(t_{m_d+1})$$

For $k = m_d + 1$;

$$-a(\theta)\alpha_d(t_{m_d+1})y_1\dots - a(\theta)\alpha_2(t_{m_d+1})y_{m_d+1-m_2} - \alpha(\theta)\alpha_1(t_{m_d+1})y_{m_d+1-m_1} - b(\theta)\alpha_d(t_{m_d+2})y_2\dots - b(\theta)\alpha_2(t_{m_d+2})y_{m_d+2-m_2} - b(\theta)\alpha_1(t_{m_d+2})y_{m_d+2-m_1} + \beta(m_d+1,\theta)y_{m_d+1} + \gamma(m_d+2,\theta)y_{m_d+2} = \alpha(\theta)\alpha_d(t_{m_d}) + b(\theta)g(t_{m_d+2})$$

For $k = N - 1$;

$$-a(\theta)\alpha_d(t_{N-1})y_{N-1-m_d}\dots - a(\theta)\alpha_2(t_{N-1})y_{N-1-m_2} - \alpha(\theta)\alpha_1(t_{N-1})y_{N-1-m_1} - b(\theta)\alpha_d(t_N)y_{N-m_d}\dots - b(\theta)\alpha_2(t_N)y_{N-m_2} - b(\theta)\alpha_1(t_N)y_{N-m_2} - b(\theta)\alpha_1(t_N)y_{N-m_1} + \beta(N-1,\theta)y_{N-1} + \gamma(N,\theta)y_N = \alpha(\theta)g(t_{N-1}) + b(\theta)g(t_N),$$

where,

$$f_k = p(t_k)y_k + \sum_{j=1}^{d} \alpha_j(t_k)y_{k-m_j} + g(t_k) \quad (20)$$

$$f_{k+1} = p(t_{k+1})y_{k+1} + \sum_{j=1}^{d} \alpha_j(t_{k+1})y_{k+1-m_j} + g(t_{k+1}) \quad (21)$$

Substituting eqns. (20) and (21) into the θ-splitting scheme in eqn. (7) for the discretization of the continuous-time DDE yields the recurrence relation

$$\beta(k,\theta)y_k + \gamma(k+1,\theta)y_{k+1} = a(\theta)\sum_{j=1}^{d} \alpha_j(t_k)y_{k-m_j} + g(t_k) + b(\theta) + \sum_{j=1}^{d} \alpha_j(t_{k+1})y_{k+1-m_j} + a(\theta)g(t_k) + b(\theta)g(t_{k+1}), k = 0,1,\dots N-1 \quad (22)$$

where $\beta(k,\theta) = \delta(\theta-1)p(t_k) - 1, \gamma(k+1,\theta) = 1 - \delta\theta(t_{k+1}), a(\theta) = \delta(1-\theta), b(\theta) = \delta\theta$ and $y_{k-m_j} = \phi(t_k - m_j\delta)$ for all $j = 1,2,\dots,d$.

For $k = 1$

$$\beta(0,\theta)y_0 + \gamma(1,\theta)y_1 = a(\theta)\sum_{j=1}^{d} \alpha_j(t_0)y_{-m_j} + g(t_0) + b(\theta) + \sum_{j=1}^{d} \alpha_j(t_1)y_{1-m_j} + a(\theta)g(t_0) + b(\theta)g(t_1), \quad m_1 > 1$$

For $k = 2$

$$\beta(1,\theta)y_0 + \gamma(2,\theta)y_2 = a(\theta)\sum_{j=1}^{d} \alpha_j(t_1)y_{1-m_j} + b(\theta) + \sum_{j=1}^{d} \alpha_j(t_2)y_{2-m_j} + a(\theta)g(t_1) + b(\theta)g(t_2), \quad m_1 > 2$$

For $k = m_1$

$$-b(\theta)\alpha_1(t_{m_1+1})y_1 + \beta(m_1,\theta)y_{m_1} + \gamma(m_1+1,\theta)y_{m_1+1} = \alpha(\theta)\sum_{j=1}^{d} \alpha_j(t_{m_1})y_{m_1-m_j} + b(\theta)\sum_{j=1}^{d} \alpha_j(t_{m_1})y_{m_1-m_j} + a(\theta)g(t_{m_1}) + b(\theta)g(t_{m_1})$$

For $k = m_1 + 1$

$$-a(\theta)\alpha(t_{=m_1+1})y_1 - b(\theta)\alpha_1(t_{m_1+2})y_1 + \beta(m_1+1,\theta)y_{m_1+1} + \gamma(m_1+2,\theta)y_{m_1+2} = \alpha(\theta)\sum_{j=1}^{d} \alpha_j(t_{m_1+1})y_{m_1+1-m_j} + b(\theta)\sum_{j=1}^{d} \alpha_j(t_{m_1+2})y_{m_1+2-m_j} + a(\theta)g(t_{m_1+1}) + b(\theta)g(t_{m_1+2})$$

For $k = m_d$;

$$-a(\theta)\alpha_{d-1}(t_{m_d})ym_d - m_{d-1}\dots - a(\theta)\alpha_2(t_{m_d})ym_{m_d-m_2} - \alpha(\theta)\alpha_1(t_{m_d})y_{m_d-m_1} - b(\theta)\alpha_d(t_{m_d+1})y_1\dots - b(\theta)\alpha_2(t_{m_d+1})y_{m_d+1-m_2} - b(\theta)\alpha_1(t_{m_d+1})y_{m_d+1-m_1} + \beta(m_d,\theta)y_{m_d} + \gamma(m_d+1,\theta)y_{m_d+1} = \alpha(\theta)\alpha_d(t_{m_d})y_0 + \alpha(\theta)g(t_{m_d}) + b(\theta)g(t_{m_d+1})$$

For $k = m_d + 1$;

$$-a(\theta)\alpha_d(t_{m_d+1})y_1 \ldots -a(\theta)\alpha_2(t_{m_d+1})y_{m_d+1-m_2} - \alpha(\theta)\alpha_1(t_{m_d+1})y_{m_d+1-m_1}$$
$$-b(\theta)\alpha_d(t_{m_d+2})y_2 \ldots -b(\theta)\alpha_2(t_{m_d+2})y_{m_d+2-m_2} - b(\theta)\alpha_1(t_{m_d+2})y_{m_d+2-m_1}$$
$$+\beta(m_d+1,\theta)y_{m_d+1} + \gamma(m_d+2,\theta)y_{m_d+2} = \alpha(\theta)\alpha_d(t_{m_d}) + b(\theta)g(t_{m_d+2})$$

For $k = N - 1$;

$$-a(\theta)\alpha_d(t_{N-1})y_{N-1-m_d} \ldots -a(\theta)\alpha_2(t_{N-1})y_{N-1-m_2} - \alpha(\theta)\alpha_1(t_{N-1})y_{N-1-m_1}$$
$$-b(\theta)\alpha_d(t_N)y_{N-m_d} \ldots -b(\theta)\alpha_2(t_N)y_{N-m_2} - b(\theta)\alpha_1(t_N)y_{N-m_2} - b(\theta)\alpha_1(t_N)y_{N-m_1} + \beta(N-1,\theta)y_{N-1}$$
$$+\gamma(N,\theta)y_N = \alpha(\theta)g(t_{N-1}) + b(\theta)g(t_N),$$

which yields the following linear system of equations:

$$AY = C + F + (D \times E) = \bar{B}, \qquad (23)$$

where $\beta(k,\theta) = \beta(k), \gamma(k+1,\theta) = \gamma(k+1), a(\theta) = a$ and $b(\theta) = b$ and

$$A = \begin{pmatrix} \gamma(1) & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \beta(1) & \gamma(2) & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \vdots \\ -b\alpha_1(t_{m_1+1}) & \ddots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \vdots \\ -a\alpha_1(t_{m_1+1}) & -b\alpha_1(t_{m_1+2}) & \ddots & \beta(m_1+1) & \gamma(m_1+2) & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ -b\alpha_d(t_{m_d+1}) & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -a\alpha(t_{N-1}) & -b\alpha_d(t_N) & \cdots & \cdots & \cdots & \beta(N-1) & \gamma(N) \end{pmatrix}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ n_{N-1} \\ y_N \end{pmatrix}, \quad C = \begin{pmatrix} a\sum_d^{j=1} \alpha_j(t_0)y_{-m_j} + b\sum_{j=1}^d \alpha_j(t_1)y_{1-m_j} \\ a\sum_d^{j=1} \alpha_j(t_1)y_{1-m_j} + b\sum_{j=1}^d \alpha_j(t_2)y_{2-m_j} \\ \vdots \\ a\sum_d^{j=1} \alpha_j(t_{m_1})y_{m_1-m_j} + b\sum_{j=2}^d \alpha_j(t_{m_1+1})y_{m_1+1-m_j} \\ \vdots \\ a\alpha(t_{m_d})y_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$t = \begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ \vdots \\ t_N \end{pmatrix}, P = \begin{pmatrix} p(t_0) \\ p(t_1) \\ \vdots \\ \vdots \\ p(t_N) \end{pmatrix}, D = \begin{pmatrix} a & b & 0 & \cdots & \cdots & 0 \\ 0 & a & b & 0 & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & a & b \end{pmatrix}, E = \begin{pmatrix} g(t_0) \\ g(t_1) \\ \vdots \\ \vdots \\ g(t_N) \end{pmatrix}$$

$$\beta(k,\theta) = \begin{pmatrix} \beta(0) \\ \beta(1) \\ \vdots \\ \vdots \\ \beta(N-1) \\ \beta(N) \end{pmatrix} = \begin{pmatrix} \delta(\theta-1)p(t_0) - 1 \\ \delta(\theta-1)p(t_1) - 1 \\ \vdots \\ \vdots \\ \delta(\theta-1)p(t_{N-1}) - 1 \\ \delta(\theta-1)p(t_N) - 1 \end{pmatrix}, F = \begin{pmatrix} \beta(0)y^{(0)} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

and $$\gamma(k+1,\theta) = \begin{pmatrix} \gamma(1) \\ \gamma(2) \\ \vdots \\ \vdots \\ \gamma(N-1) \\ \gamma(sN) \end{pmatrix} = \begin{pmatrix} 1 - \delta\theta p(t_0) \\ 1 - \delta\theta p(t_1) \\ \vdots \\ \vdots \\ 1 - \delta\theta p(t_{N-1}) \\ 1 - \delta\theta p(t_N) \end{pmatrix}$$

The dimensions of the above vectors and matrices are given as $A \in R^{N\times N}, D \in R^{N\times(N+1)}, C \in R^N, E \in R^{(N+1)}, Y \in R^N, \gamma \in R^N, \beta \in R^{(N+1)}$ and $F \in R^N$.

Case 2: Generalized case wit n>1 and $d > 1$

The generalized case is formulated from the system of n-linear DDE expressed below.The generalized case is formulated from the system of n- linear DDE expressed below.

$$y'_1(t) = p_{11}y_1(t) + p_{12}y_2(t) + \cdots + p_{1n}y_n(t) + \alpha_{11}^{(1)}(t)y_1(t-\tau_1) + \cdot + \alpha_{1n}^{(1)}(t)y_n(t-\tau_1) + \alpha_{11}^{(n)}(t)y_1(t-\tau_d) + \cdot\, \alpha_{1n}^{(n)}(t)y_n(t-\tau_d) + g_1(t)$$
$$y'_2(t) = p_{21}y_1(t) + p_{22}y_2(t) + \cdots + p_{2n}y_n(t) + \alpha_{21}^{(1)}(t)y_1(t-\tau_1) + \cdot + \alpha_{2n}^{(1)}(t)y_n(t-\tau_1) + \alpha_{21}^{(n)}(t)y_1(t-\tau_d) + \cdot\, \alpha_{2n}^{(n)}(t)y_n(t-\tau_d) + g_2(t)$$

while the pre-shaped function and initial conditions expressed as:

$$\begin{aligned} y_1(t) &= \phi_1(t) \\ y_2(t) &= \phi_2(t) \\ \vdots \quad \vdots &= \quad \vdots \quad \vdots \\ y_n(t) &= \phi_n(t) \end{aligned}$$

The above systems of equations can be represented below in the matrix form as;

$$\begin{pmatrix} y'_1(t) \\ y'_2(t) \\ \vdots \\ \vdots \\ y'_n(t) \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & \cdots & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & \cdots & \cdots & p_{nn} \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{pmatrix} + \begin{pmatrix} \alpha_{11}^{(1)} & \alpha_{12}^{(1)} & \cdots & \cdots & \alpha_{1n}^{(1)} \\ \alpha_{21}^{(1)} & \alpha_{22}^{(1)} & \cdots & \cdots & \alpha_{2n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{n1}^{(1)} & \alpha_{n2}^{(1)} & \cdots & \cdots & \alpha_{nn}^{(1)} \end{pmatrix} \begin{pmatrix} y_1(t-\tau_1) \\ y_2(t-\tau_1) \\ \vdots \\ \vdots \\ y_n(t-\tau_1) \end{pmatrix}$$
$$+ \cdots + \begin{pmatrix} \alpha_{11}^{(d)} & \alpha_{12}^{(d)} & \cdots & \cdots & \alpha_{1n}^{(d)} \\ \alpha_{21}^{(d)} & \alpha_{22}^{(d)} & \cdots & \cdots & \alpha_{2n}^{(d)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{n1}^{(d)} & \alpha_{n2}^{(d)} & \cdots & \cdots & \alpha_{nn}^{(d)} \end{pmatrix} \begin{pmatrix} y_1(t-\tau_d) \\ y_2(t-\tau_d) \\ \vdots \\ \vdots \\ y_n(t-\tau_d) \end{pmatrix} + \begin{pmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ \vdots \\ g_n(t) \end{pmatrix}$$

with delay (pre-shaped) function

$$\begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_n(t) \end{pmatrix} = \begin{pmatrix} \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \vdots \\ \phi_n(t) \end{pmatrix}$$

and initial conditions $(y_1(t_0), y_2(t_0), \cdots, y_n(t_0)) = (y_{10}, y_{20}, \cdots, y_{n0})$ as posed in eqns. (20) - (22) above. The discretization of the above equations yields similar matrix representation as (26) with the formulated discrete block-matrices have each entry as a matrix (column). The dimensions of the block-matrices are as follows: $A \in R^{nN\times nN}, D \in R^{nN\times n(N+1)}, C \in R^{nN}, E \in R^{n(N+1)}, Y \in R^{nN}, \gamma \in R^{nN}, \beta \in R^{n(N+1)}$ and $F \in R^{nN}$ The derivation is as shown in the case 1 above. The derived linear equation is made amenable to the MCGM where the matrix operator A is invertible (i.e A^{-1} exist).

Implementation of Numerical Methods

The discretization of the delay (system of) differential equations could also yield rectangular system of equations on rear cases. And suppose $A \in R^{mN\times nN}$ is rectangular, the pseudo-inverse ($A \dagger \in R^{mN\times nN}$) was computed as;

$$(A^TA)Y = A^T\bar{B},$$
$$Y = A\dagger \hat{\bar{B}}, \tag{24}$$

where $A \dagger = (A^TA)^{-1}$ and $\hat{\bar{B}} = A^T\bar{B}$. If the square matrix $\hat{\bar{A}}$ in eqn. (24) is not positive definite, we then deploy the procedure below with the condition number, $(cond.\hat{\bar{A}}) = \frac{\lambda_{max}}{\lambda_{min}}$.

In the computation of the search direction of Linear function f(x) with iterative sequence $y_0 \to y_1 \to y_2 \cdots \to y^*$,s the coeficient matrix A should be positive definite for all y; otherwise the search direction might not be a *descent direction*. In the case where the coeficient matrix is not positive definite, it is then imperative to carryout a modification on the matrix else the Conjugate Gradient Algorithm might fail. However, the modification of the Hessian matrix yields a modified matrix $\bar{A} = A + E$ that is now positive definite and preserves the information of the original matrix A with the

correction matrix $E = \tau I$ *(a multiple of the identity*) as small as possible so as to make it well-conditioned for the algorithm. In ensuring the positive definiteness of the matrix operator for smooth computation of the Cholesky factorization strategies could be deployed for modifying non-positive definite matrices using the *spectral decomposition* approach. Two decomposition strategies of the forms LL^T and LDL^T , called the *Cholesky factorization* and the *Modified Cholesky factorization* respectively, were used with L lower triangular matrix. However, the standard Cholesky factorization is preferable to the modified Cholesky because of its simplicity and convergence and as such will be deployed in this research. In this case, the symmetric Coeficient and error matrices be expressed as $A = P\lambda P^T$ and $E = P\tau IP^T$ respectively with $\tau \geq 0, \lambda = diag(\lambda_i)$ and $E = diag(\tau_i)$, then $\bar{A} = P[\lambda + \tau I]P^T = Pdiag(\lambda_i + \tau_i)P^T$ can be expressed as;

$$diag(\lambda_i + \tau_i) = \begin{bmatrix} (\lambda_1+\tau_1) & 0 & \cdots & 0 \\ 0 & (\lambda_2+\tau_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & (\lambda_n+\tau_n) \end{bmatrix}, \tag{25}$$

with $(\lambda_i + \tau_i) > 0 \forall i$ such that $\bar{A}$ is positive definite. Suppose the *minimum eigenvalue* of A, for which the matrix A is not positive definite, is $\lambda_{min}(A)$ and there exist any positive number $\delta > 0$ such that $\lambda_{min}(A) + \delta > 0$, then the minimum Euclidean (or Frobenius) norm, satisfies $\lambda_{min}(\bar{A}) = \lambda_{min}(A + \tau I) > 0$ for $(\lambda_i + \tau_i) > \delta \quad \forall i.$

Consequently, $\tau = max\{0, \delta - \lambda_{min}(A)\}$ for $\tau = \{\tau_1, \tau_2, \cdots, \tau_n\}$ and $\tau_1 = \begin{cases} \delta - \lambda_i & \lambda_i < \delta \\ 0 & \lambda_i \geq \delta \end{cases}$, provided $\lambda_{min}(A) + \delta > 0$. Therefore, the mathematical strategy is to develop an algorithm that will construct the modified (corrected) matrix at the minimum correction factor (τ^*) such that the modified matrix estimates the Coeficient matrix at a minimum error. In other words, the strategy is to search for the matrix $\bar{A}^*$ having the minimum correction

matrix (E^*) with minimum Euclidean (Frobenius) norm that satisfies$\lambda_{min}(A+E^*)+\delta>0$ and as well as the cholesky factorization (spectral decomposition) of the said matrix in the form LL^T .
The numerial solution to the derived linear system of equations can be ascertained using the direct numerical method or perhaps any of the numerical algorithms such as the Conjugate Gradient Method (CGM) or its variants (Preconditioned CGM,, Projection CGM), Newton's method etc.

RESULTS AND DISCUSSION
Example 1
Solve the delay diferential equation (DDE)

$$2y'(t)-t^2y(t)=4y(t-\tau) \quad for \quad -\tau\le t\le 2,$$

$$y(t)=t^2 \quad -\tau\le t\le 0,$$

$$y(0)=0 \ \ and \ \ \tau\in\{0.1,\ 0.3,\ 0.5,\ 0.7, 0.9\}$$

Applying the procedures in equation (20) - (23) yields the discretized matrices below for $\tau=0.3, T=2, N=10, \theta=0.5$ and $t_k=0.2k$ for $k=0,1,\cdots,10$.

$$\phi=\begin{pmatrix}-0.4000\\0\end{pmatrix}\in \mathrm{R}^2$$

$$B=\begin{pmatrix}0.2000 & 0.2000\\0 & 0.2000\\0&0\\0&0\\0&0\\0&0\\0&0\\0&0\\0&0\\0&0\end{pmatrix}\in \mathrm{R}^{10\times 2},$$

and

$$B\times\phi=\begin{pmatrix}-0.0800\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}\in \mathrm{R}^{10}.$$

$$A=\begin{pmatrix}
1.0020 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.2000 & 1.0080 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.2000 & 0.2000 & 1.0180 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0.2000 & 0.2000 & 1.0320 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0.2000 & 0.2000 & 1.0500 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0.2000 & 0.2000 & 1.0720 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.2000 & 0.2000 & 1.0980 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0.2000 & 0.2000 & 1.1280 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0.2000 & 0.2000 & 1.1620 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2000 & 0.2000 & 1.2000
\end{pmatrix}$$

The problem was solved by direct inverse method and the solution $y=(y_1,y_2,\cdots,y_{10})\in \mathrm{R}^{10}$ was represented graphically and outlined in the table below for various values of the delay constants (τ) and steplength (h).

Table 1: The Results with $T=2, \theta=0.5$

h	$\tau=0.1$	$\tau=0.3$	$\tau=0.5$	$\tau=0.7$	$\tau=0.9$
0.2000	0.0826	0.0826	1.3297	1.3297	3.8193
0.0200	0.0756	0.8151	2.7219	5.9330	9.7890
0.0020	0.2389	2.5769	8.6070	18.7396	30.8985
0.0002	0.7554	8.1487	27.2178	59.2526	97.6914

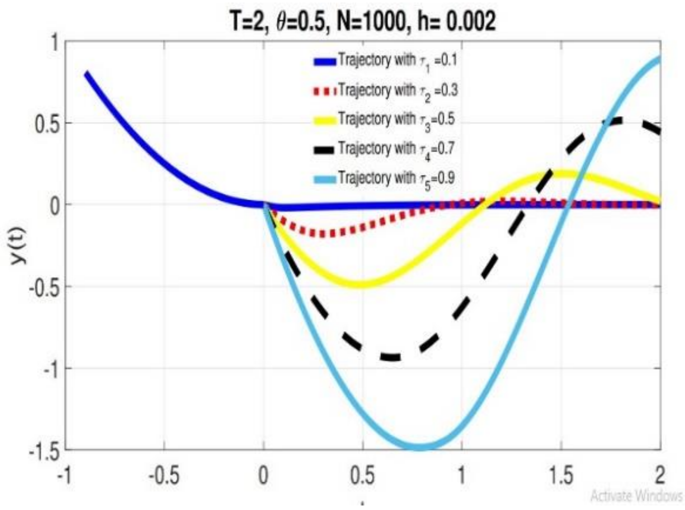

Figure 1: Results with T=2, $\theta=0.5$

Example 2

Solve the multiple delay differential equation (DDE)

$$y'(t) = t^2 y(t) + 2ty(t-0.3) + 3(t-1)y(t-0.5) + 5t^2, \quad -0.5 \le t \le T, \quad T \in \{1,2,2.5\},$$

$$y(t) = 2t^2, \quad -0.5 \le t \le 0,$$

$$y(0) = 0.$$

The discrete matrices by applying equation (20) - (23) is given in Table 2 below for $\tau = 0.3, T = 1, N = 10, \theta = 0.5$ and $t_k = 0.1k$ for $k = 0,1,\cdots,10$.

$$\begin{pmatrix} 0 \\ 0.5000 \\ 1.0000 \\ 1.5000 \\ 2.0000 \\ 2.5000 \\ 3.0000 \\ 3.5000 \\ 4.0000 \\ 4.5000 \\ 5.0000 \end{pmatrix} \in \mathbf{R}^{11},$$

$$\phi = \begin{pmatrix} 0.5000 \\ 0.3200 \\ 0.1800 \\ 0.0800 \\ 0.0200 \\ 0 \end{pmatrix} \in \mathbf{R}^{6},$$

$$B \times \phi = \begin{pmatrix} -0.1166 \\ -0.0634 \\ -0.0296 \\ -0.0102 \\ -0.0018 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \in \mathbf{R}^{10},$$

$$D \times E = \begin{pmatrix} 0.0250 \\ 0.0750 \\ 0.1250 \\ 0.1750 \\ 0.2250 \\ 0.2750 \\ 0.3250 \\ 0.3750 \\ 0.4250 \\ 0.4750 \end{pmatrix} \in \mathbf{R}^{10}$$

$$A = \begin{pmatrix}
0.9995 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1.0005 & 0.9980 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1.0020 & 0.9955 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0300 & 0 & -1.0045 & 0.9920 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0300 & 0.0400 & 0 & -1.0080 & 0.9875 & 0 & 0 & 0 & 0 & 0 \\
-0.0750 & 0.0400 & 0.0500 & 0 & -1.0125 & 0.9820 & 0 & 0 & 0 & 0 \\
-0.0750 & -0.0600 & 0.0500 & 0.0600 & 0 & -1.0180 & 0.9755 & 0 & 0 & 0 \\
0 & -0.0600 & -0.0450 & 0.0600 & 0.0700 & 0 & -1.0245 & 0.9680 & 0 & 0 \\
0 & 0 & -0.0450 & -0.0300 & 0.0700 & 0.0800 & 0 & -1.0320 & 0.9595 & 0 \\
0 & 0 & 0 & -0.0300 & -0.0150 & 0.0800 & 0.0900 & 0 & -1.0405 & 0.9500
\end{pmatrix}$$

$$D = \begin{pmatrix}
0.9995 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1.0005 & 0.9980 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0500 & 0.0500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 \\
0 & 0.0500 & 0.0500 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 4 \\
0 & 0 & 0.0500 & 0.0500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.0500 & 0.0500 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.0500 & 0.0500 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.0500 & 0.0500 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0500 & 0.0500 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0500 & 0.0500 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0500 & 0.0500 & 0
\end{pmatrix}$$

$$B = \begin{pmatrix}
-0.1500 & -0.1350 & 0 & 0.0200 & 0 & 0 \\
0 & -0.1350 & -0.1200 & 0.0100 & 0.0300 & 0 \\
0 & 0 & -0.1200 & -0.1050 & 0.0200 & 0 \\
0 & 0 & 0 & -0.1050 & -0.0900 & 0 \\
0 & 0 & 0 & 0 & -0.0900 & -0.0750 \\
0 & 0 & 0 & 0 & 0 & -0.0750 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \in \mathbf{R}^{10\times 6}.$$

$$Y = A^{-1}[(D \times E) + F + (B \times \phi)] \in \mathbf{R}^{10}. \qquad (26)$$

The solution, $\| y \|$, of MDDE is displayed in the table and graphs below. Figure 1 is the graph for Table 2 above, while Figures 1 and 2 are for varying values of the terminal time (T) for $T = 2$ and $T = 2.5$ respectively.

Table 2: The Results with $\tau_1 = 0.3,\ \tau_2 = 0.5,\ \theta = 0.5$

N	$T = 1.0$	$T = 1.5$	$T = 2.0$	$T = 2.5$	$T = 3.0$
10	3.6955	11.2757	2.7227	4.5328	5.4319
100	10.3855	24.7201	3.78190×10^1	42.9279	899.5692
1000	3.2440×10^1	7.6661×10^1	1.1482×10^2	1.3351×10^2	2.726×10^3
10000	1.0245×10^2	2.4195×10^2	3.6164×10^2	4.2164×10^2	8.5792×10^3

Figure 1 is the graph for table 2 above while figures 1 and 2 are for varying values of the terminal time (T) for $T = 2$ and $T = 2.5$ respectively.

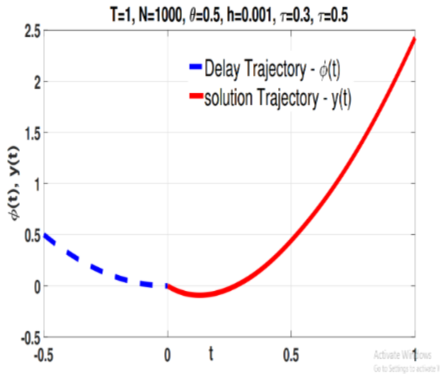

Figure 2: Results with T=1, $\theta = 0.5$

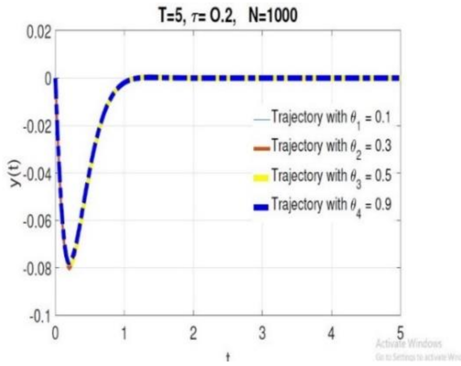

Figure 3: Results with T=5, r= 0.2

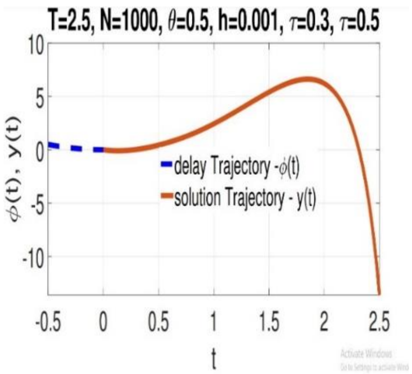

Figure 4: Results with T=2.5, $\theta = 0.5$

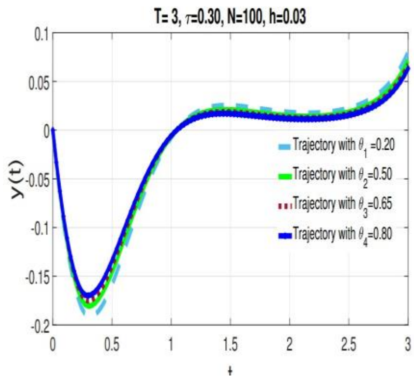

Figure 5: Results with T=0.30, N=100, h=0.03

CONCLUSION

The linear $\theta -$ method of discretization for multiple delay first order ordinary differential equation is simple, precise and reliable. The method has numerical B-stability for $\frac{1}{2} \leq \theta \leq 1$, consistent and thereby guarantees the convergence and accuracy of the discretized matrix operators in the derived linear system of equations in (Lu, 1991). The method can be extended to both linear and nonlinear complex first and higher order differential equations with or without delays.

ACKNOWLEDGMENT

The author thanks the referees for their suggestions that have greatly improved the presentation of this study.

REFERENCES

Adamu, S., Aduroja, O. O., and Bitrus, K. (2023), Numerical Solution to Optimal Control Problems Using Collocation Method via Pontryagin's Principle, *FUDMA Journzl of Sciences 7*(5) pp 228- 233.

Al-Mutib, A. N. (1984), Stability properties of numerical methods for solving delay differential equations, *J. Comput. Appl. Math.* 10 (1) 71-79. 121[31].

Barwell,V.K.(1975), Special stability problems for functional differential equations, *BIT 1*5 130-135.

Calvo, M and Grande,T. (1988) On the asymptotic stability of the θ − methods for delay differential equations, *Numer. Math*. 54 257-269.141

Dawodu, K. A. (2021) Modified ADMM Algorithm for solving proximal bound formulation of multi-delay optimal control problem with bounded control, *Iranian Journal of Numerical Analysis and Opyimization*, 2022; 1-3.

Liu, M.Z. and Spijker, M.N. (1990). The stability of the B-methods in the numerical solution of delay differential equations, *IMA J. Numer. Anal.* 10 31-48.

Lu, L (1991) Numerical stability of the B-methods for systems of differential equations with several delay terms. *Journal of Computational and Applied Mathematics* 34 pp 291-304.