

Research Article

Differential Transform Algorithm for Functional Differential Equations with Time-Dependent Delays

Josef Rebenda ^{1,2} and Zuzana Pátíková³

¹CEITEC BUT, Brno University of Technology, Purkyňova 656/123, 61200 Brno, Czech Republic

²Department of Mathematics, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 8, 616 00 Brno, Czech Republic

³Department of Mathematics, Faculty of Applied Informatics, Tomas Bata University in Zlín, Nad Stráněmi 4511, Zlín 76005, Czech Republic

Correspondence should be addressed to Josef Rebenda; josef.rebenda@ceitec.vutbr.cz

Received 23 August 2019; Accepted 31 December 2019; Published 28 February 2020

Guest Editor: Robert Hakl

Copyright © 2020 Josef Rebenda and Zuzana Pátíková. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An algorithm using the differential transformation which is convenient for finding numerical solutions to initial value problems for functional differential equations is proposed in this paper. We focus on retarded equations with delays which in general are functions of the independent variable. The delayed differential equation is turned into an ordinary differential equation using the method of steps. The ordinary differential equation is transformed into a recurrence relation in one variable using the differential transformation. Approximate solution has the form of a Taylor polynomial whose coefficients are determined by solving the recurrence relation. Practical implementation of the presented algorithm is demonstrated in an example of the initial value problem for a differential equation with nonlinear nonconstant delay. A two-dimensional neutral system of higher complexity with constant, nonconstant, and proportional delays has been chosen to show numerical performance of the algorithm. Results are compared against Matlab function DDENS.

1. Introduction

Functional differential equations (FDEs) are used to model processes and phenomena which depend on past values of the modelled entities. Indicatively, we mention models describing machine tool vibrations [1], predator-prey type models [2], and models used in economics [3]. Further models and details can be found for instance in [4, 5] or [6].

Differential transformation (DT), a semianalytical approach based on Taylor's theorem, has been proved to be efficient in solving a variety of initial value problems (IVPs), ranging from ordinary to functional, partial, and fractional differential equations [7–11]. However, there is no publication about systematic application of DT to IVP for

differential equations with nonconstant delays which are functions of the independent variable.

In this paper, we present an extension of DT to a class of IVPs for delayed differential equations with analytic right-hand side. Albeit the analyticity assumption seems to be quite restrictive, it is reasonable to develop theory for such class of equations [12, 13].

The paper is organised as follows. In Section 2, we define the subject of our study and briefly describe the methods we combine to solve the studied problem, including recalling necessary results of previous studies. Section 3 contains the main results of the paper, including algorithm description, new theorems, examples, and comparison of numerical results. In Section 4, we briefly summarise what has been done in the paper.

2. Methods

2.1. Problem Statement. The problem studied in this paper is to find a solution on a given finite interval $[t_0, T] \subset [0, \infty)$ to an IVP for the following system of p functional differential equations of n -th order with multiple delays $\alpha_1(t), \dots, \alpha_r(t)$:

$$\mathbf{u}^{(n)}(t) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{u}'(t), \dots, \mathbf{u}^{(n-1)}(t), \mathbf{u}_1(\alpha_1(t)), \dots, \mathbf{u}_r(\alpha_r(t))), \quad (1)$$

where $\mathbf{u}^{(n)}(t) = (u_1^{(n)}(t), \dots, u_p^{(n)}(t))^T$, $\mathbf{u}^{(k)}(t) = (u_1^{(k)}(t), \dots, u_p^{(k)}(t))$, $k = 0, 1, \dots, n-1$, and $\mathbf{f} = (f_1, \dots, f_p)^T$ are p -dimensional vector functions, $\mathbf{u}_i(\alpha_i(t)) = (\mathbf{u}(\alpha_i(t)), \mathbf{u}'(\alpha_i(t)), \dots, \mathbf{u}^{(m_i)}(\alpha_i(t)))$ are $(m_i \cdot p)$ -dimensional vector functions, $m_i \leq n$, $i = 1, 2, \dots, r$, $r \in \mathbb{N}$, and $f_j: [t_0, T]$ are real functions for $j = 1, 2, \dots, p$, where $\omega = \sum_{i=1}^r m_i$.

We assume that each $\alpha_i(t) = t - \tau_i(t)$, where $\tau_i(t) \geq \tau_{i0} > 0$ for $t \in [t_0, T]$, $i = 1, 2, \dots, r$, is in general a real function, that is, a time-dependent or time-varying delay. Constant and proportional delays are considered as special cases. In case that some α_i is a proportional delay, we do not require the condition $\tau_i(t) \geq \tau_{i0} > 0$ to be valid at 0 if $t_0 = 0$.

Let $t^* = \min_{1 \leq i \leq r} \{\inf_{t \in [t_0, T]} (\alpha_i(t))\}$ and $m = \max\{m_1, m_2, \dots, m_r\}$; hence, $t^* \leq t_0$ and $m \leq n$. If $m < n$, we have a retarded system (1); otherwise, if $m = n$, we call the system neutral. Furthermore, if $t^* < 0$, initial vector function $\Phi(t) = (\phi_1(t), \dots, \phi_p(t))^T$ must be prescribed on the interval $[t^*, t_0]$.

DT algorithm for the case $t^* = t_0 = 0$ with all delays being proportional is described in [14]. DT algorithm for the case $t^* < t_0$ when all delays are constant is introduced in [15]. In this paper, we develop the algorithm for the case $t^* < t_0$ when at least one delay is nonconstant.

To have a complete IVP, we consider system (1) together with initial conditions:

$$\begin{aligned} \mathbf{u}(t_0) &= \mathbf{v}_0, \\ \mathbf{u}'(t_0) &= \mathbf{v}_1, \dots, \mathbf{u}^{(n-1)}(t_0) = \mathbf{v}_{n-1}, \end{aligned} \quad (2)$$

and, since $t^* < t_0$, also subject to initial vector function $\Phi(t)$ on interval $[t^*, t_0]$ such that

$$\Phi(t_0) = \mathbf{u}(t_0), \dots, \Phi^{(n-1)}(t_0) = \mathbf{u}^{(n-1)}(t_0). \quad (3)$$

We consider the IVPs (1)–(3) under the following hypotheses:

(H1) We assume that all the functions $\phi_j(t)$, $j = 1, \dots, p$, are analytic in $[t^*, t_0]$, the functions $\alpha_i(t)$, $i = 1, \dots, r$, are analytic in $[t_0, T]$ and the functions f_j , $j = 1, \dots, p$, are analytic in an open set containing $[t_0, T] \times [\mathbf{u}(t_0), \mathbf{u}(T)] \times \dots \times [\mathbf{u}_r(\alpha_r(t_0)), \mathbf{u}_r(\alpha_r(T))]$.

(H2) If $\alpha_i(t) = q_i t$ and $m_i = n$ in f_j for some $i \in \{1, \dots, r\}$ and $j \in \{1, \dots, p\}$, that is, j th equation is neutral with respect to the proportional delay α_i , we assume that $u_l^{(n)}(\alpha_i(t)) \equiv 0$ for $l \in \{1, \dots, p\}$, $l \neq j$. This hypothesis is included since if it is not fulfilled, the existence of unique solution of IVP could be violated.

We note that these assumptions imply that the IVP (1)–(3) has a unique solution in the interval $[t_0, T]$.

2.2. Method of Steps. The basic idea of our approach is to combine DT and the general method of steps. The method of steps enables us to replace the terms including delays with initial vector function $\Phi(t)$ and its derivatives. Then, the original IVP for the delayed or neutral system of differential equations is turned into IVP for a system of ordinary differential equations.

For the sake of clarity, we include a simple explanatory example. Suppose that we have a system with three delays, one of each type considered: $\alpha_1(t) = t - \tau_1(t)$, $\alpha_2(t) = t - \tau_2$, and $\alpha_3(t) = q_3 t$. We have to distinguish two cases:

- (a) If $t_0 = 0$, applying the method of steps turns system (1) into

$$\begin{aligned} \mathbf{u}^{(n)}(t) &= \mathbf{f}(t, \mathbf{u}(t), \dots, \mathbf{u}^{(n-1)}(t), \\ &\quad \Phi_1(t - \tau_1(t)), \Phi_2(t - \tau_2), \mathbf{u}_3(q_3 t)), \end{aligned} \quad (4)$$

while

- (b) If $t_0 > 0$, system (1) is simplified to

$$\begin{aligned} \mathbf{u}^{(n)}(t) &= \mathbf{f}(t, \mathbf{u}(t), \dots, \mathbf{u}^{(n-1)}(t), \\ &\quad \Phi_1(t - \tau_1(t)), \Phi_2(t - \tau_2), \Phi_3(q_3 t)), \end{aligned} \quad (5)$$

where

$$\begin{aligned} \Phi_1(t - \tau_1(t)) &= \left(\Phi(t - \tau_1(t)), \Phi'(t - \tau_1(t)), \dots, \right. \\ &\quad \left. \Phi^{(m_1)}(t - \tau_1(t)) \right), \\ \Phi_2(t - \tau_2) &= \left(\Phi(t - \tau_2), \Phi'(t - \tau_2), \dots, \Phi^{(m_2)}(t - \tau_2) \right), \\ \mathbf{u}_3(q_3 t) &= \left(\mathbf{u}(q_3 t), \mathbf{u}'(q_3 t), \dots, \mathbf{u}^{(m_3)}(q_3 t) \right), \\ \Phi_3(q_3 t) &= \left(\Phi(q_3 t), \Phi'(q_3 t), \dots, \Phi^{(m_3)}(q_3 t) \right), \end{aligned} \quad (6)$$

and $m_l \leq n$ for $l = 1, 2, 3, 4$. More details on the general method of steps can be found, for instance, in monographs [4] or [6].

Continuation of the method of steps algorithm for equations with constant delays τ_1, \dots, τ_r is described in [15]. Briefly summarised, the interval $[t_0, T]$ is divided into subintervals $I_l = [t_{l-1}, t_l]$, $l = 1, \dots, K$, where $t_K = T$ and t_l , $l = 1, \dots, K-1$, are the principal discontinuity points which is the set of points $t_{\rho, \sigma}$, such that $t_{0,1} = t_0$ and for $\rho, \sigma \geq 1$, $t_{\rho, \sigma}$ are the minimal roots with odd multiplicity of r equations:

$$t_{\rho, (\sigma-1)r+\mu} - \tau_\mu = t_{\rho-1, \sigma}, \quad \mu = 1, \dots, r. \quad (7)$$

If nonconstant nonproportional delays α_i appear in system (1), the principal set of discontinuity points is defined as follows:

Definition 1. The principal discontinuity points for the solutions of system (1) are given by the set of points $t_{\rho,\sigma}$, such that $t_{0,1} = t_0$ and for $\rho, \sigma \geq 1$, $t_{\rho,\sigma}$ are the minimal roots with odd multiplicity of r equations:

$$\alpha_\mu(t_{\rho,(\sigma-1)r+\mu}) = t_{\rho-1,\sigma} \quad \mu = 1, \dots, r. \quad (8)$$

Similar to the case of constant delays, we break the interval $[t_0, T]$ into subintervals $I_l = [t_{l-1}, t_l]$, $l = 1, \dots, K$. We start with the mesh grid $\{t_0, \dots, t_K\}$ formed by the principal discontinuity points calculated using Definition 1. To improve convergence or performance of the algorithm, there is a possibility to refine the mesh grid by inserting other points into it. For more details on the principal discontinuity points and mesh grid, we refer to the monograph [16].

2.3. Differential Transformation

Definition 2. Differential transformation of a real function $u(t)$ at a point $t_0 \in \mathbb{R}$ is $\mathcal{D}\{u(t)\}[t_0] = \{U(k)[t_0]\}_{k=0}^\infty$, where $U(k)[t_0]$, k -th component of the differential transformation of the function $u(t)$ at t_0 , $k \in \mathbb{N}_0$, is defined as

$$U(k)[t_0] = \frac{1}{k!} \left[\frac{d^k u(t)}{dt^k} \right]_{t=t_0}, \quad (9)$$

provided that the original function $u(t)$ is analytic in a neighbourhood of t_0 .

Definition 3. Inverse differential transformation of $\{U(k)[t_0]\}_{k=0}^\infty$ is defined as

$$u(t) = \mathcal{D}^{-1}\{\{U(k)[t_0]\}_{k=0}^\infty\}[t_0] = \sum_{k=0}^\infty U(k)[t_0] (t - t_0)^k. \quad (10)$$

In applications, the function $u(t)$ is expressed by a finite sum

$$u(t) = \sum_{k=0}^N U(k)[t_0] (t - t_0)^k. \quad (11)$$

As we can observe in (10), DT is based on Taylor series; hence, any theorem about convergence of Taylor series may be used. However, we would like to point out the paper [17] where the finest general explicit *a priori* error estimates are given.

The following formulas are listed, e.g., in [18] and will be used in Section 3.3.

Lemma 1. Assume that $F(k)[t_0]$ and $U(k)[t_0]$ are differential transformations of functions $f(t)$ and $u(t)$, respectively:

$$\text{If } f(t) = \frac{d^n u(t)}{dt^n}, \text{ then } F(k)[t_0] = \frac{(k+n)!}{k!} U(k+n)[t_0].$$

$$\text{If } f(t) = t^n, \text{ then } F(k)[0] = \delta(k-n),$$

where $\delta(k-n) = \delta_{kn}$ is the Kronecker delta.

$$\text{If } f(t) = e^{\lambda t}, \text{ then } F(k)[0] = \frac{\lambda^k}{k!}. \quad (12)$$

Remark 1. Similar formulas can be obtained using numerical approach called Functional Analytical Technique based on Operator Theory [19, 20].

The main disadvantage of many papers about DT is that there are almost no examples of equations with non-polynomial nonlinear terms containing unknown function $u(t)$ like, for instance, $f(u) = \sqrt[3]{1+u^3}$ or $f(u) = e^{\sqrt{\sin u}}$. However, DT of components containing nonlinear terms can be obtained in a consistent way using the algorithm described in [21].

Theorem 1. Let g and f be real functions analytic near t_0 and $g(t_0)$, respectively, and let h be the composition $h(t) = (f \circ g)(t) = f(g(t))$. Denote $\mathcal{D}\{g(t)\}[t_0] = \{G(k)\}_{k=0}^\infty$, $\mathcal{D}\{f(t)\}[g(t_0)] = \{F(k)\}_{k=0}^\infty$, and $\mathcal{D}\{(f \circ g)(t)\}[t_0] = \{H(k)\}_{k=0}^\infty$ as the differential transformations of functions g, f , and h at $t_0, g(t_0)$, and t_0 , respectively. Then, the numbers $H(k)$ in the sequence $\{H(k)\}_{k=0}^\infty$ satisfy the relations $H(0) = F(0)$ and

$$H(k) = \sum_{l=1}^k F(l) \cdot \widehat{B}_{k,l}(G(1), \dots, G(k-l+1)), \quad \text{for } k \geq 1, \quad (13)$$

where $\widehat{B}_{k,l}(\widehat{x}_1, \dots, \widehat{x}_{k-l+1})$ are the partial ordinary Bell polynomials.

The following Lemma proved in [21] is useful when calculating partial ordinary Bell polynomials.

Lemma 2. The partial ordinary Bell polynomials $\widehat{B}_{k,l}(\widehat{x}_1, \dots, \widehat{x}_{k-l+1})$, $l = 1, 2, \dots, k \geq l$, satisfy the recurrence relation

$$\widehat{B}_{k,l}(\widehat{x}_1, \dots, \widehat{x}_{k-l+1}) = \sum_{i=1}^{k-l+1} \frac{i \cdot l}{k} \widehat{x}_i \widehat{B}_{k-i,l-1}(\widehat{x}_1, \dots, \widehat{x}_{k-i-l+2}), \quad (14)$$

where $\widehat{B}_{0,0} = 1$ and $\widehat{B}_{k,0} = 0$ for $k \geq 1$.

3. Results and Discussion

3.1. Algorithm Description. Recall system (1)

$$\mathbf{u}^{(n)}(t) = \mathbf{f}(t, \mathbf{u}(t), \mathbf{u}'(t), \dots, \mathbf{u}^{(n-1)}(t), \mathbf{u}_1(\alpha_1(t)), \dots, \mathbf{u}_r(\alpha_r(t))), \quad (15)$$

with initial conditions

$$\mathbf{u}(t_0) = \mathbf{v}_0, \mathbf{u}'(t_0) = \mathbf{v}_1, \dots, \mathbf{u}^{(n-1)}(t_0) = \mathbf{v}_{n-1}, \quad (16)$$

and initial vector function $\Phi(t)$ on interval $[t^*, t_0]$ satisfying

$$\Phi(t_0) = \mathbf{u}(t_0), \dots, \Phi^{(n-1)}(t_0) = \mathbf{u}^{(n-1)}(t_0). \quad (17)$$

Further recall that in Section 2.2, we broke the interval $[t_0, T]$ into subintervals $I_l = [t_{l-1}, t_l]$, $l = 1, \dots, K$. Define $I_0 = [t^*, t_0]$.

Then, we are looking for a solution $\mathbf{u}(t)$ of the IVP (1)–(3) in the form

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_{I_1}(t), & t \in I_1, \\ \mathbf{u}_{I_2}(t), & t \in I_2, \\ \vdots \\ \mathbf{u}_{I_K}(t), & t \in I_K, \end{cases} \quad (18)$$

where solution \mathbf{u}_{I_j} in the j th interval I_j is obtained in the following way. We solve the following equation:

$$\begin{aligned} \mathbf{u}_{I_j}^{(n)}(t) &= f\left(t, \mathbf{u}_{I_j}(t), \mathbf{u}'_{I_j}(t), \dots, \mathbf{u}_{I_j}^{(n-1)}(t), \right. \\ &\quad \left. \mathbf{u}_{j,1}(\alpha_1(t)), \dots, \mathbf{u}_{j,r}(\alpha_r(t))\right), \end{aligned} \quad (19)$$

where

$$\mathbf{u}_{j,i}(\alpha_i(t)) = \left(\mathbf{u}_{I_l}(\alpha_i(t)), \mathbf{u}'_{I_l}(\alpha_i(t)), \dots, \mathbf{u}_{I_l}^{(m_i)}(\alpha_i(t)) \right), \quad (20)$$

if $\alpha_i(t) \in I_l$, for $t \in I_j$, $l \in \{1, \dots, j\}$, $j \in \{1, \dots, K\}$.

In case that $\alpha_i(t) \in I_0 = [t^*, t_0]$ for $t \in I_j$, then again

$$\mathbf{u}_{j,i}(\alpha_i(t)) = \left(\phi(\alpha_i(t)), \phi'(\alpha_i(t)), \dots, \phi^{(m_i)}(\alpha_i(t)) \right). \quad (21)$$

Application of DT at t_{j-1} to equation (19) yields a system of recurrence algebraic equations:

$$\mathbf{U}_{I_j}(k+n)[t_{j-1}] = \mathbf{F}\left(k, \mathbf{U}_{I_j}(k), \mathbf{U}_{I_j}(k+1), \dots, \mathbf{U}_{I_j}(k+n-1)\right), \quad (22)$$

where the function \mathbf{F} is the DT of the righthand side of equation (19) and involves application of Theorem 1.

Next, we transform the initial conditions (2). Following Definition 2, we derive

$$\mathbf{U}_{I_j}(k)[t_{j-1}] = \frac{1}{k!} \mathbf{u}_{I_j}^{(k)}(t_{j-1}), \quad \text{for } k = 0, 1, \dots, n-1, j \in \{1, \dots, K\}. \quad (23)$$

Using (22) with (23) and then inverse transformation rule, we obtain approximate solution to (19) in the form of Taylor series:

$$\mathbf{u}_{I_j}(t) = \sum_{k=0}^{\infty} \mathbf{U}_{I_j}(k)[t_{j-1}](t-t_{j-1})^k, \quad t \in I_j, \quad (24)$$

for all $j \in \{1, \dots, K\}$.

To transform (20) correctly, we need the following theorem.

Theorem 2. Let $\alpha_i(t) \in I_l$ for $t \in I_j$, where $l \in \{1, \dots, j-1\}$. Let $p \in \mathbb{N}$. Denote $\mathcal{D}\{\alpha_i(t)\}[t_{j-1}] = \{A_i(k)[t_{j-1}]\}_{k=0}^{\infty}$. Then,

$$\begin{aligned} &\mathcal{D}\{u_{I_l}^{(p)}(\alpha_i(t))\}[t_{j-1}] \\ &= \left\{ \sum_{y=0}^k \frac{(y+p)!}{y!} U_{I_l}(y+p)[\alpha_i(t_{j-1})] \right. \\ &\quad \left. \cdot \widehat{B}_{k,y}(A_i(1)[t_{j-1}], \dots, A_i(k-y+1)[t_{j-1}]) \right\}_{k=0}^{\infty}, \end{aligned} \quad (25)$$

where $\widehat{B}_{0,0} = 1$, $\widehat{B}_{k,0} = 0$ for $k \geq 1$, and

$$\begin{aligned} U_{I_l}(y)[\alpha_i(t_{j-1})] &= \sum_{x=0}^{\infty} \binom{x+y}{x} (\alpha_i(t_{j-1}) - t_{l-1})^x \\ &\quad \cdot U_{I_l}(x+y)[t_{l-1}], \end{aligned} \quad (26)$$

for $y \geq 0$.

Proof. To prove (25) with $p = 0$, we use Theorem 1 with $f(t) = u_{I_l}(t)$, $g(t) = \alpha_i(t)$, and $h(t) = (f \circ g)(t)$. We immediately get

$$\begin{aligned} H(k)[t_{j-1}] &= \sum_{y=1}^k U_{I_l}(y)[\alpha_i(t_{j-1})] \\ &\quad \cdot \widehat{B}_{k,y}(A_i(1)[t_{j-1}], \dots, A_i(k-y+1)[t_{j-1}]), \end{aligned} \quad (27)$$

for $k \geq 1$. For $k = 0$, Theorem 1 yields $H(0)[t_{j-1}] = U_{I_l}(0)[\alpha_i(t_{j-1})] = U_{I_l}(0)[\alpha_i(t_{j-1})] \cdot \widehat{B}_{0,0}(A_i(1)[t_{j-1}])$. Now, (25) for $p > 0$ is a consequence of Lemma 1 and it remains to prove (26). We recall that

$$\mathbf{u}_{I_l}(t) = \sum_{k=0}^{\infty} \mathbf{U}_{I_l}(k)[t_{l-1}](t-t_{l-1})^k, \quad t \in I_l. \quad (28)$$

As the assumption was that $\alpha_i(t_{j-1}) \in I_l$, we may apply Definition 2 to (28) and obtain

$$\begin{aligned} U_{I_l}(y)[\alpha_i(t_{j-1})] &= \frac{1}{y!} \left[\frac{d^y u_{I_l}(t)}{dt^y} \right]_{t=t_0} \frac{1}{y!} \sum_{z=y}^{\infty} \frac{z!}{(z-y)!} U_{I_l}(z)[t_{l-1}] \\ &\quad \cdot (t_0 - t_{l-1})^{z-y}. \end{aligned} \quad (29)$$

Substituting $t_0 = \alpha_i(t_{j-1})$ and $z = x + y$ gives (26). \square

3.2. *New DT Formulas.* In the applications, we also use the following DT formulas.

Theorem 3. Assume that $F(k)[t_0]$ is the differential transformation of the function $f(t)$ and $r \in \mathbb{R}$:

(a) If $f(t) = t^r$, then $F(k)[t_0] = \binom{r}{k} t_0^{r-k}$ for all t such that $|t - t_0| < |t_0|$,

where $\binom{r}{k} = r(r-1)\dots(r-k+1)/k! = (r)_k/k!$ and $(r)_k$ is the Pochhammer symbol.

(b) If $f(t) = \ln(t)$, then $F(k)[t_0] = (-1)^{k-1}/(k \cdot t_0^k)$ for $k \geq 1$.

Proof

(a) Recall the Newton's generalisation of the binomial formula: if x and y are real numbers with $|x| > |y|$, and r is any complex number, one has

$$(x+y)^r = \sum_{k=0}^{\infty} \binom{r}{k} x^{r-k} y^k, \quad (30)$$

where $\binom{r}{k} = r(r-1)\dots(r-k+1)/k!$. Let us rewrite t^r as $t^r = (t - t_0 + t_0)^r = (t_0 + (t - t_0))^r$. Applying (30) yields

$$t^r = \sum_{k=0}^{\infty} \binom{r}{k} t_0^{r-k} (t - t_0)^k. \quad (31)$$

(b) We start by proving the formula

$$(\ln(t))^{(k)} = \frac{(-1)^{k-1} (k-1)!}{t^k}, \quad (32)$$

by induction. For $k = 1$, we have $(\ln(t))' = 1/t$; hence, (32) is valid. Suppose that (32) holds for k . Then,

$$\begin{aligned} (\ln(t))^{(k+1)} &= ((\ln(t))^{(k)})' = \left(\frac{(-1)^{k-1} (k-1)!}{t^k} \right)' \\ &= (-1)^{k-1} (k-1)! (t^{-k})' \\ &= (-1)^{k-1} (k-1)! (-k) t^{-k-1} \\ &= \frac{(-1)^k k!}{t^{k+1}}. \end{aligned} \quad (33)$$

Thus, formula (32) is valid for all $k \in \mathbb{N}$. Now by Definition 2,

$$\begin{aligned} F(k)[t_0] &= \frac{1}{k!} \left[\frac{d^k \ln(t)}{dt^k} \right]_{t=t_0} = \frac{1}{k!} \left[\frac{(-1)^{k-1} (k-1)!}{t^k} \right]_{t=t_0} \\ &= \frac{(-1)^{k-1}}{k \cdot t_0^k}. \end{aligned} \quad (34)$$

□

3.3. *Applications.* In this section, we introduce two test problems and show how the practical implementation of the presented algorithm looks like in concrete examples. Comparison of numerical results is given in Section 3.4.

As the first test problem, we choose an IVP for a scalar equation with one nonconstant delay where the exact solution is known to be the exponential function e^t . The purpose of including this example is to compare results obtained by DT against values of the exact solution and also against results obtained by Matlab function DDENSD which has been designed to approximate solutions to IVP for neutral delayed differential equations.

Example 1. Consider the delayed equation:

$$u'(t) = u(t) - t + u(\ln(t)), \quad (35)$$

with the initial condition

$$u(1) = e, \quad (36)$$

and with the initial function

$$\phi(t) = e^t, \quad t \in [0, 1]. \quad (37)$$

First we find the differential transform of the initial condition (36) which is $U(0)[1] = e$. Further denote $\mathcal{D}\{e^t\}[0] = \{E(k)[0]\}_{k=0}^{\infty}$ as the transformation of the exponential function with the center at 0 and $\mathcal{D}\{\ln(t)\}[1] = \{F(k)[1]\}_{k=0}^{\infty}$ as the transformation of the logarithmic function at 1, respectively. Then, Lemma 1 and Theorem 3 yield

$$E(k)[0] = \frac{1}{k!}, \quad (38)$$

$$F(k)[1] = \frac{(-1)^{k-1}}{k}, \quad \text{for } k \geq 1.$$

For $t \in [1, e]$, equation (35) is transformed into

$$(k+1)U(k+1)[1] = U(k)[1] - \delta(k) - \delta(k-1) + H(k)[1], \quad (39)$$

where

$$H(k)[1] = \sum_{l=1}^k E(l)[0] \widehat{B}_{k,l}(F(1)[1], \dots, F(k-l+1)[1]), \quad \text{for } k \geq 1,$$

$$H(0)[1] = E(0)[0] = 1. \quad (40)$$

We have

$$\begin{aligned}
U(1)[1] &= e - 1 - 0 + 1 = e, \\
H(1)[1] &= E(1)[0] \cdot F(1)[1] = 1 \cdot 1 = 1, \\
U(2)[1] &= \frac{1}{2}(U(1)[1] - 0 - 1 + H(1)[1]) \\
&= \frac{1}{2}(e - 1 + 1) = e \cdot \frac{1}{2}, \\
H(2)[1] &= E(1)[0] \cdot F(2)[1] + E(2)[0] \cdot (F(1)[1])^2 \\
&= 1 \cdot \left(-\frac{1}{2}\right) + \frac{1}{2} \cdot 1 = 0, \\
U(3)[1] &= \frac{1}{3}(U(2)[1] + H(2)[1]) = \frac{1}{3} \cdot e \cdot \frac{1}{2} = e \cdot \frac{1}{3!}, \\
&\vdots
\end{aligned} \tag{41}$$

Using the inverse transformation, we see that for $t \in [1, e]$,

$$u(t) = e \left(1 + (t-1) + \frac{(t-1)^2}{2} + \frac{(t-1)^3}{3!} \dots \right) = e \cdot e^{t-1} = e^t, \tag{42}$$

which corresponds to the exact solution to the IVPs (35)–(37).

In the second step of the method of steps, i.e., in the interval $t \in [e, e^e]$, we know that $u(t) = e^t$ for $t \in [1, e]$ and equation (35) is transformed into

$$(k+1)U(k+1)[e] = U(k)[e] - e\delta(k) - \delta(k-1) + H(k)[e], \tag{43}$$

where

$$\begin{aligned}
H(0)[e] &= U(0)[1], \\
H(k)[e] &= \sum_{l=1}^k U(l)[1] \widehat{B}_{k,l}(F(1)[e], \dots, F(k-l+1)[e]), \\
&\text{for } k \geq 1.
\end{aligned} \tag{44}$$

Here, $F(k)[e]$, according to Theorem 3, are coefficients of Taylor series of logarithmic function with the center at e :

$$\begin{aligned}
F(0)[e] &= 1, \\
F(k)[e] &= \frac{(-1)^{k-1}}{k \cdot e^k}, \quad \text{for } k \geq 1.
\end{aligned} \tag{45}$$

Taking the values calculated in the first step and substituting them into the recurrence formulas (43) and (44), we obtain

$$\begin{aligned}
U(0)[e] &= u(e) = e^e, \\
U(1)[e] &= U(0)[e] - e\delta(0) - \delta(-1) + H(0)[e] \\
&= e^e - e - 0 + e = e^e, \\
H(1)[e] &= U(1)[1] \cdot F(1)[e] = e \cdot \frac{1}{e} = 1, \\
U(2)[e] &= \frac{1}{2}(U(1)[e] - 0 - 1 + H(1)[e]) \\
&= \frac{1}{2}[e^e - 1 + 1] = \frac{1}{2}e^e, \\
H(2)[e] &= U(1)[1] \cdot \widehat{B}_{2,1}(F(1)[e], F(2)[e]) \\
&\quad + U(2)[1] \cdot \widehat{B}_{2,2}(F(1)[e]) \\
&= e \cdot F(2)[e] + \frac{e}{2} \cdot (F(1)[e])^2 \\
&= e \cdot \frac{1}{2} \left(\frac{-1}{e^2}\right) + \frac{e}{2} \cdot \left(\frac{1}{e}\right)^2 = 0, \\
U(3)[e] &= \frac{1}{3}(U(2)[e] + H(2)[e]) = \frac{1}{3!}e^e, \\
H(3)[e] &= U(1)[1] \cdot \widehat{B}_{3,1}(F(1)[e], F(2)[e], F(3)[e]) \\
&\quad + U(2)[1] \cdot \widehat{B}_{3,2}(F(1)[e], F(2)[e]) + U(3)[1] \\
&\quad \cdot \widehat{B}_{3,3}(F(1)[e]) \\
&= e \cdot F(3)[e] + \frac{e}{2} \cdot 2 \cdot F(1)[e] \cdot F(2)[e] + \frac{e}{3!} (F(1)[e])^3 \\
&= e \cdot \frac{2}{e^3} \frac{1}{3!} + e \cdot \frac{1}{e} \cdot \frac{(-1)}{e^2} \cdot \frac{1}{2} + \frac{e}{6} \cdot \frac{1}{e^3} = \frac{1}{e^2} \left(\frac{1}{3} - \frac{1}{2} + \frac{1}{6}\right) = 0, \\
U(4)[e] &= \frac{1}{4}(U(3)[e] + H(3)[e]) = \frac{1}{4!}e^e.
\end{aligned} \tag{46}$$

Hence, for $t \in [e, e^e]$, we have

$$\begin{aligned}
u(t) &= e^e + e^e(t-e) + \frac{1}{2}e^e(t-e)^2 + \frac{1}{3!}e^e(t-e)^3 \\
&\quad + \frac{1}{4!}e^e(t-e)^4 + \dots \\
&= e^e \left(1 + (t-e) + \frac{(t-e)^2}{2} + \frac{(t-e)^3}{3!} + \frac{(t-e)^4}{4!} + \dots \right) \\
&= e^e \cdot e^{t-e} = e^t,
\end{aligned} \tag{47}$$

which again coincides with the exact solution to problems (35)–(37).

In the second application, we have chosen an IVP for a nonlinear system of neutral delayed differential equations taken from the fully open access paper [18]. There are several reasons to test the proposed algorithm on the particular problem. The first is that the problem involves a nonlinear system of neutral equations of high complexity whose exact solution is unknown. Secondly, the proposed algorithm is a

complete differential transform version of the algorithm presented in [18] where modified Adomian formula has been used. Furthermore, the calculations done in [18] are shown only for the first step of the method of steps up to the first principal discontinuity point, whereas we continue calculations beyond that point in this paper. Last but not least, we want to verify performance and reproduce values obtained by DT and published in [18]. Rebenda et al. [18] has been submitted 4 years ago for the first time, and since that time, the Maple source code has been lost.

Example 2. Consider a nonlinear system of neutral delayed differential equations:

$$\begin{aligned} u_1''' &= u_1'''(t-2)u_1\left(\frac{t}{3}\right) + \sqrt[3]{(u_1(t))^2} + u_2'\left(t - \frac{1}{2}e^{-t}\right), \\ u_2''' &= \frac{1}{2}u_2'''\left(\frac{t}{2}\right) + u_2'(t-1)u_1\left(\frac{t}{3}\right), \end{aligned} \quad (48)$$

with initial functions

$$\begin{aligned} \phi_1(t) &= e^t, \\ \phi_2(t) &= t^2, \end{aligned} \quad (49)$$

for $t \in [-2, 0]$, and initial conditions

$$\begin{aligned} u_1(0) &= 1, \\ u_1'(0) &= 1, \\ u_1''(0) &= 1, \\ u_2(0) &= 0, \\ u_2'(0) &= 0, \\ u_2''(0) &= 2. \end{aligned} \quad (50)$$

For $t \in [0, t_1]$, where $t_1 \approx 0,351734$ is the minimal root of $t - (1/2)e^{-t} = 0$, using the method of steps, we obtain

$$\begin{aligned} u_1''' &= e^{(t-2)}u_1\left(\frac{t}{3}\right) + \sqrt[3]{(u_1(t))^2} + 2t - e^{-t}, \\ u_2''' &= \frac{1}{2}u_2'''\left(\frac{t}{2}\right) + 2(t-1)u_1\left(\frac{t}{3}\right). \end{aligned} \quad (51)$$

We need to find the differential transform of the considered problem. We notice that system (2) contains nonlinear term $h(t) = \sqrt[3]{(u_1(t))^2}$. To get DT of this term, $\mathcal{D}\{h(t)\}[0] = \{H_1(k)[0]\}_{k=0}^{\infty}$, and we apply Theorem 1. First, applying DT to system (2) at $t_0 = 0$, we get the recurrent system:

$$\begin{aligned} (k+1)(k+2)(k+3)U_1(k+3)[0] \\ = e^{-2} \sum_{l=0}^k \frac{1}{l!} \left(\frac{1}{3}\right)^{k-l} U_1(k-l)[0] + H_1(k)[0] + 2\delta(k) - \frac{(-1)^k}{k!}, \end{aligned} \quad (52)$$

$$\begin{aligned} (k+1)(k+2)(k+3) \left(1 - \frac{1}{2^{k+1}}\right) U_2(k+3)[0] \\ = 2 \left(\frac{1}{3}\right)^{k-1} U_1(k-1)[0] - \frac{2}{3^k} U_1(k)[0]. \end{aligned} \quad (53)$$

Denote $g(t) = t^{2/3}$; then, $h(t) = (g \circ u_1)(t)$, and following Theorem 1, we obtain

$$\begin{aligned} H_1(0)[0] &= G_1(0)[1], \\ H_1(k)[0] &= \sum_{l=1}^k G_1(l)[1] \tilde{B}_{k,l}(U_1(1)[0], \dots, U_1(k-l+1)[0]), \end{aligned} \quad (54)$$

for $k \geq 1$, where $\mathcal{D}\{g(t)\}[1] = \{G_1(k)[1]\}_{k=0}^{\infty}$ and, Theorem 3 being applied, $G_1(k)[1] = \binom{2/3}{k}$ for $k \geq 0$. Furthermore, the transformed initial conditions are

$$\begin{aligned} U_1(0)[0] &= 1, \\ U_1(1)[0] &= 1, \\ U_1(2)[0] &= \frac{1}{2}, \\ U_2(0)[0] &= 0, \\ U_2(1)[0] &= 0, \\ U_2(2)[0] &= 1. \end{aligned} \quad (55)$$

Using them, we compute the first three coefficients of the nonlinear term $h(t)$:

$$\begin{aligned} H_1(0)[0] &= G_1(0)[1] = 1, \\ H_1(1)[0] &= G_1(1)[1] \cdot B_{1,1}(U_1(1)[0]) = \frac{2}{3} \cdot 1 = \frac{2}{3}, \\ H_1(2)[0] &= G_1(1)[1] \cdot B_{2,1}(U_1(1)[0], U_1(2)[0]) \\ &\quad + G_1(2)[1] \cdot B_{2,2}(U_1(1)[0]) = \frac{2}{3} \cdot \frac{1}{2} - \frac{1}{9} \cdot 1 = \frac{2}{9}. \end{aligned} \quad (56)$$

Solving recurrent systems (52) and (53), we get

$$\begin{aligned} k=0: U_1(3)[0] &= \frac{1}{6}(e^{-2}U_1(0)[0] + H_1(0)[0] + 1) \\ &= \frac{2+e^{-2}}{6}, \\ U_2(3)[0] &= \frac{1}{3}(-2U_1(0)[0]) = -\frac{2}{3}, \\ k=1: U_1(4)[0] &= \frac{e^{-2}}{24} \left(\frac{1}{3}U_1(1)[0] + U_1(0)[0]\right) \\ &\quad + \frac{1}{24}(H_1(1)[0] + 1) = \frac{4e^{-2}+5}{72}, \\ U_2(4)[0] &= \frac{1}{18} \left(2U_1(0)[0] - \frac{2}{3}U_1(1)[0]\right) = \frac{2}{27}, \\ k=2: U_1(5)[0] &= \frac{e^{-2}}{60} \left(\frac{1}{9}U_1(2)[0] + \frac{1}{3}U_1(1)[0]\right) \\ &\quad + \frac{1}{2}U_1(0)[0] + \frac{1}{60}(H_1(2)[0] - \frac{1}{2}) \\ &= \frac{16e^{-2}-5}{1080}, \\ U_2(5)[0] &= \frac{2}{105} \left(\frac{2}{3}U_1(1)[0] - \frac{2}{9}U_1(2)[0]\right) = \frac{2}{189}. \end{aligned} \quad (57)$$

Using the inverse DT (Definition 3), we get approximate solution for the IVPs (48)–(50) on the interval $[0, t_1]$:

$$u_{1,I_1}(t) = 1 + t + \frac{1}{2}t^2 + \frac{2 + e^{-2}}{6}t^3 + \frac{4e^{-2} + 5}{72}t^4 + \frac{16e^{-2} - 5}{1080}t^5 + \dots, \quad (58)$$

$$u_{2,I_1}(t) = t^2 - \frac{2}{3}t^3 + \frac{2}{27}t^4 + \frac{2}{189}t^5 + \dots,$$

which is exactly the same approximate solution which has been obtained in [18].

The second step brings us to solving the given IVP on the interval $[t_1, t_2]$, where t_2 is the minimal root of $t - (1/2)e^{-t} = t_1$, $t_2 \approx 0,620556$. Now taking into account that both proportional delays $q_1 t = (1/3)t$ and $q_2 t = (1/2)t$ and also the time-dependent delay $t - \tau_1(t) = t - (1/2)e^{-t}$ map the interval $[t_1, t_2]$ into the interval $[0, t_1]$, system (2) becomes

$$u_1''' = e^{(t-2)} u_{1,I_1}\left(\frac{t}{3}\right) + \sqrt[3]{(u_1(t))^2} + u_{2,I_1}'\left(t - \frac{1}{2}e^{-t}\right), \quad (59)$$

$$u_2''' = \frac{1}{2}u_{2,I_1}''\left(\frac{t}{2}\right) + 2(t-1)u_{1,I_1}\left(\frac{t}{3}\right).$$

Denote $\mathcal{D}\{h(t)\}[t_1] = \{H_2(k)[t_1]\}_{k=0}^{\infty}$ and $\mathcal{D}\{f(t)\}[t_1] = \{F_2(k)[t_1]\}_{k=0}^{\infty}$, where $f(t) = u_2'(t - (1/2)e^{-t})$. By application of Theorem 2 to corresponding terms, system (2) transformed at $t_0 = t_1$ reads as

$$\begin{aligned} & (k+1)(k+2)(k+3)U_1(k+3)[t_1] \\ &= e^{-2} \sum_{l=0}^k \frac{e^{t_1}}{l!} \left(\frac{1}{3}\right)^{k-l} \sum_{x=0}^{\infty} \binom{x+k-l}{x} U_1(x+k-l)[0] \left(\frac{t_1}{3}\right)^x \\ &+ H_2(k)[t_1] + F_2(k)[t_1], \end{aligned} \quad (60)$$

$$\begin{aligned} & (k+1)(k+2)(k+3)U_2(k+3)[t_1] \\ &= \left(\frac{1}{2}\right)^{k+1} (k+3)(k+2)(k+1) \sum_{x=0}^{\infty} \binom{x+k+3}{x} \\ &\cdot U_2(x+k+3)[0] \left(\frac{t_1}{2}\right)^x + 2 \sum_{l=0}^k [(t_1-1)\delta(l) + \delta(l-1)] \\ &\cdot \left(\frac{1}{3}\right)^{k-l} \sum_{x=0}^{\infty} \binom{x+k-l}{x} U_1(x+k-l)[0] \left(\frac{t_1}{3}\right)^x. \end{aligned} \quad (61)$$

Now denote $\mathcal{D}\{g(t)\}[u_1(t_1)] = \{G_2(k)[u_1(t_1)]\}_{k=0}^{\infty}$; then, according to Theorem 3, $G_2(k)[u_1(t_1)] = \binom{2/3}{k} (u_1(t_1))^{2/3-k}$ for $k \geq 0$ and Theorem 1 implies

$$\begin{aligned} H_2(0)[t_1] &= G_2(0)[u_1(t_1)] = \sqrt[3]{u_1(t_1)^2}, \\ H_2(k)[t_1] &= \sum_{l=1}^k G_2(l)[u_1(t_1)] \cdot \widehat{B}_{k,l}(U_1(1)[t_1], \dots, \\ &U_1(k-l+1)[t_1]), \quad \text{for } k \geq 1. \end{aligned} \quad (62)$$

Further denote $e(t) = t - (1/2)e^{-t}$ and $\mathcal{D}\{e(t)\}[t_1] = \{E_2(k)[t_1]\}_{k=0}^{\infty}$. Then, $f(t) = (u_2' \circ e)(t)$ and, since $e(t_1) = 0$, Theorem 1 in combination with Lemma 1 yields

$$\begin{aligned} E_2(k)[t_1] &= t_1 \cdot \delta(k) + \delta(k-1) - \frac{1}{2} \cdot \frac{e^{-t_1}(-1)^k}{k!}, \quad k \geq 0, \\ F_2(0)[t_1] &= U_2(1)[0] = 0, \\ F_2(k)[t_1] &= \sum_{l=1}^k (l+1)U_2(l+1)[0] \cdot \widehat{B}_{k,l}(E_2(1)[t_1], \dots, \\ &E_2(k-l+1)[t_1]), \quad \text{for } k \geq 1. \end{aligned} \quad (63)$$

To get the initial data $U_1(k)[t_1]$ and $U_2(k)[t_1]$ for $k = 0, 1, 2$, we have to transform

$$u_i(t) = U_i(0)[0] + U_i(1)[0]t + U_i(2)[0]t^2 + U_i(3)[0]t^3 + \dots, \quad (64)$$

at t_1 , $i = 1, 2$. For $k = 0, 1, 2$, we have

$$\begin{aligned} U_i(k)[t_1] &= U_i(0)[0] + U_i(1)[0] \binom{1}{k} t_1^{1-k} \\ &+ U_i(2)[0] \binom{2}{k} t_1^{2-k} + U_i(3)[0] \binom{3}{k} t_1^{3-k} + \dots, \end{aligned} \quad (65)$$

i.e.,

$$\begin{aligned} U_i(0)[t_1] &= \sum_{k=0}^{\infty} U_i(k)[0] t_1^k, \\ U_i(1)[t_1] &= \sum_{k=0}^{\infty} (k+1)U_i(k+1)[0] t_1^k, \\ U_i(2)[t_1] &= \sum_{k=0}^{\infty} \binom{k+2}{2} U_i(k+2)[0] t_1^k. \end{aligned} \quad (66)$$

The initial values at t_1 will be approximated by taking finite sums in computer evaluations of the infinite sums above. Observe that $u_1(t_1) = U_1(0)[t_1]$.

Now let us compute the first few values of $H_2(k)[t_1]$. Denote $\widehat{B}_{k,l} = \widehat{B}_{k,l}(U_1(1)[t_1], \dots, U_1(k-l+1)[t_1])$. Then, the first values of $\widehat{B}_{k,l}$ are

$$\begin{aligned}
\widehat{B}_{0,0} &= 1, \\
\widehat{B}_{1,1} &= U_1(1)[t_1] \cdot \widehat{B}_{0,0} = U_1(1)[t_1], \\
\widehat{B}_{2,1} &= \frac{1}{2}U_1(1)[t_1] \cdot \widehat{B}_{1,0} + U_1(2)[t_1] \cdot \widehat{B}_{0,0} = U_1(2)[t_1], \\
\widehat{B}_{2,2} &= U_1(1)[t_1] \cdot \widehat{B}_{1,1} = (U_1(1)[t_1])^2,
\end{aligned} \tag{67}$$

and the coefficients H_2 for $k = 0, 1, 2$ are

$$\begin{aligned}
H_2(0)[t_1] &= u_1(t_1), \\
H_2(1)[t_1] &= \frac{2}{3}(u_1(t_1))^{-1/3} \cdot \widehat{B}_{1,1} = \frac{2}{3}(u_1(t_1))^{-1/3} \\
&\quad \cdot U_1(1)[t_1], \\
H_2(2)[t_1] &= \frac{2}{3}(u_1(t_1))^{-1/3} \cdot \widehat{B}_{2,1} - \frac{1}{9}(u_1(t_1))^{-4/3} \cdot \widehat{B}_{2,2} \\
&= \frac{2}{3}(u_1(t_1))^{-1/3} \cdot U_1(2)[t_1] - \frac{1}{9}(u_1(t_1))^{-4/3} \\
&\quad \cdot (U_1(1)[t_1])^2.
\end{aligned} \tag{68}$$

Let us turn our attention to the first few values of $F_2(k)[t_1]$. Starting with $E_2(k)[t_1]$,

$$\begin{aligned}
E_2(0)[t_1] &= t_1 - \frac{1}{2}e^{-t_1} = 0, \\
E_2(1)[t_1] &= 1 + \frac{1}{2}e^{-t_1} = 1 + t_1, \\
E_2(2)[t_1] &= -\frac{1}{2} \cdot \frac{e^{-t_1}}{2} = -\frac{t_1}{2}.
\end{aligned} \tag{69}$$

Now, $\widehat{B}_{k,l}$ are $\widehat{B}_{k,l}(E_2(1)[t_1], \dots, E_2(k-l+1)[t_1])$:

$$\begin{aligned}
\widehat{B}_{0,0} &= 1, \\
\widehat{B}_{1,1} &= E_2(1) \cdot \widehat{B}_{0,0} = 1 + t_1, \\
\widehat{B}_{2,1} &= \frac{1}{2}E_2(1) \cdot \widehat{B}_{1,0} + E_2(2) \cdot \widehat{B}_{0,0} = \frac{t_1}{2}, \\
\widehat{B}_{2,2} &= E_2(1) \cdot \widehat{B}_{1,1} = (1 + t_1)^2.
\end{aligned} \tag{70}$$

Finally, coefficients F_2 for $k = 0, 1, 2$ are

$$\begin{aligned}
F_2(0)[t_1] &= U_2(1)[0] = 0, \\
F_2(1)[t_1] &= 2U_2(2)[0] \cdot \widehat{B}_{1,1} = 2(1 + t_1), \\
F_2(2)[t_1] &= 2U_2(2)[0] \cdot \widehat{B}_{2,1} + 3U_2(3)[0] \cdot \widehat{B}_{2,2} \\
&= -t_1 - 2(1 + t_1)^2.
\end{aligned} \tag{71}$$

At this moment, we substitute H_2 and F_2 into systems (60) and (61). The next three coefficients at t_1 for U_1 are

$$\begin{aligned}
k=0: U_1(3)[t_1] &= \frac{1}{6}(e^{-2}e^{t_1} \cdot u_1(\frac{t_1}{3}) + u_1(t_1)), \\
k=1: U_1(4)[t_1] &= \frac{1}{24}(e^{-2}(e^{t_1} \cdot \frac{1}{3} \cdot \sum_{x=0}^{\infty} (x+1)U_1(x+1)[0](\frac{t_1}{3})^x \\
&\quad + e^{t_1} \cdot u_1(\frac{t_1}{3})) + \frac{2}{3} \cdot (u_1(t_1))^{-1/3} \cdot U_1(1)[t_1] \\
&\quad + 2(1+t_1)), \\
k=2: U_1(5)[t_1] &= \frac{1}{60}e^{-2}(e^{t_1}(\frac{1}{3})^2 \cdot \sum_{x=0}^{\infty} \binom{x+2}{x} U_1(x+2)[0] \\
&\quad \cdot (\frac{t_1}{3})^x + e^{t_1} \cdot \frac{1}{3} \cdot \sum_{x=0}^{\infty} (x+1)U_1(x+1)[0](\frac{t_1}{3})^x \\
&\quad + \frac{e^{t_1}}{2} \cdot u_1(\frac{t_1}{3})) + \frac{1}{60}(\frac{2}{3}(u_1(t_1))^{-1/3} \cdot (U_1(2)[t_1]) \\
&\quad - \frac{1}{9}(u_1(t_1))^{-4/3} \cdot (U_1(1)[t_1])^2 - t_1 - 2(1+t_1)^2),
\end{aligned} \tag{72}$$

and for U_2 , we obtain

$$\begin{aligned}
k=0: U_2(3)[t_1] &= \frac{1}{2}(\sum_{x=0}^{\infty} \binom{x+3}{x} U_2(x+3)[0](\frac{t_1}{2})^x) \\
&\quad + \frac{1}{3}(t_1 - 1) \cdot u_1(\frac{t_1}{3}), \\
k=1: U_2(4)[t_1] &= \frac{1}{4}(\sum_{x=0}^{\infty} \binom{x+4}{x} U_2(x+4)[0](\frac{t_1}{2})^x) \\
&\quad + \frac{1}{12}((t_1 - 1) \cdot \frac{1}{3} \cdot \sum_{x=0}^{\infty} (x+1)U_1(x+1)[0](\frac{t_1}{3})^x \\
&\quad + u_1(\frac{t_1}{3})), \\
k=2: U_2(5)[t_1] &= \frac{1}{8}(\sum_{x=0}^{\infty} \binom{x+5}{x} U_2(x+5)[0](\frac{t_1}{2})^x) \\
&\quad + \frac{1}{30}((t_1 - 1) \cdot \frac{1}{9} \cdot \sum_{x=0}^{\infty} \binom{x+2}{x} U_1(x+2)[0](\frac{t_1}{3})^x \\
&\quad + \frac{1}{3} \sum_{x=0}^{\infty} (x+1)U_1(x+1)[0](\frac{t_1}{3})^x).
\end{aligned} \tag{73}$$

Using the inverse DT, again we get approximate solution for the IVPs (48), (49), and (50) on the interval $[t_1, t_2]$:

TABLE 1: Example 1, error analysis of u in $[1, e]$.

	Exact solution	DT 5	DT 10	DT 25	Matlab DDENS
t	e^t	$ u - e^t $	$ u - e^t $	$ u - e^t $	$ u - e^t $
1	2.7183	0	0	0	0
1.4296	4.1769	0.0000	$6.4890E-12$	$8.8818E-16$	$1.6976E-5$
1.8591	6.4182	0.0017	$1.0381E-8$	$8.8818E-16$	$3.6095E-5$
2.2887	9.8622	0.0211	$1.2409E-6$	$3.5527E-15$	$9.3438E-5$
e	15.1543	0.1273	$3.0578E-5$	$1.2861E-12$	$1.4012E-4$

TABLE 2: Example 1, error analysis of u in $[e, e^e]$.

	Exact solution	DT 5	DT 10	DT 25	Matlab DDENS
t	e^t	$ u - e^t $			
e	15.1543	$3.5527E-15$	$3.5527E-15$	$3.5527E-15$	0
5.8273	339.4331	32.3491	1.333	$2.2737E-13$	0.0743
8.9363	$7.6028E+3$	$4.4755E+3$	397.5509	$2.2638E-5$	3.1741
12.0453	$1.7029E+5$	$1.5373E+5$	$5.6777E+4$	0.9751	112.2467
e^e	$3.8143E+6$	$3.7554E+6$	$2.6576E+6$	$2.0358E+3$	$3.1395E+3$

$$\begin{aligned}
u_{1,t_2}(t) &= U_1(0)[t_1] + U_1(1)[t_1](t - t_1) + U_1(2)[t_1] \\
&\quad \cdot (t - t_1)^2 + U_1(3)[t_1](t - t_1)^3 + \dots, \\
u_{2,t_2}(t) &= U_2(0)[t_1] + U_2(1)[t_1](t - t_1) + U_2(2)[t_1] \\
&\quad \cdot (t - t_1)^2 + U_2(3)[t_1](t - t_1)^3 + \dots
\end{aligned} \tag{74}$$

As the calculations are getting more complicated, all the calculations have been done numerically only.

3.4. Numerical Results and Discussion. Table 1 shows a comparison of results for Example 1 obtained by DT algorithm with the orders of Taylor polynomials of the approximate solution $N = 5, 10, 25$ to results of Matlab function DDENS in the interval $[1, e]$. Since the exact solution is known, absolute errors illustrate precision of each algorithm setting. All numbers are rounded to four decimal places. We see that DDENS performs satisfactory well and DT for $N = 10, 25$ does even better, whereas DT for $N = 5$ does not show satisfactory precision.

Table 2 brings the same comparison in the second interval $[e, e^e]$. We can observe a fast growth rate of the function values of the exact solution, which leads to the growth of absolute errors and loss of precision in all settings. It indicates that at the end of the considered interval $[e, e^e]$, the rate of precision would be better seen using relative errors.

Implementation of DT in Matlab in case of Example 2 produces numerical results which are listed in Table 3. The results of DT with order of the Taylor polynomial $N = 10$ are compared to values obtained by DT combined with modified Adomian formula in [18] and to values produced by Matlab function DDENS.

First, we should say that the function DDENS had difficulty at 0 where the value of the delayed argument $t/2$

was equal to the argument itself. Hence, to make DDENS work, we replaced $t/2$ by $t/2 - 10^{-16}$ in the second equation of (2). Our hypothesis is that the reason of the DDENS failure is a combination of two facts: the second equation is neutral with respect to a proportional delay and the interval where the problem is considered contains 0.

Second, we should mention that the numerical results for DDENS were obtained by looking for approximate solutions on the whole interval $[0, t_2]$. When trying to follow the method of steps, i.e., using DDENS on $[0, t_1]$ and then on $[t_1, t_2]$, the results on the second interval $[t_1, t_2]$ did not correspond to reality: there was a discontinuity in u_2 at t_1 .

Furthermore, we recall that the values taken from [18] have been computed using symbolic software Maple and the source code of the computation has been lost.

Now, we can see a very good concordance of all algorithms in numerical values of the second component u_2 , while we observe a growing distance between the values of the first component u_1 computed by presented DT algorithm and values computed by the other two algorithms. As u_1 has exponential characteristics, we interpret the growing distance as growing lack of precision of DT algorithm which is based on approximation by Taylor polynomials. We suppose that dividing the intervals $[0, t_1]$ and $[t_1, t_2]$ into smaller subintervals, i.e., refining the mesh grid, and applying the DT algorithm on those smaller intervals consecutively will improve the performance of the presented algorithm.

Although it seems that the algorithm used in [18] shows better performance than the one presented in this paper, we cannot claim it with certainty as the source code got lost and we are not able to reproduce the data. Moreover, the approach used in [18] involves calculations of symbolic derivatives which makes it difficult to implement in numerical software like Matlab.

TABLE 3: Example 2, comparison of presented DT, DT in [18] and Matlab function DDENSND for u_1 and u_2 in the first step for $t \in [0, t_1]$, and presented DT and DDENSND in the second step for $t \in [t_1, t_2]$.

Method	DT 10	DT 10	Matlab DDENSND	Matlab DDENSND	DT [18]	DT [18]
t	u_1	u_2	u_1	u_2	u_1	u_2
0.00	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
0.05	1.0513	0.0024	1.0513	0.0024	1.0513	0.0024
0.10	1.1054	0.0093	1.1050	0.0093	1.1051	0.0093
0.15	1.1625	0.0203	1.1614	0.0203	1.1618	0.0203
0.20	1.2230	0.0348	1.2204	0.0348	1.2209	0.0348
0.25	1.2871	0.0524	1.2822	0.0524	1.2832	0.0524
0.30	1.3552	0.0726	1.3469	0.0726	1.3481	0.0726
0.35	1.4277	0.0951	1.4146	0.0951	1.4160	0.0951
0.3520	1.4306	0.0960	1.4174	0.0960	—	—
0.3904	1.4896	0.1146	1.4717	0.1146	—	—
0.4289	1.5514	0.1340	1.5280	0.1340	—	—
0.4673	1.6161	0.1541	1.5864	0.1541	—	—
0.5057	1.6839	0.1747	1.6469	0.1748	—	—
0.5441	1.7548	0.1957	1.7098	0.1957	—	—
0.5826	1.8290	0.2169	1.7750	0.2169	—	—
0.6210	1.9065	0.2380	1.8428	0.2381	—	—

4. Conclusion

In the paper, we presented an algorithm which makes use of the differential transformation to initial value problems for systems of delayed or neutral differential equations with nonconstant delays. Two examples have been chosen to validate and test the algorithm. Numerical comparison of the presented semianalytical approach to Matlab function DDENSND brought interesting and promising results.

Example 1 showed expected and reliable behaviour of the differential transform in the first step of the method of steps and expected deviation in the numerical results from values of the exact solution in the second step. Furthermore, we could observe a good concordance between the presented algorithm and DDENSND.

After facing difficulties with DDENSND in Example 2, we could confirm a very good concordance of both differential transform and DDENSND in values of the component u_2 which has a polynomial character on the considered intervals. On the other hand, we observed a growing discrepancy between the two methods in values of the component u_1 which has an exponential character. Our conclusion is that the disagreement is caused by large lengths of the intervals where the approximate solution is computed using the differential transform and that refining the mesh grid is necessary to obtain better performance.

Further investigation will be focused on experimenting with different densities of mesh grids and studying convergence of the algorithm to find the optimal mesh grid. Numerical experiments will be focused on tuning the performance on problems with high complexity whose exact solutions are known and subsequently on applications to nonartificial real-life problems whose exact solutions are unknown.

Data Availability

The numerical data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The first author was supported by the project CEITEC 2020 (LQ1601) with financial support from the Ministry of Education, Youth and Sports of the Czech Republic under the National Sustainability Programme II, and by the grant FEKT-S-17-4225 of the Faculty of Electrical Engineering and Communication of Brno University of Technology. The Article Processing Charge was funded by Open Access Fund of Brno University of Technology. This support is gratefully acknowledged.

Supplementary Materials

We have included the commented Matlab source code for the first step of the method of steps in Example 1 as supplementary material. (*Supplementary Materials*)

References

- [1] T. Kalmar-Nagy, G. Stepan, and F. C. Moon, "Subcritical Hopf bifurcation in the delay equation model for machine tool vibrations," *Nonlinear Dynamics*, vol. 26, pp. 121–142, 2011.
- [2] D. Adak, N. Bairagi, and R. Hakl, "Chaos in delay-induced Leslie-Gower prey-predator-parasite model and its control through prey harvesting," *Nonlinear Analysis: Real World Applications*, vol. 51, pp. 102998–103020, 2020.
- [3] A. Matsumoto and F. Szidarovszky, "Delay dynamics of a Cournot game with heterogeneous duopolies," *Applied Mathematics and Computation*, vol. 269, pp. 699–713, 2015.
- [4] V. Kolmanovskii and A. Myshkis, *Introduction to the Theory and Applications of Functional Differential Equations*, Kluwer, Dordrecht, Netherlands, 1999.

- [5] I. Györi and G. Ladas, *Oscillation Theory of Delay Differential Equations with Applications*, Clarendon Press, Oxford, UK, 1991.
- [6] J. K. Hale and S. M. Verduyn Lunel, *Introduction to Functional Differential Equations*, Springer, New York, NY, USA, 1993.
- [7] J. Rebenda and Z. Šmarda, “Comparison of differential transformation method with adomian decomposition method for functional differential equations with proportional delays,” in *Proceedings of the 11th International Conference on Numerical Analysis and Applied Mathematics (ICNAAM 2013)*, T. E. Simos, Ed., vol. 1558, pp. 1190–1193, AIP Publishing, Melville, NY, USA, 2013.
- [8] H. Šamajová and T. Li, “Oscillators near Hopf bifurcation,” *Communications-Scientific Letters of the University of Žilina*, vol. 17, pp. 83–87, 2015.
- [9] J. Rebenda, Z. Šmarda, and Y. Khan, “A new semi-analytical approach for numerical solving of Cauchy problem for differential equations with delay,” *Filomat*, vol. 31, no. 15, pp. 4725–4733, 2017.
- [10] H. Šamajová, “Semi-analytical approach to initial problems for systems of nonlinear partial differential equations with constant delay,” in *Proceedings of the EQUADIFF 2017 Conference*, K. Mikula, D. Sevcovic, and J. Urban, Eds., pp. 163–172, Spektrum STU Publishing, Bratislava, Slovakia, 2017.
- [11] J. Rebenda and Z. Šmarda, “A numerical approach for solving of fractional Emden-Fowler type equations,” in *Proceedings of the International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2017)*, T. E. Simos, Ed., vol. 1978, p. 140006, AIP Publishing, Melville, NY, USA, 2018.
- [12] T. Krisztin, “Analyticity of solutions of differential equations with a threshold delay,” in *Springer Proceedings in Mathematics & Statistics, Recent Advances in Delay Differential and Difference Equations*, F. Hartung and M. Pituk, Eds., vol. 94, pp. 173–180, Springer, Basel, Switzerland, 2014.
- [13] J. Duarte, C. Janeiro, and N. Martins, “Analytical solutions of an economic model by the homotopy analysis method,” *Applied Mathematical Sciences*, vol. 10, no. 49, pp. 2483–2490, 2016.
- [14] Z. Šmarda, J. Diblík, and Y. Khan, “Extension of the differential transformation method to nonlinear differential and integro-differential equations with proportional delays,” *Advances in Difference Equations*, vol. 69, pp. 1–13, 2013.
- [15] J. Rebenda and Z. Šmarda, “A differential transformation approach for solving functional differential equations with multiple delays,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 48, pp. 246–257, 2017.
- [16] A. Bellen and M. Zennaro, *Numerical Methods for Delay Differential Equations*, Oxford University Press, Oxford, UK, 2003.
- [17] P. G. Warne, D. A. P. Warne, J. S. Sochacki, G. E. Parker, and D. C. Carothers, “Explicit *A-Priori* error bounds and adaptive error control for approximation of nonlinear initial value differential systems,” *Computers & Mathematics with Applications*, vol. 52, no. 12, pp. 1695–1710, 2006.
- [18] J. Rebenda and Z. Šmarda, “Numerical algorithm for nonlinear delayed differential systems of n th order,” *Advances in Difference Equations*, vol. 2019, no. 26, pp. 1–13, 2019.
- [19] E. N. Petropoulou, P. D. Sifarakas, and E. E. Tzirtzilakis, “A “discretization” technique for the solution of ODEs,” *Journal of Mathematical Analysis and Applications*, vol. 331, no. 1, pp. 279–296, 2007.
- [20] E. N. Petropoulou, P. D. Sifarakas, and E. E. Tzirtzilakis, “A “discretization” technique for the solution of ODEs II,” *Numerical Functional Analysis and Optimization*, vol. 30, no. 5-6, pp. 613–631, 2009.
- [21] J. Rebenda, “An application of Bell polynomials in numerical solving of nonlinear differential equations,” in *Proceedings of the 17th Conference on Applied Mathematics, APLIMAT 2018*, pp. 891–900, Slovak University of Technology, SPEKTRUM STU, Bratislava, Slovakia, 2018.