Hindawi
Journal of Mathematics
Volume 2023, Article ID 3620870, 13 pages
https://doi.org/10.1155/2023/3620870

Research Article

The Differential Transform Method as an Effective Tool to Solve Implicit Hessenberg Index-3 Differential-Algebraic Equations

Brahim Benhammouda

Higher Colleges of Technology, Abu Dhabi Men's College, P.O. Box 25035, Abu Dhabi, UAE

Correspondence should be addressed to Brahim Benhammouda; bbenhammouda@hct.ac.ae

Received 25 October 2022; Revised 28 November 2022; Accepted 6 December 2022; Published 23 January 2023

Academic Editor: Predrag S. Stanimirović

Differential-algebraic equations (DAEs) are important tools to model complex problems in various application fields easily. Those DAEs with an index-3, even the linear ones, are known to cause problems when solving them numerically. The present article proposes a new algorithm together with its multistage form to efficiently solve a class of nonlinear implicit Hessenberg index-3 DAEs. This algorithm is based on the idea of applying the differential transform method (DTM) directly to the DAE without applying the traditional index reduction methods, which can be complex and often result in violations of the DAE constraints. Also, to deal with the nonlinear terms in the DAE, we approximate them using the Adomian polynomials. This new idea has given us a simple and efficient algorithm, which involves the solution of linear algebraic systems except for the initial recursion terms. This algorithm is easy to implement in Maple or Mathematica. Furthermore, to enlarge the interval of convergence of the power series solution obtained from the DTM, an algorithm for the multistage DTM is also given. Both algorithms are applied to solve two examples of highly nonlinear implicit index-3 Hessenberg DAEs. Numerical results show that the DTM can determine the exact solution in convergent power series, while the multistage DTM can compute accurate numerical solutions over large intervals.

1. Introduction

Differential-algebraic equations (DAEs) are systems formed by differential and algebraic equations. Simulations of models from various fields such as mechanical systems [1–3], vehicle industry [4, 5], bio-mechanics [6], machinery [7], aircraft landing gears [8], towed rockets [9], and unmanned aerial vehicles (UAVs) [10, 11] often lead to DAEs. DAEs are classified by a natural number called the index which has several definitions. In this article we use the differential index which is defined as the minimum number of differentiations required to transform a DAE to an ordinary differential equation [12]. Roughly speaking, the index of a DAE measures the difficulty this DAE can pose. If the index is greater or equal to two, we say that the DAE is a higher index DAE. The solution techniques for higher index DAEs, even the linear ones, are known to face numerical and analytical difficulties that are not encountered when solving ordinary differential equations [13–16]. As a consequence, many numerical methods were developed to solve higher index DAEs such as range-Kutta [13, 17], projected Taylor series methods [18], hybrid block algorithms [19], stabilization methods [20–22], augmented Lagrangian method [23], sequential regularization methods [24], and the differential transform method (DTM) [25, 26]. For the solution of DAEs, one can also find in the literature the power series method combined with the Adomian polynomials [27, 28] and other methods [29–33]. A very popular approach to treat higher index DAEs is first to reduce the index by differentiating the constraints one or more times with respect to time to obtain an ordinary differential system or an index-1 DAE. Then apply numerical integration methods to the index reduced DAE. However, the main difficulty with this index reduction technique is that the computed numerical solution of the resulting DAE may no longer fulfill the constraints of the original DAE. Solving this index reduced system will result in constraints violation in general and therefore gives a solution with no physical meaning. To resolve this problem when solving higher index DAEs in Lagrangian formulation form, some methods such as stabilization or augmented Lagrangian formulations were proposed to control the constraint violations during the computation of the numerical solution [20–22]. The most used method of stabilization is that of Baumgarte [20], but its drawback is the way of choosing its feedback parameters. The augmented Lagrangian formulation proposed in [23] has the same problem of the parameter selection. The challenge is then to construct efficient numerical methods that provide accurate solutions for index-3 DAEs while preserving all the constraints of the original DAE.

In this article, we propose a new efficient algorithm to solve the following class of nonlinear implicit Hessenberg index-3 DAEs:

$$\begin{cases} u' = v, \\ f(v', v, u, \lambda) = 0, \quad t \in [0, T], \\ g(u) = 0, \end{cases} \tag{1}$$

where the dash $(')$ denotes the time derivative, $u(t) \in \mathbb{R}^{n_u}$, $v(t) \in \mathbb{R}^{n_u}$, and $\lambda(t) \in \mathbb{R}^{n_\lambda}$. The vector function f is such that $f\colon \mathbb{R}^{n_u} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\lambda} \longrightarrow \mathbb{R}^{n_u}$. The vector function $g\colon \mathbb{R}^{n_u} \longrightarrow \mathbb{R}^{n_\lambda}$, $(n_\lambda \le n_u)$ defines the apparent constraints of the DAE where $G(u) = g_u'(u) \in \mathbb{R}^{n_\lambda \times n_u}$ denotes the Jacobian of g.

The following initial conditions must be provided to solve DAE (1) uniquely:

$$\begin{aligned} u(0) &= \alpha, \\ v(0) &= \beta. \end{aligned} \tag{2}$$

It should be noted that the initial conditions α and β must be chosen consistent, that is to fulfill the following algebraic system:

$$g(u) = 0, \tag{3}$$

$$G(u)v = 0. \tag{4}$$

Note here that we do not need to provide an initial condition for the unknown vector $\lambda(t)$ as the initial value $\lambda(0)$ is already fixed by the initial condition (2) and the hidden constraints in the DAE (1). Throughout this paper, we assume the functions $f(w, v, u, \lambda)$ and $g(u)$ to be analytical. In addition, the Jacobian matrix $A \coloneqq f_w'(w, v, u, \lambda) \in \mathbb{R}^{n_u \times n_u}$ is assumed to be nonsingular on $\operatorname{Ker} G(u)$. We assume further that the square matrix $GA^{-1}B \in \mathbb{R}^{n_\lambda \times n_\lambda}$ is nonsingular, where $B \coloneqq f_\lambda'(w, v, u, \lambda) \in \mathbb{R}^{n_u \times n_\lambda}$, which means that DAE (1) has index-3. To see this, we differentiate (3) with respect to time to get equation (4). Then, we differentiate this equation once with respect to time to get

$$G(u)v' + G_u(u)(v, v) = 0. \tag{5}$$

The set of equations formed by the second equation of (1) and (5) is uniquely solvable for the variables v' and λ since its Jacobian

$$\begin{pmatrix} A & B \\ G & 0 \end{pmatrix} \in \mathbb{R}^{(n_u + n_\lambda) \times (n_u + n_\lambda)}, \tag{6}$$

is nonsingular due to the assumption that the matrix $GA^{-1}B$ is nonsingular. Therefore,

$$v' = p(v, u, \lambda), \tag{7}$$

$$\lambda = q(v, u), \tag{8}$$

where p and q are some functions. Now, differentiating (8) once with respect to time, we get an ordinary differential system

$$\lambda' = q_v'(v, u)p(v, u, \lambda) + q_u'(v, u)v. \tag{9}$$

This can be written in the form

$$\lambda' = s(v, u, \lambda). \tag{10}$$

Hence, differentiation of three times has led to an ordinary differential system for the algebraic variable λ and thus DAE (1) has an index-3.

In this manuscript, we present a new algorithm together with its multistage form to efficiently solve the class of implicit Hessenberg index-3 DAEs (1). This algorithm is based on an effective combination of the DTM and the Adomian polynomials [28, 34–39]. The main idea in this paper is to apply the DTM directly to DAE (1) without using index reduction methods which are complex and computationally expensive. These methods also often result in constraints violations. Furthermore, to approximate the nonlinear terms f and g, we expand them in power series using the Adomian polynomials. This hybrid technique provides the exact solution in convergent power series. To extend the interval of convergence of the DTM solution, a multistage DTM (MsDTM) algorithm is also given. The MsDTM has a simple algorithm, which can be easily coded in Maple or Mathematica. To demonstrate the effectiveness and accuracy of the DTM algorithm and that of its multistage form MsDTM, two highly nonlinear implicit index-3 Hessenberg DAEs are solved by coding these algorithms in Maple 15. The numerical results show that the DTM provides the solution in convergent power series while the MsDTM gives accurate approximate solutions over large intervals.

The organisation of this manuscript is as follows: in Section 2, we review the Adomian polynomials and the DTM to solve ordinary differential equations. Next, in Section 3, we present the new proposed algorithms to solve the implicit Hessenberg index-3 DAE initial-value problem (1) and (2). Then, in Section 4, two examples of highly nonlinear implicit index-3 Hessenberg DAEs are solved to illustrate the accuracy of this new method. Finally, a conclusion is given in Section 5.

2. Adomian Polynomials and the Differential Transform Method

In this section, we give a brief review to the Adomian polynomials [34–39], which are useful in the expansion of the nonlinear terms f and g of the DAE (1), then we review the differential transform method. Usually, a nonlinear term $h(u)$ in a differential equation is decomposed as follows:

$$h(u) = \sum_{k=0}^{\infty} h_k(u_0, u_1, \ldots, u_k), \tag{11}$$

where the Adomian polynomials h_k are computed, for all nonlinearities, from

$$\begin{aligned} h_k &:= h_k(u_0, u_1, \ldots, u_k) \\ &= \frac{1}{k!}\frac{d^k}{d\xi^k}\left(h\left(\sum_{i=0}^{\infty}\xi^i u_i\right)\right)_{\xi=0}, \\ k &= 0, 1, 2, \ldots. \end{aligned} \tag{12}$$

Here, $u_k, k = 0, 1, 2, \ldots$, are the terms used in the expansion

$$u(t) = \sum_{k=0}^{\infty} u_k. \tag{13}$$

Making use of (12), we can compute the first Adomian polynomials:

$$\begin{aligned} h_0 &= h(u_0), \\ h_1 &= u_1 h^{(1)}, \\ h_2 &= u_2 h^{(1)} + \frac{u_1^2}{2} h^{(2)}, \\ h_3 &= u_3 h^{(1)} + u_1 u_2 h^{(2)} + \frac{u_1^3}{3!} h^{(3)}, \\ h_4 &= u_4 h^{(1)} + \left(\frac{u_2^2}{2} + u_1 u_3\right) h^{(2)} + \frac{u_1^2 u_2}{2} h^{(3)} + \frac{u_1^4}{4!} h^{(4)}, \\ h_5 &= u_5 h^{(1)} + (u_2 u_3 + u_1 u_4) h^{(2)} + \left(\frac{u_1 u_2^2}{2} + \frac{u_1^2 u_3}{2}\right) h^{(3)} + \frac{u_1^3 u_2}{3!} h^{(4)} + \frac{u_1^5}{5!} h^{(5)}, \end{aligned} \tag{14}$$

where $h^{(k)} := d^k h(u_0)/du_0^k, k = 0, 1, 2, \ldots$

For a function $h(u, v)$ with two variables, we have the following first few Adomian polynomials:

$$
\begin{aligned}
h_0 &= h(u_0, v_0),\\
h_1 &= u_1 h^{(1,0)} + v_1 h^{(0,1)},\\
h_2 &= u_2 h^{(1,0)} + v_2 h^{(0,1)} + \frac{u_1^2}{2} h^{(2,0)} + \frac{v_1^2}{2} h^{(0,2)} + u_1 v_1 h^{(1,1)},\\
h_3 &= u_3 h^{(1,0)} + v_3 h^{(0,1)} + u_1 u_2 h^{(2,0)} + v_1 v_2 h^{(0,2)} + (u_1 v_2 + u_2 v_1) h^{(1,1)} + \frac{u_1^3}{6} h^{(3,0)} + \frac{v_1^3}{6} h^{(0,3)} + \frac{u_1^2 v_1}{2} h^{(2,1)} + \frac{u_1 v_1^2}{2} h^{(1,2)},\\
h_4 &= u_4 h^{(1,0)} + v_4 h^{(0,1)} + \left(u_1 u_3 + \frac{u_2^2}{2}\right) h^{(2,0)} + \left(v_1 v_3 + \frac{v_2^2}{2}\right) h^{(0,2)} + (u_1 v_3 + u_2 v_2 + u_3 v_1) h^{(1,1)} + \frac{u_1^2 u_2}{2} h^{(3,0)} + \frac{v_1^2 v_2}{2} h^{(0,3)}\\
&\quad + \left(u_1 u_2 v_1 + \frac{u_1^2 v_2}{2}\right) h^{(2,1)} + \left(u_1 v_1 v_2 + \frac{u_2 v_1^2}{2}\right) h^{(1,2)} + \frac{u_1^4}{4!} h^{(4,0)} + \frac{u_1^3 v_1}{6} h^{(3,1)} + \frac{u_1^2 v_1^2}{4} h^{(2,2)} + \frac{u_1 v_1^3}{4} h^{(1,3)} + \frac{v_1^4}{4!} h^{(0,4)},
\end{aligned}
\tag{15}
$$

where $h^{(k,l)} := \partial^{k+l} h(u_0, v_0)/\partial u_0^k \partial v_0^l, k = 0, 1, 2, \ldots, l = 0, 1, 2, \ldots$

The differential transform of a function $u(t)$ is defined by

$$
u_k = \frac{1}{k!}\left[\frac{d^k u(t)}{dt^k}\right]_{t=0}, \quad k = 0, 1, 2, \ldots \tag{16}
$$

and the inverse differential transform of $u_k, k = 0, 1, 2, \ldots$ is given by

$$
u(t) = \sum_{k=0}^{\infty} u_k t^k. \tag{17}
$$

From (16) and (17), we have

$$
u(t) = \sum_{k=0}^{\infty} \frac{1}{k!}\left[\frac{d^k u(t)}{dt^k}\right]_{t=0} t^k. \tag{18}
$$

An approximate solution is given by

$$
u(t) = \sum_{k=0}^{K} u_k t^k, \tag{19}
$$

where K is the number of terms in the approximation.

If $u(t)$ is expanded as (17), then the nonlinear term $h(u)$ can be expanded using the Adomian polynomials as [40]

$$
h\left(\sum_{k=0}^{\infty} u_k t^k\right) = \sum_{k=0}^{\infty} h_k(u_0, u_1, \ldots, u_k) t^k, \tag{20}
$$

where now $u_k, k = 0, 1, 2, \ldots$ are the coefficients of expansion (19). One important property of the Adomian polynomials which we will make use of to develop our algorithms is the fact that the Adomian polynomials h_k, $k = 1, 2, 3, \ldots$ are affine with respect to u_k.

3. The New Algorithm

The numerical solution of index-3 DAEs, including the linear ones, is known to be hard to obtain. In this section, we present a new algorithm for solving the class (1) of nonlinear implicit Hessenberg index-3 DAEs. The proposed algorithm is based on an effective combination of the differential transform method (DTM) with the Adomian polynomials [34–39]. The main idea of our technique is to apply the DTM directly to this class of DAEs, where the nonlinear terms are expanded in power series using the Adomian polynomials [40]. Then, by using the fact that the Adomian polynomial h_k is affine with respect to u_k, a linear algebraic recursion system for the differential transforms of the solution is obtained. Next, since the DAE (1) is an index-3 DAE, then the derived algebraic system is shown to have a unique solution. The main advantage of our technique is that it does not need to reduce the index of the DAE (1) before applying the DTM. This has given us a simple and efficient algorithm that can be easily coded in Maple or Mathematica. The following new theorems are important for derivation of our technique.

Theorem 1. *Consider the fully implicit ordinary differential system $f(u', u) = 0$, with the initial condition $u(0) = u_0$ where the Jacobian $f_{u'}(u', u)$ is nonsingular. Assume that the function f is analytical. Then, the differential transform of u' is given by the recursion formula $(k+1) f_{u'}(u_1, u_0) u_{k+1} = -f_k(u_1, \ldots, k u_k, 0, u_0, \ldots, u_k), k = 1, 2, 3, \ldots,$ with $u_1 = u'(0)$ computed from $f(u'(0), u_0) = 0$. Here $f_k = f_k(u_1, \ldots, k u_k, (k+1) u_{k+1}, u_0, \ldots, u_k), k = 1, 2, 3, \ldots$ is the vector of k-th two variable Adomian polynomials of the components of the vector function f.*

Proof. Assume that the solution $u(t)$ can be expanded as follows:

$$u(t) = \sum_{k=0}^{\infty} u_k t^k. \tag{21}$$

Then, we expand the nonlinear term $f(u', u)$ in terms of the Adomian polynomials as follows:

$$f\left(\sum_{k=0}^{\infty}(k+1)u_{k+1}t^k, \sum_{k=0}^{\infty} u_k t^k\right) = \sum_{k=0}^{\infty} f_k t^k = 0. \tag{22}$$

This gives

$$f_k(u_1, \ldots, ku_k, (k+1)u_{k+1}, u_0, \ldots, u_k) = 0, \quad k = 1, 2, 3, \ldots \tag{23}$$

Or,

$$(k+1) f_{u'}(u_1, u_0)u_{k+1} + f_k(u_1, \ldots, ku_k, 0, u_0, \ldots, u_k), k = 1, 2, 3, \ldots \tag{24}$$

This leads to the following recursion formula to compute the differential transform of the derivative of u:

$$(k+1) f_{u'}(u_1, u_0)u_{k+1} = -f_k(u_1, \ldots, ku_k, 0, u_0, \ldots, u_k), \quad k = 1, 2, 3, \ldots \tag{25}$$

Note here that the right hand side of the above formula does not depend on $u_{k+1}, k = 1, 2, 3, \ldots$

In a similar manner, one can show the following theorem. □

Theorem 2. *Consider the algebraic system $g(u) = 0$ where we assume that the function g is analytical. Then, the differential transform of u satisfies the following recursion formula $g_u(u_0)u_k = -g_k(u_0, \ldots, u_{k-1}, 0), k = 1, 2, 3, \ldots$ Here, $g_k = g_k(u_0, \ldots, u_{k-1}, u_k), k = 1, 2, 3, \ldots$ is the vector of k-th Adomian polynomials of the components of the vector function g.*

To derive our algorithm, we start by writing the second equation of DAE (1) as follows:

$$v' = w, \tag{26}$$

and

$$f(w, v, u, \lambda) = 0. \tag{27}$$

Then, we expand $w(t), u(t), v(t)$, and $\lambda(t)$ in terms of their differential transforms as follows:

$$\begin{aligned} w(t) &= \sum_{k=0}^{\infty} w_k t^k, \\ u(t) &= \sum_{k=0}^{\infty} u_k t^k, \\ v(t) &= \sum_{k=0}^{\infty} v_k t^k, \\ \lambda(t) &= \sum_{k=0}^{\infty} \lambda_k t^k, \end{aligned} \tag{28}$$

where the unknown vectors $w_k \in \mathbb{R}^{n_u}, u_k \in \mathbb{R}^{n_u}, v_k \in \mathbb{R}^{n_u}$ and $\lambda_k \in \mathbb{R}^{n_\lambda}, k = 0, 1, 2, \ldots$, will be determined later by our algorithm. Then, we expand the nonlinear term $f(w, v, u, \lambda)$ of (27) in terms of the Adomian polynomials as follows:

$$f\left(\sum_{k=0}^{\infty} w_k t^k, \sum_{k=0}^{\infty} v_k t^k, \sum_{k=0}^{\infty} u_k t^k, \sum_{k=0}^{\infty} \lambda_k t^k\right) = \sum_{k=0}^{\infty} f_k t^k, \tag{29}$$

where $f_k = f_k(w_0, \ldots, w_k, v_0, \ldots, v_k, u_0, \ldots, u_k, \lambda_0, \ldots, \lambda_k)$, $k = 0, 1, 2, \ldots$ denotes the vector of the Adomian polynomials of the components of f. From the first equation of DAE (1) and equation (26), we obtain the following recursions for the differential transforms of the solution components $u(t), v(t)$, and $w(t)$:

$$v_{k-1} = ku_k, \quad k = 1, 2, 3, \ldots, \tag{30}$$

$$w_{k-1} = kv_k, \quad k = 1, 2, 3, \ldots, \tag{31}$$

from which, we have

$$k(k-1)u_k = w_{k-2}, \quad k = 2, 3, 4, \ldots \tag{32}$$

Now, since $f_k = 0$, then using Theorem 1, we obtain

$$A_0 w_k + B_0 \lambda_k = -r_k, \quad k = 1, 2, 3, \ldots, \tag{33}$$

where $r_k = f_k(w_0, \ldots, w_{k-1}, 0, v_0, \ldots, v_k, u_0, \ldots, u_k, \lambda_0, \ldots, \lambda_{k-1}, 0)$, $A_0 = f'_w(w_0, v_0, u_0, \lambda_0) \in \mathbb{R}^{n_u \times n_u}$, and $B_0 = f'_\lambda(w_0, v_0, u_0, \lambda_0) \in \mathbb{R}^{n_u \times n_\lambda}$. Note here that r_k depends only on the recursion terms $w_0, \ldots, w_{k-1}, v_0, \ldots, v_k$, $u_0, \ldots, u_k$ and $\lambda_0, \ldots, \lambda_{k-1}$. System (33) represents a set of n_u linear algebraic equations with $(n_u + n_\lambda)$ variables. To solve it, we need to complete it by another n_λ linear algebraic equations. To do this, we approximate the nonlinear term $g(u)$ of equation (1) using the Adomian polynomials as follows:

$$g\left(\sum_{k=0}^{\infty} u_k t\right) = \sum_{k=0}^{\infty} g_k t^k, \tag{34}$$

where $g_k \coloneqq g_k(u_0, \ldots, u_k), k = 0, 1, 2, \ldots$, denotes the vector of the Adomian polynomials of the components of the vector g.

Next substituting the expansion of g from (34) in the last equation of DAE (1), we deduce that

$$\begin{aligned} g_k &= 0, \\ k &= 0, 1, 2, \ldots, \end{aligned} \tag{35}$$

which leads, using Theorem 2, to the following algebraic linear system for the unknown vector u_k:

$$\begin{aligned} G_0 u_k &= -s_k, \\ k &= 1, 2, 3, \ldots, \end{aligned} \tag{36}$$

where $s_k = g_k(u_0, \ldots, u_{k-1}, 0)$ and $G_0 = g_u'(u_0) \in \mathbb{R}^{n_\lambda \times n_u}$. It should be noted here that s_k depends only on the recursion terms $u_0, \ldots, u_{k-1}$. Before solving the linear algebraic system formed by (33) and (36), we first calculate the initial recursion terms u_0 and v_0 from the initial conditions $u_0 = u(0)$ and $v_0 = v(0)$. Furthermore, setting $k = 1$ in (30), we obtain $u_1 = v_0$. Now, we need to determine the consistent initial value $\lambda(0) = \lambda_0$. Using the second equation of DAE (1) and equations (30) and (34) for $k = 2$, we obtain the $(n_u + n_\lambda) \times (n_u + n_\lambda)$ algebraic system for the unknown vectors w_0 and λ_0:

$$\begin{cases} f(w_0, v_0, u_0, \lambda_0) = 0, \\ G_0 w_0 = -2 s_2. \end{cases} \tag{37}$$

This algebraic system is nonlinear when f is nonlinear with respect to the unknown vectors w_0 and/or λ_0. Since the DAE (1) is index-3, then the square matrix $G_0 A_0^{-1} B_0 \in \mathbb{R}^{n_\lambda \times n_\lambda}$ is nonsingular. Therefore, the Jacobian (38) of system (37) is nonsingular, and hence, system (37) is uniquely solvable for w_0 and λ_0:

$$\begin{pmatrix} A_0 & B_0 \\ G_0 & 0 \end{pmatrix} \in \mathbb{R}^{(n_u + n_\lambda) \times (n_u + n_\lambda)}. \tag{38}$$

Once the unknown w_0 is determined, we can then find the values of $u_2 = 0.5 w_0$ and $v_1 = w_0$ by setting $k = 2$ in equations (30) and (31), respectively.

Next, we need to determine the unknown vectors u_k and λ_{k-2} for $k \geq 3$. Using (32), system (33) leads to

$$k(k-1) A_0 u_k + B_0 \lambda_{k-2} = -r_{k-2}, k \geq 3. \tag{39}$$

Finally, we can determine the unknown vectors u_k and λ_{k-2} by solving the linear algebraic system formed by (36) and (39), for $k \geq 3$. This linear algebraic system is uniquely solvable since its coefficient matrix

$$\begin{pmatrix} k(k-1) A_0 & B_0 \\ G_0 & 0 \end{pmatrix} \in \mathbb{R}^{(n_u + n_\lambda) \times (n_u + n_\lambda)}, \tag{40}$$

is nonsingular due to the fact that the matrix $G_0 A_0^{-1} B_0 \in \mathbb{R}^{n_\lambda \times n_\lambda}$ is nonsingular. To solve the system formed by the (36) and (39), we multiply system (39) from the left by the matrix $G_0 A_0^{-1}$ and then use system (35) to obtain the following linear algebraic system for the unknown vector λ_{k-2}:

$$\left(G_0 A_0^{-1} B_0\right) \lambda_{k-2} = -G_0 A_0^{-1} r_{k-2} + k(k-1) s_k, k \geq 3. \tag{41}$$

From system (41), we determine the unknown vector λ_{k-2} uniquely for $k \geq 3$. Substituting the expression of λ_{k-2} obtained from (41) into system (39), we can calculate the unknown vector u_k, for $k \geq 3$ as follows:

$$k(k-1) A_0 u_k = -r_{k-2} + B_0 \left(G_0 A_0^{-1} B_0\right)^{-1} \left(G_0 A_0^{-1} r_{k-2} - k(k-1) s_k\right), k \geq 3. \tag{42}$$

Once we have computed the unknown u_k, we use system (28) to find the unknown vector v_k, for $k = 2, 3, 4, \ldots$ Finally, we obtain an approximate solution for the implicit Hessenberg index-3 DAE initial-value problems (1) and (2) as follows:

$$\begin{aligned} u(t) &= \sum_{k=0}^{K} t^k u_k, \\ v(t) &= \sum_{k=0}^{K-1} t^k v_k, \\ \lambda(t) &= \sum_{k=0}^{K-2} t^k \lambda_k, \end{aligned} \tag{43}$$

where K is the order of approximation. This completes the solution process.

The above process of the DTM for solving the DAE initial-value problems (1) and (2) is given by Algorithm 1.

We can extend the interval of convergence of the solution by using a multistage DTM (MsDTM). The procedure of the MsDTM is described in Algorithm 2.

4. Numerical Examples

In this section, two examples of highly nonlinear implicit DAEs (1) and (2) are solved to demonstrate the effectiveness and accuracy of the presented technique. The implementation of our algorithms was performed in Maple 15. For each example, we first apply the algorithm of the standard DTM given in Section 3 to compute the solution in power series. Then, we apply the multistage DTM algorithm given in Section 3 to compute a numerical solution over large intervals.

Example 1. In this first example, we consider the DAE (1) with the vector function f given by

$$f(v', v, u, \lambda) = \begin{pmatrix} 1 + 4\cos u_1 & 1 + \cos u_1 \\ 1 + \cos u_1 & 1 - \cos u_1 \end{pmatrix} \begin{pmatrix} v_1'^3 + v_1' + u_1^3 + u_1 \\ v_2'^3 + v_2' + u_2^3 + u_2 \end{pmatrix} - \begin{pmatrix} \left(v_1^3 - v_2 - \lambda^3 - 2\lambda\right)\left(\cos u_1 + \cos(u_1 + u_2)\right) \\ \left(v_1^3 - v_2 - \lambda^3 - 2\lambda\right)\cos(u_1 + u_2) \end{pmatrix}, \quad (44)$$

and where the function g defining the constraint is given by

$$g(u) = \sin u_1 + \sin(u_1 + u_2), \quad (45)$$

where $u = (u_1, u_2)^T \in \mathbb{R}^2$, $v = (v_1, v_2)^T \in \mathbb{R}^2$, and $\lambda \in \mathbb{R}$.

The Jacobian of g is $G(u) = g_u'(u) = (\cos u_1 + \cos(u_1 + u_2),\ \cos(u_1 + u_2))$. We can easily check that the exact solution of the DAE of this example is $u(t) = (\sin t, -2\sin t)^T$, $v(t) = (\cos t, -2\cos t)^T$, and $\lambda(t) = \cos t$. This exact solution is obtained from the initial conditions

$$\begin{aligned} u(0) &= (0, 0)^T, \\ v(0) &= (1, -2)^T. \end{aligned} \quad (46)$$

It is easy to check that the above initial conditions are consistent, since they fulfill $g(u(0)) = 0$ and $g_u'(u(0))v(0) = 0$. The DAE corresponding to this example has an index-3, hence difficult to treat even numerically. To compute the solution in power series, we apply the algorithm of the standard DTM given in Section 3. This DTM algorithm was coded in Maple 15. By choosing the order of approximation $K = 10$, our DTM algorithm has computed the following power series approximations:

$$\begin{aligned} u(t) &= t\begin{pmatrix} 1 \\ -2 \end{pmatrix} - t^3\begin{pmatrix} \frac{1}{6}; \\ \frac{-1}{3} \end{pmatrix} + t^5\begin{pmatrix} \frac{1}{120} \\ \frac{-1}{60} \end{pmatrix} - t^7\begin{pmatrix} \frac{1}{5040} \\ \frac{-1}{2520} \end{pmatrix} + t^9\begin{pmatrix} \frac{1}{362880} \\ \frac{-1}{181440} \end{pmatrix}, \\ v(t) &= \begin{pmatrix} 1 \\ -2 \end{pmatrix} - t^2\begin{pmatrix} \frac{1}{2} \\ -1 \end{pmatrix} + t^4\begin{pmatrix} \frac{1}{24} \\ \frac{-1}{12} \end{pmatrix} - t^6\begin{pmatrix} \frac{1}{720} \\ \frac{-1}{360} \end{pmatrix} + t^8\begin{pmatrix} \frac{1}{40320} \\ \frac{-1}{20160} \end{pmatrix}, \\ \lambda(t) &= 1 - \left(\frac{1}{2}\right)t^2 + \left(\frac{1}{24}\right)t^4 - \left(\frac{1}{720}\right)t^6 + \left(\frac{1}{40320}\right)t^8. \end{aligned} \quad (47)$$

One can easily check that the above approximations represent the first terms of the Maclauran power series of the exact solution. Note here that all our computations were performed exactly using Maple. To compute accurate numerical solutions over large internals and extend the interval of convergence of the solution, we apply the multistage DTM (MsDTM) algorithm given in Section 3. For this, we used an order of approximation $K = 10$ and subdivided the solution interval $[0, T] = [0, 10]$ into $N = 100$ subintervals. Starting the recursion with $u_0 = u(0) = (0, 0)^T$ and $v_0 = v(0) = (1, -2)^T$, we obtained the numerical results shown in Figures 1–6. Each of Figures 1–5 show the numerical solution (solid line), the exact solution (dashed line), and the corresponding absolute error. From the graphs (a), we can see that the numerical solution approximates well the exact solution. In each graph (b) of Figures 1–5, we can see the corresponding absolute error. These graphs show that the errors of approximation using the MsDTM are less than 4×10^{-12}. The drift errors in the position and the velocity constraints $g(u) = 0$ and $g_u'(u)v = 0$ are found to be 10^{-14} and 4×10^{-15}, as shown in Figure 6. These numerical results show that DAE constraints are satisfied with a high accuracy and that the MsDTM computes accurate numerical solutions over large intervals while satisfying all the DAE constraints. To increase the accuracy in the approximate MsDTM solution, one can increase the approximation order K or the number N of subdivisions of the solution interval $[0, T]$.

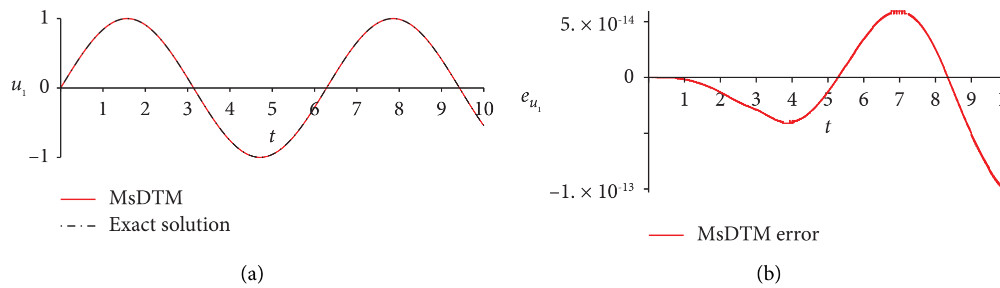


FIGURE 1: The approximate solution component u_1 (a) and the absolute error e_{u_1} (b).

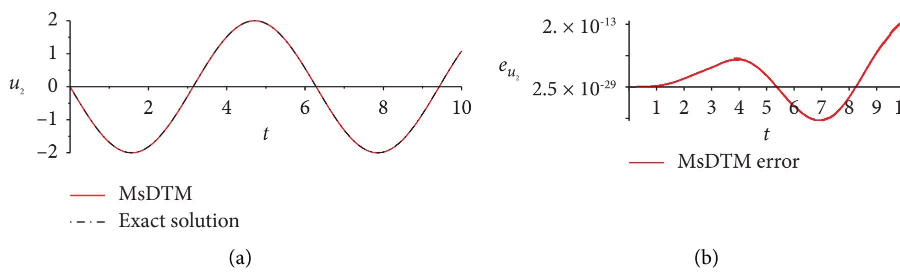


FIGURE 2: The approximate solution component u_2 (a) and the absolute error e_{u_2} (b).

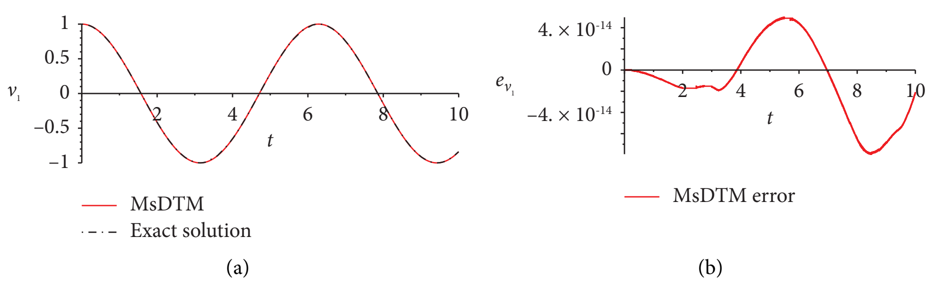


FIGURE 3: The approximate solution component v_1 (a) and the absolute error e_{v_1} (b).

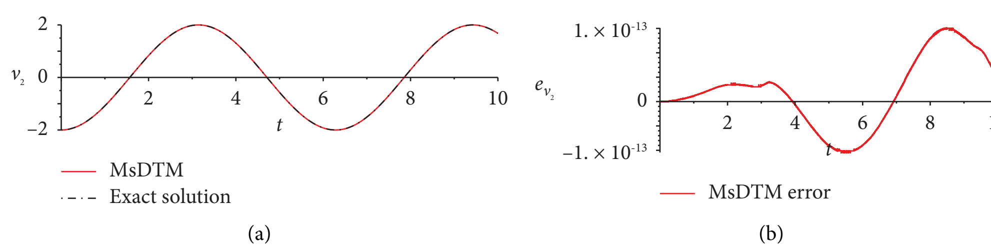


FIGURE 4: The approximate solution component v_2 (a) and the absolute error e_{v_2} (b).

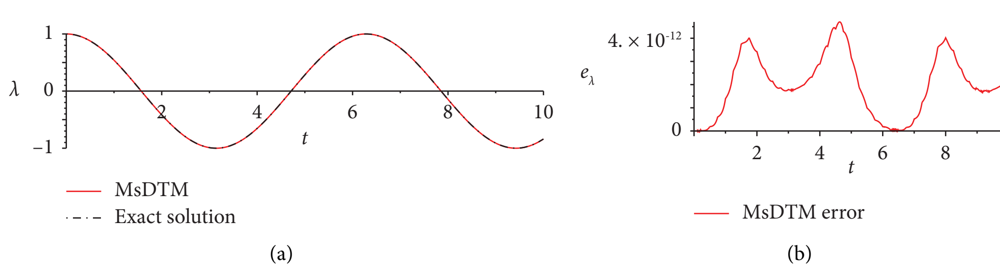
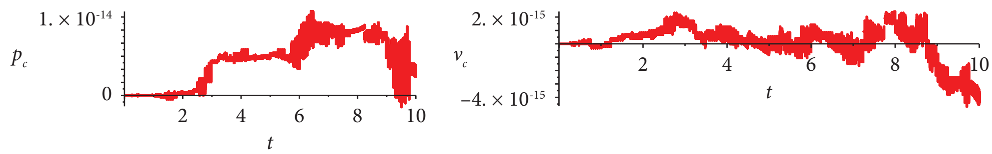
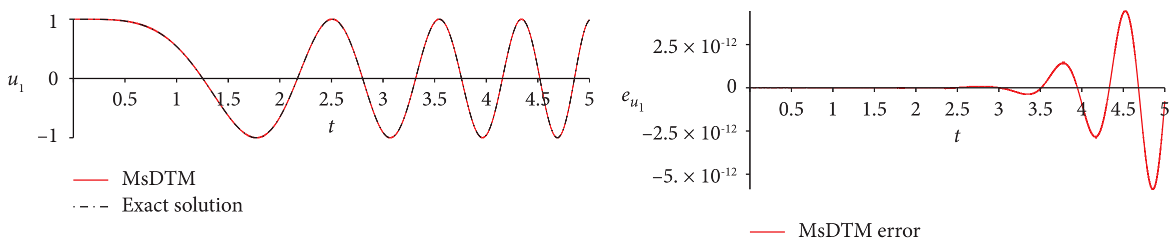


FIGURE 5: The approximate solution component λ (a) and the absolute error e_λ (b).



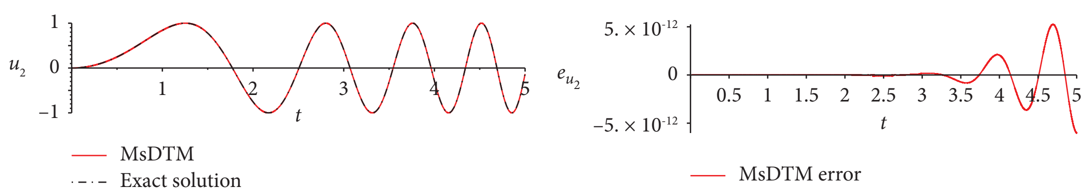
(a) (b)

FIGURE 6: The error in the position constraint $g(u) = 0$ (a) and the error in the velocity constraint $g_u'(u)v = 0$ (b).



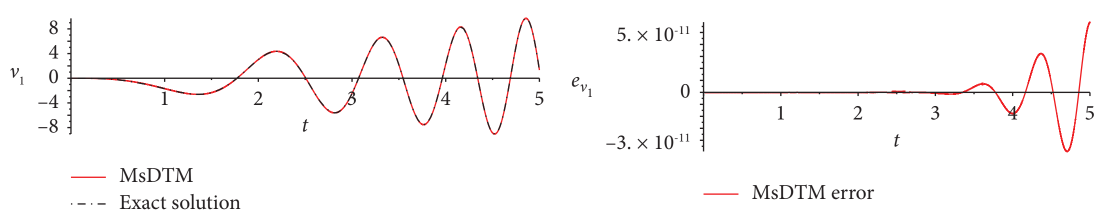
(a) (b)

FIGURE 7: The approximate solution component u_1 (a) and the absolute error e_{u_1} (b).



(a) (b)

FIGURE 8: The approximate solution component u_2 (a) and the absolute error e_{u_2} (b).



(a) (b)

FIGURE 9: The approximate solution component v_1 (a) and the absolute error e_{v_1} (b).

Example 2. In this example, we consider the DAE (1) with the vector function f given by

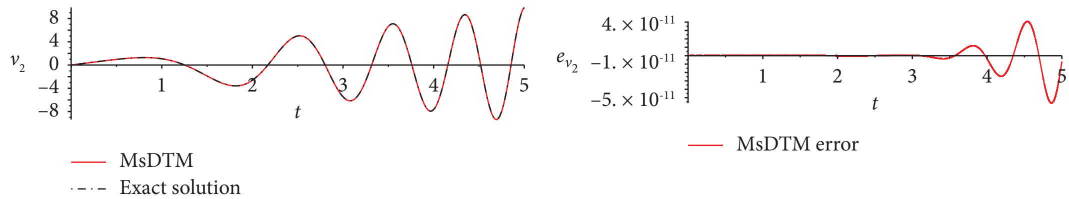
$$f(v', v, u, \lambda) = \begin{pmatrix} \tan\left(v_1' + 2u_2 - u_1\left(v_1^2 + v_2^2\right) + 8u_1\lambda\right) + 2\left(v_1' + 2u_2 - u_1\left(v_1^2 + v_2^2\right) + 8u_1\lambda\right) \\ \tan\left(v_2' - 2u_1 - u_2\left(v_1^2 + v_2^2\right) + 8u_2\lambda\right) + 3\left(v_2' - 2u_1 - u_2\left(v_1^2 + v_2^2\right) + 8u_2\lambda\right) \end{pmatrix}, \tag{48}$$

and the function g defining the constraint is given by

$$g(u) = u_1^2 + u_2^2 - 1, \tag{49}$$

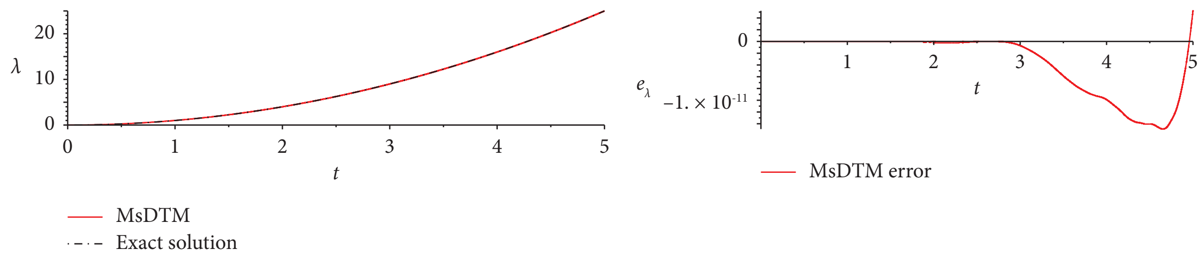
where $u = (u_1, u_2)^T \in \mathbb{R}^2$, $v = (v_1, v_2)^T \in \mathbb{R}^2$, and $\lambda \in \mathbb{R}$.

The Jacobian of g is $G(u) = g_u'(u) = (2u_1, 2u_2)$. We can easily check that the exact solution of the DAE for this Example 2 is $u(t) = (\cos(t^2), \sin(t^2))^T$, $v(t) = (-2t \sin$



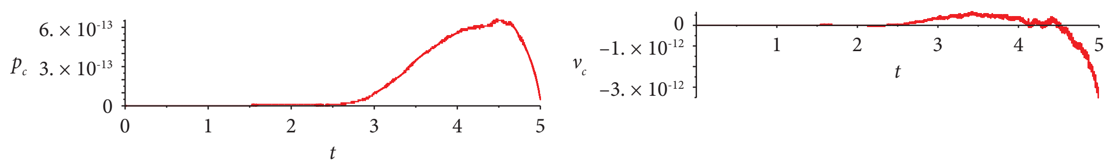
(a) (b)

Figure 10: The approximate solution component v_2 (a) and the absolute error e_{v_2} (b).



(a) (b)

Figure 11: The approximate solution component λ (a) and the absolute error e_λ (b).



(a) (b)

Figure 12: The error in the position constraint $g(u) = 0$ (a) and the error in the velocity constraint $g_u'(u)v = 0$ (b).

$(t^2), 2t\cos(t^2))^T$, and $\lambda(t) = t^2$. This exact solution is obtained from the initial conditions

$$u(0) = (1,0)^T, \qquad v(0) = (0,0)^T. \tag{50}$$

We can easily check that the above initial conditions are consistent, since they satisfy $g(u(0)) = 0$ and $g_u'(u(0))v(0) = 0$. The DAE corresponding to this example has an index-3, hence difficult to treat even numerically. To compute the solution in power series, we apply the algorithm of the standard DTM given in Section 3. By choosing the order of approximation $K = 10$, our DTM algorithm has computed the following power series approximations:

$$\begin{aligned} u(t) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + t^2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} - t^4 \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} - t^6 \begin{pmatrix} 0 \\ \frac{1}{6} \end{pmatrix} + t^8 \begin{pmatrix} \frac{1}{24} \\ 0 \end{pmatrix} + t^{10} \begin{pmatrix} 0 \\ \frac{1}{120} \end{pmatrix}, \\ v(t) &= t \begin{pmatrix} 0 \\ 2 \end{pmatrix} - t^3 \begin{pmatrix} 2 \\ 0 \end{pmatrix} - t^5 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + t^7 \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix} + t^9 \begin{pmatrix} 0 \\ \frac{1}{12} \end{pmatrix}, \\ \lambda(t) &= t^2. \end{aligned} \tag{51}$$

input
read K;
read n_u, n_λ, $f \in \mathbb{R}^{n_u}$ and $g \in \mathbb{R}^{n_\lambda}$;
read $u(0) \in \mathbb{R}^{n_u}$ and $v(0) \in \mathbb{R}^{n_u}$
output
the approximate solution: $u(t), v(t), \lambda(t)$;
while $0 \le k \le K$ **do**
  compute f_k, g_k
end
initialization
$u_0 := u(0), v_0 := v(0), u_1 := v_0$;
Compute $A_0 = f_w'(w_0, v_0, u_0, \lambda_0), B_0 = f_\lambda'(w_0, v_0, u_0, \lambda_0), G_0 = g_u'(u_0)$;
Solve the algebraic system $f(w_0, v_0, u_0, \lambda_0) = 0$ and $G_0 w_0 = -2g_2$ for w_0 and λ_0;
$u_2 := 0.5w_0$, $v_1 = 2u_0$;
$H := G_0 A_0^{-1} B_0$;
while $3 \le k \le K$ **do**
  Solve $H\lambda_{k-2} = k(k-1)s_k - G_0 A_0^{-1} r_{k-2}$ for λ_{k-2};
  Solve $k(k-1)A_0 u_k := -r_{k-2} - B_0 \lambda_{k-2}$ for u_k;
  $v_{k-1} := k u_k$;
end
$u(t) := \sum_{k=0}^{K} t^k u_k$, $v(t) := \sum_{k=0}^{K-1} t^k v_k$, and $\lambda(t) := \sum_{k=0}^{K-2} t^k \lambda_k$.

Algorithm 1: DTM algorithm.

(1) Choose the number N of subdivisions and the order of approximation K.
(2) Subdivide the interval $[0, T]$ into N equal subintervals $[t_{i-1}, t_i]$, $i = 1, \ldots, N$ of size $h = (T/N)$ where $t_i = ih$.
(3) Let $i = 1$, apply the DTM to the DAE (1) on $[0, t_1]$ using $u_0^1 = u(0)$ and $v_0^1 = v(0)$ to obtain the approximate solution $u^1(t) = \sum_{k=0}^{K} u_k^1 t^k$, $v^1(t) = \sum_{k=0}^{K-1} v_k^1 t^k$, and $\lambda^1(t) = \sum_{k=0}^{K-2} \lambda_k^1 t^k$, $0 \le t \le t_1$.
(4) From $i = 2$ until N, apply the DTM to DAE (1) on $[t_{i-1}, t_i]$ using $u_0^i = u^{i-1}(t_{i-1})$ and $v_0^i = v^{i-1}(t_{i-1})$ to obtain the approximate solutions $u^i(t) = \sum_{k=0}^{K} u_k^i (t - t_{i-1})^k$, $v^i(t) = \sum_{k=0}^{K-1} v_k^i (t - t_{i-1})^k$, and $\lambda^i(t) = \sum_{k=0}^{K-2} \lambda_k^i (t - t_{i-1})^k$, $t_{i-1} \le t \le t_i$.
(5) Repeat step 4.
(6) An approximate solution to the DAE initial-value problems (1) and (2) on $[0, T]$ is given by

$$u(t) = \begin{cases} u^1(t), & 0 \le t \le t_1 \\ u^2(t), & t_1 \le t \le t_2 \\ \vdots & \\ u^N(t), & t_{N-1} \le t \le T \end{cases}, v(t) = \begin{cases} v^1(t), & 0 \le t \le t_1 \\ v^2(t), & t_1 \le t \le t_2 \\ \vdots & \\ v^N(t), & t_{N-1} \le t \le T \end{cases} \text{ and } \lambda(t) = \begin{cases} \lambda^1(t), & 0 \le t \le t_1 \\ \lambda^2(t), & t_1 \le t \le t_2 \\ \vdots & \\ \lambda^N(t), & t_{N-1} \le t \le T. \end{cases}$$

Algorithm 2: MsDTM algorithm.

One can easily check that the above approximations represent the first terms of the Maclauran power series of the exact solution. Note here that all our computations were performed exactly using Maple. To compute accurate numerical solutions over large internals and extend the interval of convergence of the solution, we apply the multistage DTM (MsDTM) algorithm given in Section 3. For this, we used an order of approximation $K = 12$ and subdivided the solution interval $[0, T] = [0, 5]$ into $N = 300$ subintervals. Starting the recursion with $u_0 = u(0) = (1, 0)^T$ and $v_0 = v(0) = (0, 0)^T$, we obtained the numerical results shown in Figures 7–12. Each of Figures 7 to 11 show the numerical solution (solid line), the exact solution (dashed line), and the corresponding absolute error. From the graphs (a), we can see that the numerical solution approximates well the exact solution. In each graph (b) of Figures 7 to 11, we can see the corresponding absolute error. These graphs show that the errors of approximation using the MsDTM are less than 10^{-11}. The drift errors in the position and the velocity constraints $g(u) = 0$ and $g_u'(u)v = 0$ are found to be 6×10^{-13} and 3×10^{-12}, as shown in Figure 12. These numerical results show that DAE constraints are satisfied with a high accuracy and that the MsDTM computes accurate numerical solutions over large intervals while satisfying all the DAE constraints. To increase the accuracy in the approximate MsDTM solution, one can increase the approximation order K or the number N of subdivisions of the solution interval $[0, T]$.

5. Conclusion

Differential-algebraic equations (DAEs) are systems of differential and algebraic equations. Index-3 DAEs are known to be hard to treat even by using numerical methods. In this manuscript, we have proposed a new efficient algorithm that combines the differential transform method

(DTM) with the Adomian polynomials to solve the nonlinear implicit Hessenberg index-3 DAEs (1). The main advantage of this algorithm is that it is simple and does not require the use of the usual index reduction techniques, which are complex and often lead to DAEs constraints violations. The second advantage of this algorithm is that it can be easily coded in Maple or Mathematica. To illustrate the effectiveness and accuracy of our technique, two highly nonlinear implicit Hessenberg index-3 DAEs are solved by the DTM and the MsDTM. Numerical results show that the DTM algorithm has computed the exact solutions in a power series forms. To extend the interval of convergence of the DTM solution, we have developed a multistage DTM (MsDTM) version of the DTM algorithm. This DTM multistage algorithm is useful in particular when simulating engineering problems over large intervals. The numerical results show that the MsDTM has successfully handled this class of nonlinear implicit index-3 DAEs over large time intervals without violating the original DAE constraints. The MsDTM provides more accurate numerical solutions if more terms are included in the series or if more interval subdivisions are used. Our algorithms can be easily modified to solve other implicit nonlinear Hessenberg DAEs. Future work will be the design of efficient algorithms to solve more general higher index DAEs.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] E. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods*, Allyn & Bacon, Boston, MA, USA, 1989.

[2] C. A. Manrique-Escobar, C. M. Pappalardo, and D. A. Guida, “A multibody system approach for the systematic development of a closed-chain kinematic model for two-wheeled vehicles,” *Machines*, vol. 9, no. 11, p. 245, 2021.

[3] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros, “Analytically differentiable dynamics for multi-body systems with friction contact,” *ACM Transactions on Graphics*, vol. 39, no. 6, pp. 1–15, 2020.

[4] B. Simeon, F. Grupp, C. Führer, and P. Rentrop, “A nonlinear truck model and its treatment as a multibody system,” *Journal of Computational and Applied Mathematics*, vol. 50, no. 1-3, pp. 523–532, 1994.

[5] M. Balena, G. Mantriota, and G. Reina, “Dynamic handling characterization and set-up optimization for a formula SAE race car via multi-body simulation,” *Machines*, vol. 9, no. 6, p. 126, 2021.

[6] Y. Bei and B. J. Fregly, “Multibody dynamic simulation of knee contact mechanics,” *Medical Engineering & Physics*, vol. 26, no. 9, pp. 777–789, 2004.

[7] J. S. Jang, J. H. Jin, H. Y. Jung, J. H. Park, J. W. Lee, and W. S. Yoo, “Multibody dynamic analysis of a washing machine with a rapid change of mass during dehydration,” *International Journal of Precision Engineering and Manufacturing*, vol. 17, no. 1, pp. 91–97, 2016.

[8] Z. Terze, M. Vrdoljak, and H. Wolf, “Numerical simulation of landing aircraft dynamics,” *Stojarstvo*, vol. 51, no. 6, pp. 657–665, 2009.

[9] F. Han, Q. Zhou, and F. Chen, “Dynamic modeling, simulation, and flight test of a rocket-towed net system,” *Mathematical Problems in Engineering*, vol. 2019, Article ID 1523828, 21 pages, 2019.

[10] B. Rattanagraikanakorn, D. I. Gransden, M. Schuurman et al., “Multibody system modelling of unmanned aircraft system collisions with the human head,” *International Journal of Crashworthiness*, vol. 25, no. 6, pp. 689–707, 2020.

[11] T. Uppal, S. Raha, and S. Srivastava, “Trajectory feasibility evaluation using path prescribed control of unmanned aerial vehicle in differential algebraic equations framework,” *The Aeronautical Journal*, vol. 121, no. 1240, pp. 770–789, 2017.

[12] W. S. Martinson and P. I. Barton, “A differentiation index for partial differential-algebraic equations,” *SIAM Journal on Scientific Computing*, vol. 21, no. 6, pp. 2295–2315, 2000.

[13] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *The Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, Haarlem, Netherlands, 1989.

[14] J. T. Betts, S. L. Campbell, and K. C. Thompson, “Direct transcription solution of optimal control problems with differential algebraic equations with delays,” in *Proceedings of the 14 th. IASTED, International Symposium on Intelligent Systems and Control (ISC 2013)*, pp. 166–173, Cambridge, MA, USA, November 2013.

[15] S. L. Campbell and P. Kunkel, “Solving higher index DAE optimal control problems,” *Numerical Algebra, Control and Optimization*, vol. 6, no. 4, pp. 447–472, 2016.

[16] S. L. Campbell and J. T. Betts, “Comments on direct transcription solution of DAE constrained optimal control problems with two discretization approaches,” *Numerical Algorithms*, vol. 73, no. 3, pp. 807–838, 2016.

[17] V. H. Linh and N. D. Truong, “Runge-Kutta methods revisited for a class of structured strangeness-free differential-algebraic equations,” *ETNA - Electronic Transactions on Numerical Analysis*, vol. 48, pp. 131–155, 2018.

[18] D. Estévez Schwarz and R. Lamour, “Projected explicit and implicit Taylor series methods for DAEs,” *Numerical Algorithms*, vol. 88, no. 2, pp. 615–646, 2021.

[19] B. I. Akinnukawe, O. A. Akinfenwa, and S. A. Okunuga, “Hybrid block algorithm for solving differential-algebraic equation with Hessenberg index 3,” *FUTA Journal of Research in Sciences*, vol. 15, no. 1, pp. 32–39, 2019.

[20] J. Baumgarte, “Stabilization of constraints and integrals of motion in dynamical systems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 1, pp. 1–16, 1972.

[21] M. A. Neto and J. Ambrósio, “Stabilization methods for the integration of DAE in the presence of redundant constraints,” *Multibody System Dynamics*, vol. 10, no. 1, pp. 81–105, 2003.

[22] E. Bayo and A. Avello, “Singularity-free augmented Lagrangian algorithms for constrained multibody dynamics,” *Nonlinear Dynamics*, vol. 5, no. 2, pp. 209–231, 1994.

[23] E. Bayo, J. García De Jalón, and M. A. Serna, “A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 183–195, 1988.

[24] U. Ascher and P. Lin, “Sequential regularization methods for nonlinear higher-index DAEs,” *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 160–181, 1997.

[25] B. Benhammouda and H. Vazquez-Leal, "Analytical solution of a nonlinear index-three DAEs system modelling a slider-crank mechanism," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 206473, 14 pages, 2015.
[26] L. Yang and S. Kai, "Solving power system differential algebraic equations using the differential transformation," *IEEE transaction on power*, vol. 35, no. 3, 2020.
[27] B. Benhammouda, "A new numerical technique for index-3 DAEs arising from constrained multibody mechanical systems," *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 2712196, 11 pages, 2022.
[28] B. Benhammouda, "Approximate analytical solution of index-2 DAEs arising from constrained multibody systems," *Asian Research Journal of Mathematics*, vol. 8, no. 3, pp. 1–15, 2018.
[29] G. F. Abd and R. A. Zaboon, "Approximate Solution of a Reduced-Type Index-k Hessenberg Differential-Algebraic Control System," *Journal of Applied Mathematics*, vol. 2021, Article ID 9706255, 13 pages, 2021.
[30] S. Thota, "On solving system of linear differential-algebraic equations using reduction algorithm," *International Journal of Mathematics and Mathematical Sciences*, vol. 2020, Article ID 6671926, 10 pages, 2020.
[31] S. Thota, "On Maple implementation of a reduction algorithm for differential-algebraic systems," in *Proceedings of the International Conference on Mathematical Advances and Applications 2020*, Gangtok, Sikkim, July 2020.
[32] M. Golchian, M. Gachpazan, and S. H. Tabasi, "A new approach for computing the exact solutions of DAEs in generalized Hessenberg forms," *International Journal of Nonlinear Analysis and Applications*, vol. 11, no. 1, pp. 199–206, 2020.
[33] C. M. Pappalardo, A. Lettieri, and D. Guida, "Stability analysis of rigid multibody mechanical systems with holonomic and nonholonomic constraints," *Archive of Applied Mechanics*, vol. 90, no. 9, pp. 1961–2005, 2020.
[34] R. C. Rach, "A new definition of the adomian polynomials," *Kybernetes*, vol. 37, no. 7, pp. 910–955, 2008.
[35] R. Rach, "A convenient computational form for the adomian polynomials," *Journal of Mathematical Analysis and Applications*, vol. 102, no. 2, pp. 415–419, 1984.
[36] A. M. Wazwaz, "A new algorithm for calculating adomian polynomials for nonlinear operators," *Applied Mathematics and Computation*, vol. 111, no. 1, pp. 33–51, 2000.
[37] J. S. Duan, "Recurrence triangle for adomian polynomials," *Applied Mathematics and Computation*, vol. 216, no. 4, pp. 1235–1241, 2010.
[38] J. S. Duan, "An efficient algorithm for the multivariable adomian polynomials," *Applied Mathematics and Computation*, vol. 217, no. 6, pp. 2456–2467, 2010.
[39] J. S. Duan, "Convenient analytic recurrence algorithms for the adomian polynomials," *Applied Mathematics and Computation*, vol. 217, no. 13, pp. 6337–6348, 2011.
[40] G. Adomian and R. Rach, "Transformation of series," *Applied Mathematics Letters*, vol. 4, no. 4, pp. 69–71, 1991.