Caspian Journal of Mathematical Sciences (CJMS)
University of Mazandaran, Iran
http://cjms.journals.umz.ac.ir
ISSN: 1735-0611

CJMS. **7**(1)(2018), 1-15

Numerical solution of higher index DAEs using their IAE's structure: Trajectory-prescribed path control problem and simple pendulum

G. Karamali [1] and B. Shiri[2]
[1,2] PhD, Shahid Sattari Aeronautical University of Science and Technology, South Mehrabad, Tehran, Iran.

Abstract. In this paper, we solve higher index differential algebraic equations (DAEs) by transforming them into integral algebraic equations (IAEs). We apply collocation methods on continuous piecewise polynomials space to solve the obtained higher index IAEs. The efficiency of the given method is improved by using a recursive formula for computing the integral part. Finally, we apply the obtained algorithm to solve a trajectory-prescribed path control problem and a model of simple pendulum. The numerical experiments show efficiency of the given techniques.

Keywords: Differential algebraic equations, integral algebraic equations, trajectory-prescribed path control problem, simple pendulum, continuous piecewise collocation methods.

2000 Mathematics subject classification: 65L80, 65R20, 45L05.

1. Introduction

For almost half a century, differential algebraic equations (DAEs) have been studied and used for modeling of complicated technical processes [2, 4, 10, 12, 16]. Most numerical methods for DAEs are based on standard methods of the ordinary differential equations (ODEs) [4, 10]. It is well known that the robust and numerically stable application on these ODE

[1] Corresponding author: g_karamali@iust.ac.ir
Received: 17 October 2015
Revised: 8 January 2017
Accepted: 2 February 2017

methods for higher index DAEs (index greater than 1) have to be based on the structure of DAEs ([4]). Roughly speaking, the index (differential index), is the minimum number of differentiations needed to transform the DAE system into an ODE system. The higher the index is, the higher the numerical problems we get.

Motivated by physical examples, in this paper we solve the DAEs of the form

$$Ax'(t) = F(t, x(t)), \tag{1.1}$$

where $A \in L(R^r, R^r)$ is a nonsingular matrix with constant rank and $F \in \mathbf{C}(I \times R^r, R^r)$. The idea behind the method we introduce, is easy. We integrate the equation (1.1) and we obtain an integral algebraic equation (IAE) and then we solve it numerically. However it needs some consideration to operate efficiently that we will discuss in the next section.

A system of Voltera integral equations of the form

$$A(t)y(t) + \int_0^t \kappa(t, s, y(s))ds = f(t), \qquad t \in I := [0, T], \tag{1.2}$$

where $A \in \mathbf{C}(I, R^{r\times r})$, $f \in \mathbf{C}(I, R^r)$, and $\kappa \in \mathbf{C}(\mathbb{D} \times R^r, R^r)$, with $\mathbb{D} := \{(t, s) : 0 \le s \le t \le T\}$ is an IAE if $A(t)$ be a singular matrix with constant rank on $I = [0, T]$. In recent decades, IAEs have got popular and have been studied by researchers ([6, 9, 7, 17, 18, 15]). Numerical solutions of higher index IAEs of Hessenberg type using collocation methods on piecewise polynomials space are studied in [17, 18]. It is well-known that by increasing the index of DAEs, the order of the numerical methods decreases. We can choose an appropriate method of higher order to prevent these reductions, by applying the collocation methods on piecewise polynomials space. Therefore, in this paper, we use these methods to obtain efficient algorithm for solving higher index.

One of the interesting applied model in physics is a model of simple pendulum which leads to DAEs of the form (1.1). The equation of motion of a point mass m suspended from a massless rod of length l under the influence of gravity g is

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} sin(\theta) = 0 \tag{1.3}$$

where θ is the angular displacement [13]. In spite of simple construction of this well-known equation, it is still a subject of research [13, 14], since the analytical solutions of the nonlinear differential equation (1.3) is not in the form of finite series of elementary functions. The analytical solution (not in the form of finite series of elementary functions) of this equation can be found in [14] and its numerical solution by using Runge-Kutta can be found in [10].

The next problem that we investigate in this paper, is a trajectory-prescribed path control (TPPC) problem [3, 4]. This problem was introduced in [3]. We consider a space shuttle, returning from its mission which has to reenter the atmosphere. In the simulation of space shuttle, the shape of the trajectory is often prescribed by appending a set of path constraints to the equations of motion. The model equation then become a nonlinear semi-explicit DAE system.

The next sections are organized as follows: In section 2, we recall implementation of the continuous piecewise collocation methods from [6] for IAEs. In section 3, we show how we can apply these methods for DAEs. Finally, in section 4, we report numerical examples of the selected problems to show the effectiveness and efficiency of the methods.

2. Preliminaries

The collocation methods on the piecewise polynomials spaces are very suitable for the equations with integral or differential operators. They have easily implementable strategy, rapid convergent properties, less computational cost, and good stablity properties. Especially in solving many operator equations, their convergence orders don't change when the integrals are discretized (See [6, 17, 18]). Let

$$I_h := \{t_n : 0 = t_0 < t_1 < ... < t_N = T\}$$

be a given (not necessarily uniform) partition of I, and set $\sigma_n := (t_n, t_{n+1}]$, $\overline{\sigma}_n := [t_n, t_{n+1}]$, with $h_n = t_{n+1} - t_n$, $n = 0, 1, ..., N-1$ and diameter $h = max\{h_n : 0 \leq n \leq N\}$. Each component of the solution of (1.2) (the vector function $y(t)$) is approximated by elements of the piecewise polynomial space

$$\mathcal{S}_m^{(0)}(I_h) := \{v \in C(I) : v|_{\overline{\sigma}_n} \in \pi_m (n = 0, 1, ..., N-1)\}, \tag{2.1}$$

where π_m denotes the space of all (real valued) polynomials of degree not exceeding m and $v|_{\overline{\sigma}_n}$ is the restriction of function v to closure of σ_n. The collocation solution $u_h \in \left(\mathcal{S}_m^{(0)}(I_h)\right)^r$ for (1.2) is defined by the equation

$$A(t)u_h(t) + \int_0^t k(t, s, u_h(s))ds = f(t), \tag{2.2}$$

for $t \in X_h = \{t_{n,i} := t_n + c_i h_n : 0 = c_0 < c_1 < \ldots < c_m \leq 1, n = 0, \cdots, N-1\}$ and the continuity conditions

$$u_{n-1}(t_n) = u_n(t_n), \quad n = 1, \cdots, N-1. \tag{2.3}$$

The collocation parameters c_i completely determine the set of collocation points X_h. By defining $u_n = u_h|_{\overline{\sigma}_n} \in (\pi_m)^r$, we have

$$u_n(t_n + sh_n) = \sum_{j=0}^{m} L_j(s)U_{n,j}, \quad s \in (0,1], \tag{2.4}$$

$$U_{n,i} := u(t_{n,i}),$$

where the polynomials

$$L_j(v) := \prod_{\substack{k=0 \\ k \neq j}}^{m} \frac{v - c_k}{c_j - c_k}, \quad j = 0, \ldots, m$$

denote the Lagrange fundamental polynomials with respect to the distinct collocation parameters c_i. By partitioning the domain of integral in (2.2) and changing of variables, we have

$$\begin{aligned} A(t_{n,i})U_{n,i} + F_{n,i} + h \int_0^{c_i} \kappa(t_{n,i}, t_n + sh_n, L_0(s)U_{n-1}(t_n) \\ + \sum_{j=1}^{m} L_j(s)U_{n,j})ds = f(t_{n,i}), \end{aligned} \tag{2.5}$$

for $i = 1, \ldots, m$, where the lag terms are defined by

$$F_{n,i} = h \sum_{l=0}^{n-1} \int_0^1 \kappa(t_{n,i}, t_l + sh_l, L_0(s)U_{l-1}(t_n) + \sum_{j=1}^{m} L_j(s)U_{l,j}). \tag{2.6}$$

By solving the system (2.5), approximate solution of (1.2) is determined at the collocation points and at t_{n+1} by

$$u_n(t_{n+1}) = L_0(1)u_{n-1}(t_n) + \sum_{j=1}^{m} L_j(1)u_n(t_{n,j}).$$

Remark 2.1. A suitable method to solve the system (2.5) is Newton's iterative method, since it can be proved (see [1]) that this method converges to the solution $U_{n,i}$ by the initial guess $u_{n-1}(t_n)$ for sufficiently small h. Therefore, in the prescribed method we use the initial guess

$$[\underbrace{u_{n-1}(t_n), \ldots, u_{n-1}(t_n)}_{m \text{ times}}].$$

For applying this method, it is necessary to compute the appeared integrals in (2.5) and (2.6). To do this, we apply the following quadrature rule by using the same collocation parameters c_i, $i = 0, \ldots, m$, such that the order of the quadrature rule would be at least in the same order of the method $\left(\mathcal{O}(h^{m+1})\right)$,

$$\int_0^{c_i} k(t_{n,i},t_n + sh_n, L_0(s)U_{n-1}(t_n) + \sum_{j=1}^{m} L_j(s)U_{n,j})ds$$
$$\simeq \sum_{j=1}^{m} a_{i,j}k(t_{n,i}, t_n + c_jh_n, U_{n,j}),$$

$$\int_0^{1} k(t_{n,i},t_l + sh_l, L_0(s)U_{l-1}(t_n) + \sum_{j=1}^{m} L_j(s)U_{l,j})ds$$
$$\simeq \sum_{j=1}^{m} b_jk(t_{n,i}, t_n + c_jh_n, U_{l,j}),$$

with $a_{i,j} = \int_0^{c_i} L_j(t)dt$ and $b_j = \int_0^1 L_j(t)dt$. Using this quadrature rule simplifies our computations considerably. If all of the integrals are computed by quadrature rule then the method is called fully discretised continuous colocation method (DCCM).

Remark 2.2. By choosing $c_m = 1$, for $m \geq 2$, we have $t_{n+1} = t_{n,m}$ and $u(t_{n+1}) = u(t_{n,m})$, thus we obtain $u_{n+1} = U_{n,m}$ from the previous subinterval without reusing (2.4).

3. The continuous piecewise collocation methods for DAEs

To the DAEs of the form (1.1), by integrating we obtain

$$Ax(t) - \int_0^t F(s, x(s))ds = Ax(0), \tag{3.1}$$

which is an IAEs of the form (1.2) and hence one can use the numerical methods of the pervious section to solve this equation. Considering the kernel of this equation which is independent of t, we can restate the equations of lag terms in such a way that the computational cost decreases considerably. In this case the lag terms are computed as

$$F_{n,i} = h\sum_{l=0}^{n-1}\sum_{j=0}^{m} b_jF(t_l + c_jh_n, U_{l,j}), \tag{3.2}$$

where m is a fixed integer depend on the method (usually we choose m less than 7) but the parameter n may increase more and more for some stiff problems. In IAEs case, the computational cost of $F_{n+1,i}$ is $\mathcal{O}(n^2)$, because of the presence of $t_{n,i}$ for each n and i. But for DAEs, we can use the recursive formula

$$F_{n+1,i} = F_{n,i} + \sum_{j=0}^{m} b_j F(t_{n,j}, U_{n,j})$$

to obtain $F_{n,i}$ which is independent of i, (and we can drop index i). Thus the computation will be of order $\mathcal{O}(n)$, which makes this method compatible with the Runge-Kutta methods.

Therefore, for the DAEs (1.1), the method can be simplified to

$$\begin{aligned} F_0 &= A(t_0)y_0, \quad U_{0,0} = y_0, \\ A(t_{n,i})U_{n,i} &= F_n + h\sum_{j=0}^{m} a_{ij} F(t_{n,j}, U_{n,j}), \quad i = 1, \dots, m \\ F_{n+1} &= F_n + h\sum_{j=0}^{m} b_j F(t_{n,j}, U_{n,j}), \\ U_{n+1,0} &:= u_n(t_{n+1}) = \sum_{j=0}^{m} L_j(1) U_{n,j}, \end{aligned} \tag{3.3}$$

for $n = 1, \dots, N-1$, and $h = \frac{T}{N}$.

Remark 3.1. Note that, these methods are of Runge-Kutta type iff $U_{n+1} = F_{n+1}$. This condition holds for the case $c_m = 1$, for $m \geq 2$, which has considerable simplification in the formula (3.3). In this case, $U_{n+1} = F_{n+1} = U_{n,m}$.

The restrictions exist in choosing c_is for the first IAEs of Hessenberg type [17]. Due to [17], we should choose c_i such that $c_0 = 0$ and

$$\varrho = \max\{|\lambda_1|, |\lambda_2|\} \leq 1,$$

where λ_1 and λ_2 are eigenvalues of the stability matrix $\widetilde{A}^{-1}B$. The matrices $\widetilde{A}^{-1}$ and B are introduced in [17] as follow

$$\widetilde{A} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ a_{10} & a_{11} & \dots & a_{11} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m0} & a_{m1} & \dots & a_{mm} \end{pmatrix}$$

and

$$B = \begin{pmatrix} L_0(1) & \dots & L_m(1) \\ a_{m0} - b_0 & \dots & a_{mm} - b_m \\ \vdots & \ddots & \vdots \\ a_{m0} - b_0 & \dots & a_{mm} - b_m \end{pmatrix}.$$

These eigenvalues can be computed as follow

$$\lambda_1 = \frac{1}{2}\left(tr(\widetilde{A}^{-1}B) + \sqrt{(tr(\widetilde{A}^{-1}B))^2 - 4(L_0(1))^2} \right), \tag{3.4}$$

$$\lambda_2 = \frac{1}{2}\left(tr(\widetilde{A}^{-1}B) - \sqrt{(tr(\widetilde{A}^{-1}B))^2 - 4(L_0(1))^2} \right), \tag{3.5}$$

$$tr(\widetilde{A}^{-1}B) = L_0(1)\left(2 + \sum_{i=1}^{m} \frac{1}{c_i} + \sum_{i=1}^{m} \frac{1}{1-c_i} \right), \tag{3.6}$$

for $c_m < 1$ and

$$\lambda_1 = 0, \tag{3.7}$$

$$\lambda_2 = (-1)^m \prod_{i=1}^{m-1} \frac{1-c_i}{c_i}, \tag{3.8}$$

for $c_m = 1$ (see [17]). Since the solution obtained from (3.3) is exactly the same solution obtained from (3.1), by applying (DCCM), the convergence results of [17] also hold for (3.3). Therefore, the order of the method decreases by increasing the index of the DAEs and for higher index DAEs, we should use higher order methods which can be achieved by increasing m.

4. Sample problems

In this section, we use some selected sample problems to illustrate the effectiveness and high accuracy of the given method.

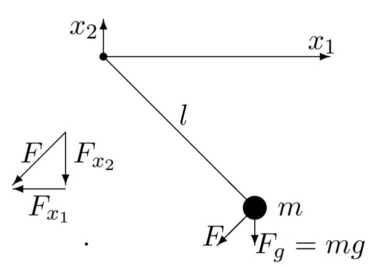

Figure 1. Simple pendulum-scheme

4.1. **Simple pendulum.** In this section we consider the simple pendulum of length l, mass m under the influence of gravity $F_g = -mg$, where $g = 9.8\ m/s^2$ is the gravity constant (Figure 1). Using the classical Euler-Lagrange formula [8, 11] in Cartesian coordinates $[x_1, x_2]^T =$

Table 1. Numerical results of index 3 DAE (4.1) for $x_1(t)$ with $N = 500$.

t	method 1	method 2	ode15s
2	0.791415503888909	0.791415099102267	0.79141509926
4	−0.584175128656931	−0.584197053914579	−0.58419714676
6	−0.999568975065766	−0.999569743078752	−0.99956974661
8	−0.915348262923567	−0.915330990754696	−0.91533091583
10	0.296123754641507	0.296271070783072	0.29627172089
Computing time	$2.06s$	$3.14s$	$3.76s$

Table 2. Numerical results of index 3 DAE (4.1) for $x_2(t)$ with $N = 500$.

t	method 1	method 2	ode15s
2	−0.611278553797112	−0.611279102303482	−0.61127910209
4	−0.811627593357415	−0.811611854396871	−0.81161178758
6	−0.029358053740116	−0.029331360716471	−0.02933124145
8	−0.402663087278485	−0.402702343380354	−0.40270251353
10	−0.955149523562636	−0.955103896242212	−0.95510369463
Computing time	$2.06s$	$3.14s$	$3.76s$

Table 3. Numerical results of index 2 DAE (4.2) for $x_1(t)$ with $N = 500$.

t	method 1	method 2	ode15s
2	0.791415170748194	0.791415099376876	0.79141509926
4	−0.584196752919579	−0.584197220670537	−0.58419714676
6	−0.999569736371046	−0.999569755689938	−0.99956974661
8	−0.915331234981749	−0.915330861481158	−0.91533091583
10	0.296269503434385	0.296272220867141	0.29627172089
Computing time	$1.99s$	$3.07s$	$3.76s$

$[x, y]^T$ and the velocity vector $[x_3, x_4]^T = [\dot{x}, \dot{y}]^T$, one obtain the following index 3 DAE problem

$$\begin{aligned} \dot{x}_1 &= x_3, \\ \dot{x}_2 &= x_4, \\ \dot{x}_3 &= -x_1\lambda, \\ \dot{x}_4 &= -g - x_2\lambda, \\ 0 &= x_1^2 + x_2^2 - l^2, \end{aligned} \tag{4.1}$$

TABLE 4. Numerical results of index 2 DAE (4.2) for $x_2(t)$ with $N = 500$.

t	method 1	method 2	ode15s
2	-0.611279003691347	-0.611279108658166	-0.61127910209
4	-0.811612064882334	-0.811611737838670	-0.81161178758
6	-0.029331555088146	-0.029331165734774	-0.02933124145
8	-0.402701741692676	-0.402702663299050	-0.40270251353
10	-0.955104342069682	-0.955103550550979	-0.95510369463
Computing time	$1.99s$	$3.07s$	$3.76s$

TABLE 5. Numerical results of index 1 DAE (4.3) for $x_1(t)$ with $N = 500$.

t	method 1	method 2	ode15s
2	0.791415239425474	0.791415065252080	0.79141509926
4	-0.584206187091868	-0.584196780385478	-0.58419714676
6	-0.999571288011394	-0.999569533743246	-0.99956974661
8	-0.915318931747990	-0.915331688232221	-0.91533091583
10	0.296463994504867	0.296250795527552	0.29627172089
Computing time	$1.6s$	$2.59s$	$3.76s$

TABLE 6. Numerical results of index 1 DAE (4.3) for $x_2(t)$ with $N = 500$.

t	method 1	method 2	ode15s
2	-0.611278141437346	-0.611279176060579	-0.61127910209
4	-0.811606367517056	-0.811611944121050	-0.81161178758
6	-0.029307506124896	-0.029333223880495	-0.02933124145
8	-0.402723355480016	-0.402701610197581	-0.40270251353
10	-0.955045825877874	-0.955109908908153	-0.95510369463
Computing time	$1.6s$	$2.59s$	$3.76s$

which is equivalent to (1.3) with some consideration. Differentiating from the last equation of system (4.1), we obtain the index 2 DAE

$$
\begin{aligned}
\dot{x}_1 &= x_3, \\
\dot{x}_2 &= x_4, \\
\dot{x}_3 &= -x_1\lambda, \\
\dot{x}_4 &= -g - x_2\lambda, \\
0 &= x_1x_3 + x_2x_4,
\end{aligned}
\tag{4.2}
$$

and another differentiating from the last equation of system (4.2), we can obtain an index 1 DAE

$$\begin{aligned} \dot{x}_1 =& x_3, \\ \dot{x}_2 =& x_4, \\ \dot{x}_3 =& -x_1\lambda, \\ \dot{x}_4 =& -g - x_2\lambda, \\ 0 =& x_3^2 + x_4^2 - gx_2 - l^2\lambda. \end{aligned} \tag{4.3}$$

This is called the index reduction procedure. After this index reduction one can solve the index 1 DAE (4.3) by available software and codes like "ode15s" in MATLAB. However, this index reduction cannot be obtained for many DAEs because of their nonlinearity or complexity (sometimes the obtained equations are very complicated with many nonlinear long terms). Consequently, we are interested in direct solving of index 3 DAE (4.1). Thus, we solve all of the equations (4.1)-(4.3) as a test problem and compare them with the numerical solution of applying "ode15s" on (4.3).

Example 4.1. We consider the system (4.1)-(4.3) by the parameter $l = 1$ and the consistency initial conditions

$$x_1(0) = 1,\ x_2(0) = 0,\ x_3(0) = 0,\ x_4(0) = 0, \lambda(0) = 0.$$

We use the following methods to get numerical solutions of this example on $t \in [0, 10]$ and $h = \frac{10}{N}$ with $N = 500$.

- **Method 1::** Let $c = [0, 0.5, 0.8, 0.88]$. For this c, we have $\lambda_1 = -0.0016$ and $\lambda_2 = -0.7389$.
- **Method 2::** Let $c = [0, 0.5, 0.8, 0.88, 1]$. For this c, we have $\lambda_1 = 0$ and $\lambda_2 = -0.034$.

In Tables 1 and 2, we compared the obtained results with the numerical results of the command "ode15s" in MATLAB software. We set the extreme ode options abstol $= 100eps$ and reltol $= 100eps$ for the command "ode15s". These Tables show the efficiency of the method.

4.2. **Trajectory-prescribed path control problem.** Trajectory- prescribed path control problem for shuttle reentry was solved for different shuttles and different conditions (see for examples [3, 5, 19]). Here, we use the scale and data of [5]. Suppose a space shuttle with mass $m = 2.890532728$ slugs, cross sectional reference area $S = 1$ ft^2, and with relative velocity $V_R = 12000$ ft/s, at the altitude of $H = 100000$ ft, longitude $\epsilon = 0°$, and latitude $\lambda = 0°$, has to reenter the atmosphere.

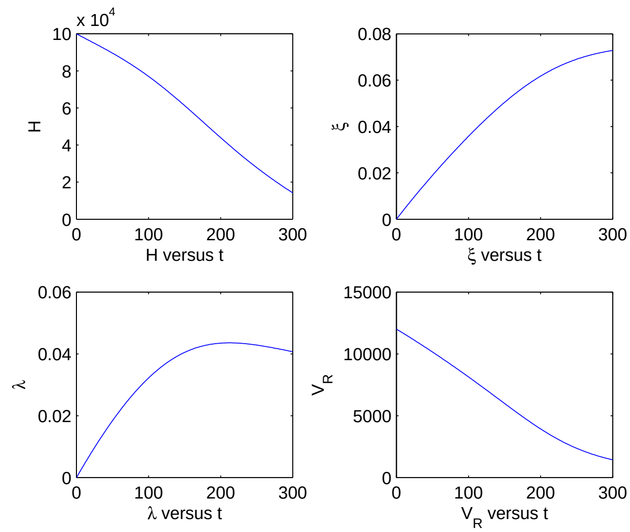

FIGURE 2. Numerical solution of TPPC problem using method 1, with $N = 250$, state variables.

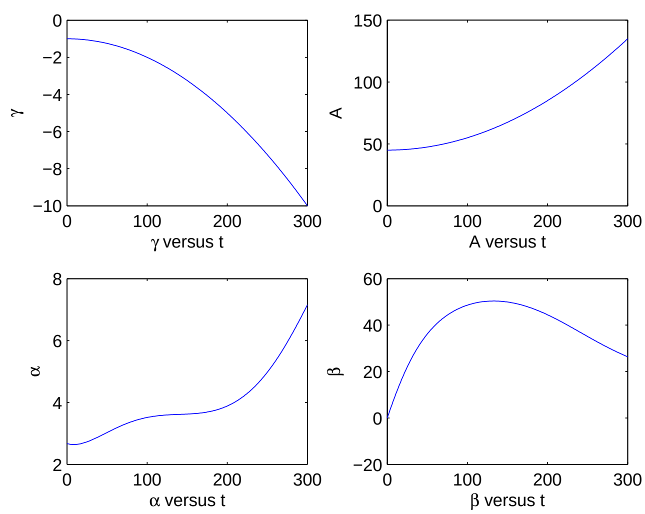

FIGURE 3. Numerical solution of TPPC problem using method 1, with $N = 250$, state and control variables.

Here, the aerodynamic lift and drag force are given by $L = \frac{1}{2}\rho V_R^2 SC_L$, and $D = \frac{1}{2}\rho V_R^2 SC_D$, respectively, where, $\rho(H) = 0.002378e^{-H/23800}$, and the drag and lift coefficients are calculated by $C_L = 0.01\alpha$, and

Table 7. Numerical results of index 2 DAE (4.4)-(4.5) at $t = 300$ with $N = 250$.

	method 1	method 2	method 3
H	14200.786553704	14200.786553653	14200.786553654
ϵ	0.0727991723743	0.0727991723742	0.0727991723742
λ	0.0406923062717	0.0406923062717	0.0406923062717
V_R	1433.2929482465	1433.2929482436	1433.2929482436
γ	−0.174532925199	−0.174532925199	−0.174532925199
A	2.3561944901923	2.3561944901923	2.3561944901923
α	7.1564690302731	7.1564690301584	7.1564690301516
β	0.4603119285387	0.4603119285335	0.4603119285300
Computing time	1.5s	2.6s	30s

Table 8. Numerical results of index 2 DAE (4.4)-(4.5) at $t = 300$ with $N = 500$.

	method 1	method 2	method 3
H	14200.786553656	14200.786553653	14200.786553654
ϵ	0.0727991723742	0.0727991723742	0.0727991723742
λ	0.0406923062717	0.0406923062717	0.0406923062717
V_R	1433.2929482437	1433.2929482436	1433.2929482436
γ	−0.174532925199	−0.174532925199	−0.174532925199
A	2.3561944901923	2.3561944901923	2.3561944901923
α	7.1564690301574	7.1564690301595	7.1564690301516
β	0.4603119285320	0.4603119285337	0.4603119285300
Computing time	3.1s	5.3s	30s

$0.04 + 0.1C_L^2$, respectively. The gravity force is mg, where the gravitational acceleration g is given by $g = \frac{\mu}{r^2}$. Here $\mu = 1.407653916 \times 10^{16}$, is the gravitational constant, $r = H + a_e$, is the distance of shuttle from the center of the earth, and $a_e = 20902900$, is the earth radius. The centrifugal and coriolis forces are proportional to Ω^2 and Ω, respectively, where $\Omega = 2\pi/(24 \times 60 \times 60)$, is the angular velocity of the earth. The dynamic of the shuttle is obtained by the following 6 dimensional system of first order differential equations:

$$
\begin{aligned}
H' &= V_R \sin(\gamma), \\
\epsilon' &= \frac{V_R \cos(\gamma)\sin(A)}{r\cos(\lambda)}, \\
\lambda' &= \frac{V_R}{r}\cos(\gamma)\cos(A), \\
V_R' &= -\frac{D}{m} - g\sin(\gamma) \\
&\quad -\Omega^2 r\cos(\lambda)\left(\sin(\lambda)\cos(A)\cos(\gamma) - \cos(\lambda)\sin(\gamma)\right), \\
\gamma' &= \frac{L\cos(\beta)}{mV_R} + \frac{\cos(\gamma)}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega\cos(\lambda)\sin(A), \\
&\quad +\frac{\Omega^2 r\cos(\lambda)}{V_R}\left(\sin(\lambda)\cos(A)\cos(\gamma) - \cos(\lambda)\sin(\gamma)\right), \\
A' &= \frac{L\sin(\beta)}{mV_R\cos(\gamma)} + \frac{V_R}{r}\cos(\gamma)\sin(A)\tan(\lambda), \\
&\quad -2\Omega\left(\cos(\lambda)\cos(A)\tan(\gamma) - \sin(\lambda)\right), \\
&\quad +\frac{\Omega^2 r\cos(\lambda)\sin(\lambda)\sin(A)}{V_R\cos(\gamma)}.
\end{aligned} \tag{4.4}
$$

Here, the control parameters are α and β which show the angle and the bank of the attack, respectively. The constraints for the trajectory to be followed by the vehicle are given only in terms of zenith angle γ, and azimuth angle A :

$$
\begin{aligned}
\gamma &= -1 - 9\left(\frac{t}{300}\right)^2, \\
A &= 45 + 90\left(\frac{t}{300}\right)^2.
\end{aligned} \tag{4.5}
$$

The system (4.4)-(4.5) is a 8 dimensional nonlinear DAE of index2. Substituting the derivative of (4.5) into the last equations of (4.4), we can obtain algebraic equations for α, and β. For $t = 0$, the solution of these equations gives initial conditions $\alpha_0 = 2.6733$ and $\beta_0 = -0.0520$.

Example 4.2. Now, we can apply the (3.3) to the DAE (4.4)-(4.5). To obtain more accurate results, we can increase m, the number of collocation points or increase N. For this index 2 problem, we can obtain the methods with order of convergence $\mathcal{O}\big(\frac{T}{N}\big)^{m+1-2}$, [17]. Therefore, we choose

Method 3: $c = [0, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 1]$, with $\lambda_1 = 0$, and $\lambda_2 = 0.00042$, and set $N = 1000$ to get a more accurate result. We apply the method 1 and 2, of Example 4.1. Figures 2 and 3 illustrate the numerical solutions using method 1, with $N = 250$. Tables 7 and 8 show the numerical solutions of Methods 1, 2 at the end point $t = 300$ for $N = 250, 500$. These tables show efficiency and effectiveness of the introduced methods.

CONCLUSION

In this paper we used continuous piecewise collocation methods for solving DAEs appeared in some physical models. The equations that we dealt with did not have closed form analytical solutions. Our numerical experiments show efficiency and effectiveness of the proposed methods with rapid convergence, easily implementable properties, and less computational cost.

References

[1] K. Atkinson and W. Han, Theoretical numerical analysis, (Vol. 39), Berlin: Springer, 2005.

[2] U. M. Ascher and L. R. Petzold, Computer methods for ordinary differential equations and differential-algebraic equations, (Vol. 61), SIAM, 1998.

[3] K. E. Brenan, Stability and convergence of difference approximations for higher index differential algebraic systems with applications in trajectory control, PhD diss., 1983.

[4] K. E. Brenan, S. L. Campbell and L. R. Petzold Numerical solution of initial-value problems in differential-algebraic equations, (Vol. 14). SIAM, 1996.

[5] K. E. Brenan and L. R. Petzold, The Numerical Solution of Higher Iindex Differential/Algebraic Equations by Implicit Methods, *SIAM J. Numer. Anal.,* **26(4)** *(1989), 976-996.*

[6] H. Brunner, Collocation methods for Volterra integral and related functional differential equations, (Vol. 15), Cambridge University Press, 2004.

[7] O. G. S. Budnikova and M. V. Bulatov, Numerical solution of integral-algebraic equations for multistep methods, *Comput. Math. Math. Phys.,* **52(5)** *(2012), 691-701.*

[8] E. Eich-Soellner and C. Führer, Numerical methods in multibody dynamics, (Vol. 45), Stuttgart: Teubner, 1998.

[9] C. W. Gear, Differential algebraic equations, indices, and integral algebraic equations, *SIAM J. Numer. Anal.,* **27(6)** *(1990), 1527-1534.*

[10] E. Hairer, C. Lubich and M. Roche, The numerical solution of differential-algebraic systems by Runge-Kutta methods, Springer, 1989.

[11] E. Hairer and G. Wanner, Solving ordinary differential equations ii: Stiff and differential-algebraic problems second revised edition with 137 figures, Springer series in computational mathematics, 14, 1996.

[12] P. Kunkel and V. L. Mehrmann, Differential-algebraic equations: analysis and numerical solution, Eur. Math. Soc., 2006.

[13] R. A. Nelson, and M. G. Olsson, The pendulum-rich physics from a simple system, *Am. J. Phys,* **54(2)** *(1986), 112–121.*

[14] K. Ochs, A comprehensive analytical solution of the nonlinear pendulum, *Eur. J. Phys.,* **32(2)***(2011), 479.*

[15] S. Pishbin, Optimal convergence results of piecewise polynomial collocation solutions for integralalgebraic equations of index-3,*J. Comput. Appl. Math.,* **279***(2015), 209-224.*

[16] P. J. Rabier and W. C. Rheinboldt, Theoretical and numerical analysis of differential-algebraic equations,*Handbook of numerical analysis,* **8***(2002), 183-540.*

[17] B. Shiri, S. Shahmorad and G. Hojjati, Convergence analysis of piecewise continuous collocation methods for higher index integral algebraic equations of the Hessenberg type,*Int. J. Appl. Math. Comput. Sci,* **23(2)***(2013), 341-355.*

[18] B. Shiri, Numerical solution of higher index nonlinear integral algebraic equations of Hessenberg type using discontinuous collocation methods, *Math. Model. anal.,* **19(1)***(2014), 99-117.*

[19] H. S. Wang, W. L. Jhu, C. F. Yung and P. F. Wang, Numerical solutions of differential-algebraic equations and its applications in solving TPPC problems, *J. Mar. Sci. Technol.,* **19(1)***(2011), 76-88.*