

Article

Not peer-reviewed version

Modified Extended Lie Group Method for Hessenberg Differential Algebraic Equations with Index 3

[Juan Tang](#) and [Jianguang Lu](#) *

Posted Date: 14 April 2023

doi: 10.20944/preprints202304.0359.v1

Keywords: Differential algebraic equations; Lie group; Hessenberg; High index



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: 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.

Article

Modified Extended Lie Group Method for Hessenberg Differential Algebraic Equations with Index 3

Juan Tang¹  and Jianguang Lu^{2,3,*}

¹ School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China; tangjn16@gzhu.edu.cn(J.T.);

² State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China; jglu@gzu.edu.cn(J.L.);

³ Chongqing Innovation Center of Industrial Big-Data Co., Ltd., Chongqing 400707, China;

* Correspondence: jglu@gzu.edu.cn.

Abstract: Hessenberg differential algebraic equations (Hessenberg-DAEs) with high index play a critical role in the modeling of mechanical systems and multibody dynamics. Motivated by the widely used Lie Group Differential Algebraic Equation (LGDAE) method which only handles index 2 systems, we propose a Modified Extended Lie Group Differential Algebraic Equation (MELGDAE) method for solving index 3 Hessenberg-DAEs, and provide theoretical analysis to deepen the foundation of the MELGDAE method. The performance of the MELGDAE method is compared with the standard methods RADAU and MEBDF on index 2 and 3 DAE systems, and it is demonstrated that the MELGDAE integrator exhibits competitive performance in terms of high accuracy and the preservation of algebraic constraints. In particular, all differential variables in index 3 Hessenberg DAEs achieve second-order convergence using the MELGDAE method, which suggests potential for extension to Hessenberg-DAEs with an index of 4 or higher.

Keywords: differential algebraic equations; lie group; Hessenberg; high index

1. Introduction

Differential-Algebraic Equations (DAEs) are a class of mathematical models that arise in various fields, such as multibody dynamics [1], electrical circuits [2], fluid mechanics [3], optimal control [4] and chemical reactions [5]. DAEs are a natural extension of ordinary differential equations that also involve algebraic equations. As such, they offer a convenient way to model complex systems that involve both dynamic and algebraic components. The study of DAEs has been an active area of research for many years, and numerous numerical methods have been developed to solve these DAEs, particularly for low index systems. These methods include well known Runge-Kutta methods [6–8], backward differentiation formulae [9,10], and others. However, the efficient solution of DAEs remains a challenging problem due to the presence of algebraic equations that may be nonlinear, implicit, and of high index.

Hessenberg differential algebraic equations (Hessenberg-DAEs) are an important type of differential equation which are also commonly used in the modeling of mechanical systems and electrical circuits. Because of their unique structure, numerical methods for solving Hessenberg-DAEs can differ significantly from those used for other types of differential equations. Over the years, some approaches have been proposed for the numerical solution of Hessenberg-DAEs, including variational iteration method [11], adomian decomposition method for [12], exponential integrators [13]. These methods either lack rigorous theoretical analysis or are only suitable for low-index systems.

One promising approach is the Lie group method whose several variations have been developed specifically for solving (Hessenberg) DAEs with high index in recent years. Lie group methods mostly constitute one-step extensions of classical methods, including Runge-Kutta type formulas [14]. In mechanical engineering, BDF methods, which have traditionally been used, have been extended to Lie groups as well [15]. Despite their usefulness, these methods suffer from a numerical disadvantage. Various numerical techniques can be used to compute the matrix exponential, but the properties

of these methods, such as stability, accuracy, and efficiency, can impact the results of the Lie-group methods. Based on the structural information of Hessenberg DAEs, C.S.Liu [16–18] propose a Lie-group differential algebraic equations (LGDAE) method. By converting the canonical representation of the governing equations for nonholonomic systems, the LGDAE method expresses them in the form of DAE systems, comprising of nonlinear ordinary differential equations and nonlinear algebraic equations. The method is composed of two components: the first component pertains to the Lie-group structure and concerns the numerical solution of nonlinear ordinary differential equations. The second component, the Newton iterative scheme, is used to solve the nonlinear algebraic equations. The LGDAE method has been successfully employed to solve various practical problems such as the heat source recovery problem [19,20], sliding mode control problem [21], and inverse nonlinear vibration problems [22]. However, it is noteworthy that the LGDAE method is only applicable to systems with index 2.

The main focus of this paper is to investigate Lie group methods for Hessenberg-DAEs with an index of 3 or higher. We present a modified extended Lie group method that incorporates the benefits of Liu's original method, along with the structural characteristics of Hessenberg-DAEs. Our proposed method enables the accurate solution of high-index Hessenberg-DAEs and effectively captures the behavior of solutions to algebraic constraints. We demonstrate the effectiveness of our method through numerical experiments on some benchmark problems.

The structure of this paper is as follows. Section 2 presents a brief introduction to Hessenberg-DAEs and Lie group methods. In Section 3, we present our novel modified extended Lie group DAE method for Hessenberg-DAEs with an index of 3, and provide a theoretical justification for the proposed method. Section 4 presents the results of numerical experiments on several benchmark problems. Finally, we conclude with a discussion of our results and directions for future research in Section 5.

2. Preliminaries

In this section, we provide a concise overview of Hessenberg-DAEs and an implicit $GL(n, \mathbb{R})$ Lie group methods.

2.1. Hessenberg DAEs

A system of ordinary differential equations (ODEs) with algebraic constraints which can be written in the general form as

$$\mathbf{f}(t, \mathbf{x}(t), \mathbf{x}'(t)) = 0 \quad (1)$$

is called differential algebraic equation, where $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ may be singular and the structure and rank of the of this Jacobian matrix may depend on the solution $\mathbf{x}(t)$. An important special case of 1 is the semi-explicit DAE of the form

$$\begin{cases} \mathbf{x}_1'(t) = \mathbf{f}_1(t, \mathbf{x}_1(t), \mathbf{x}_2(t)) \\ 0 = \mathbf{f}_2(t, \mathbf{x}_1(t), \mathbf{x}_2(t)), \end{cases} \quad (2)$$

whose index is 1 if the Jacobian matrix function $\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2}$ is assumed to be nonsingular for all t . Hessenberg Index-1 is also often referred to as a semi-explicit index-1 system.

Definition 1. *Hessenberg Index-2*

$$\begin{cases} \mathbf{x}_1'(t) = \mathbf{f}_1(t, \mathbf{x}_1(t), \mathbf{x}_2(t)), & (3a) \\ 0 = \mathbf{f}_2(t, \mathbf{x}_1(t)), & (3b) \end{cases}$$

where f_i are sufficiently differentiable, the variables x_2 of the algebraic part is absent from the constraint and the product of Jacobians $\frac{\partial f_2}{\partial x_1} \frac{\partial f_1}{\partial x_2}$ is nonsingular. Thus, all algebraic variables play the role of index-2 therefore this is a pure index-2 DAE.

Definition 2. *Hessenberg Index-3*

$$\begin{cases} x_1'(t) = f_1(t, x_1(t), x_2(t), x_3(t)), & (4a) \\ x_2'(t) = f_2(t, x_1(t), x_2(t)), & (4b) \\ 0 = f_3(t, x_2(t)). & (4c) \end{cases}$$

In this case the product of three matrix functions $F = \frac{\partial f_3}{\partial x_2} \frac{\partial f_2}{\partial x_1} \frac{\partial f_1}{\partial x_3}$ is nonsingular, where x_1 , x_2 and x_3 are n_1 , n_2 and n_3 dimensions respectively. The index of a Hessenberg DAE is found, as in the general case, by differentiation. However, only the algebraic constraints must be differentiated.

Definition 3. *Hessenberg Index-r*

$$\begin{cases} x_1' = f_1(t, x_1, x_2, \dots, x_r), & (5a) \\ x_2' = f_2(t, x_1, x_2, \dots, x_{r-1}), & (5b) \\ \vdots & (5c) \\ x_i' = f_i(t, x_{i-1}, x_i, \dots, x_{r-1}), \quad 3 \leq i \leq r-1, & (5d) \\ \vdots & (5e) \\ 0 = f_r(t, x_{r-1}). & (5f) \end{cases}$$

In this case the product of three matrix functions $\frac{\partial f_r}{\partial x_{r-1}} \frac{\partial f_{r-1}}{\partial x_{r-2}} \dots \frac{\partial f_2}{\partial x_1} \frac{\partial f_1}{\partial x_r}$ is nonsingular.

2.2. Lie group structure of ODEs

A Lie group is a differentiable manifold equipped with a group structure that is consistent with the underlying topology of the manifold. The general linear group is a Lie group, whose manifold is an open subset $GL(n, \mathbb{R}) := \{G \in \mathbb{R}^{n \times n} | \det G \neq 0\}$. Thus, $GL(n, \mathbb{R})$ is also an $n \times n$ -dimensional manifold. The group composition is given by the matrix multiplication.

The general linear group $GL(n, \mathbb{R})$ gives uniquely a real Lie-algebra $gl(n, \mathbb{R})$. Consider a one-parameter subgroup $G(t)$, $t \in \mathbb{R}$, of the general linear group $GL(n, \mathbb{R})$, which is a curve passing through the group identity at $t = 0$, $G(0) = I_n$, and which left acts on the n -dimensional Euclidean space \mathbb{R}^n , resulting in a congruence of curves in \mathbb{R}^n ,

$$X(t) = G(t)X(0) = [G(t)G^{-1}(t_0)]X(t_0), \quad t_0, t \in \mathbb{R}. \quad (6)$$

Owing to the closure property of the Lie group, $G(t)G^{-1}(t_0)$ also belongs to $GL(n, \mathbb{R})$. When t_0 is put very close to t , $G(t)G^{-1}(t_0)$ is very close to the identity I_n . Moreover,

$$A(t) := \left. \frac{\partial}{\partial t} [G(t)G^{-1}(t_0)] \right|_{t_0=t} = \dot{G}(t)G^{-1}(t) \quad (7)$$

defines a string of tangent vectors on the tangent space at the group identity of the group manifold, more precisely, a continuously singly parametrized series of one-dimensional sub-algebra of the real Lie algebra $gl(n, \mathbb{R})$.

Differentiating Eq. (6), setting $t_0 = t$ and then using Eq. (7) yields

$$X(t)' = A(t)X(t). \quad (8)$$

The flow generated by such a $gl(n, \mathbb{R})$ vector field is the congruence of curves resulting from solving the system (8). Essentially, it is very important that when one wants to establish a local

coordinate on a smooth manifold on which a finite dimensional Lie-group of transformations is acting, the solution of the differential equations system is necessary.

The Lie-group method possesses a greater advantage than other numerical methods due to its Lie-group structure, and it is a powerful technique to solve ordinary differential equations (ODEs). Here we give a new form of the dynamics in Eq. (4a) or (4b) from the $GL(n, \mathbb{R})$ Lie-group structure. In order to fit the format in Eq. (8), the vector field \mathbf{f}_i on the right-hand side of Eq. (4a) or (4b) can be written as

$$\begin{cases} \mathbf{x}'_1 &= \mathbf{A}_1(t, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)\mathbf{x}_1 & (9a) \\ \mathbf{x}'_2 &= \mathbf{A}_2(t, \mathbf{x}_1, \mathbf{x}_2)\mathbf{x}_2 & (9b) \end{cases}$$

where $\mathbf{A}_i = \frac{\mathbf{f}_i}{\|\mathbf{x}_i\|} \otimes \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$ is the coefficient matrix and $\mathbf{a} \otimes \mathbf{b}$ denotes the dyadic operation of \mathbf{a} and \mathbf{b} , i.e., $(\mathbf{a} \otimes \mathbf{b})\mathbf{c} = (\mathbf{b} \cdot \mathbf{c})\mathbf{a}$. Here we suppose that $\|\mathbf{x}_i\| > 0$.

2.3. An implicit $GL(n, \mathbb{R})$ Lie-group method

The equation (9) provides a new starting point for constructing the Lie-group $GL(n, \mathbb{R})$ approach. In order to develop a numerical scheme, we suppose that the coefficient matrix \mathbf{A}_i is constant with

$$\bar{\mathbf{a}}_i = \frac{\bar{\mathbf{f}}_i}{\|\bar{\mathbf{x}}_i\|}, \bar{\mathbf{b}}_i = \frac{\bar{\mathbf{x}}_i}{\|\bar{\mathbf{x}}_i\|}, \quad (10)$$

which can be obtained by taking the values of \mathbf{f}_i and \mathbf{x}_i at a suitable mid-point of $\bar{t} \in [t_0 = 0, t]$, where $t \leq t_0 + h$ and h is a small time stepsize.

By introducing new variables $w_i(t) = \bar{\mathbf{b}}_i^T \cdot \mathbf{x}_i(t)$, we obtain an augmented dynamical system

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}_i(t) \\ w_i(t) \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \bar{\mathbf{a}}_i \\ \mathbf{0} & \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i \end{pmatrix} \begin{pmatrix} \mathbf{x}_i(t) \\ w_i(t) \end{pmatrix}. \quad (11)$$

Because the state matrix in Eq. (11) can be decomposed as

$$\begin{pmatrix} \mathbf{0} & \bar{\mathbf{a}}_i \\ \mathbf{0} & \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & \bar{\mathbf{a}}_i \\ \mathbf{0} & \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \bar{\mathbf{a}}_i \\ \mathbf{0} & \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i \end{pmatrix}^{-1}, \quad (12)$$

the solution of the linear dynamical system can be written analytically as follows

$$\begin{pmatrix} \mathbf{x}_i(t) \\ w_i(t) \end{pmatrix} = \exp \left[\begin{pmatrix} \mathbf{0} & \bar{\mathbf{a}}_i \\ \mathbf{0} & \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i \end{pmatrix} (t - t_k) \right] \begin{pmatrix} \mathbf{x}_i(t_k) \\ w_i(t_k) \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & \rho(c_i, t - t_k) \bar{\mathbf{a}}_i \\ \mathbf{0} & \exp[\bar{c}_i(t - t_k)] \end{pmatrix} \begin{pmatrix} \mathbf{x}_i(t_k) \\ w_i(t_k) \end{pmatrix}, \quad (13)$$

where $\rho(u, t) = \frac{e^{ut} - 1}{u}$, $c_i = \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_i$. We reduce the augmented $\mathbf{x}_i - w_i$ system to the \mathbf{x}_i system, obtaining

$$\mathbf{x}_i(t) = [\mathbf{I}_n + \rho(c_i, t - t_k) \bar{\mathbf{a}}_i \bar{\mathbf{b}}_i^T] \mathbf{x}_i(t_k) = \mathbf{G}_i(t - t_k) \mathbf{x}_i(t_k), \quad (14)$$

and $\det \mathbf{G}_i(t - t_k) > 0$. Note that \mathbf{G}_i is indeed an element in the Lie group $GL(n, \mathbb{R})$. Therefore, by introducing the auxiliary variable w_i and the augmented system, we are able to derive an explicit expression for the components of a one-parameter group $GL(n, \mathbb{R})$.

3. MELGDAE method for index-3 Hessenberg-DAEs

In this section, we present a novel modified extended Lie group DAE (MELGDAE) method for Hessenberg-DAEs with an index of 3, and provide a theoretical justification for the proposed method.

We first calculate the solution $\mathbf{x}_2(t_{k+1})$ by the given variables $\mathbf{x}_1(t_k)$, $\mathbf{x}_2(t_k)$, then obtain the solution $\mathbf{x}_1(t_{k+1})$ via given variables $\mathbf{x}_1(t_k)$, $\mathbf{x}_3(t_k)$ and $\mathbf{x}_2(t_{k+1})$. Moreover, we compute the solution $\mathbf{x}_3(t_{k+1})$ by the resulting value of $\mathbf{x}_1(t_{k+1})$ and $\mathbf{x}_2(t_{k+1})$. Hereafter, we denote the loop that solves for the variable \mathbf{x}_i as the \mathbf{x}_i -loop. The MELGDAE method, which is presented in detail below, utilizes the notation $\mathbf{x}_{i[k]} = \mathbf{x}_i(t_k)$.

For the \mathbf{x}_2 -Loop:

S1: Give $0 \leq \theta_2 \leq 1$, $\mathbf{x}_{1[k]}$, $\mathbf{x}_{2[k]}$.

S2: Estimate $\bar{\mathbf{x}}_{2[k+1]}$ via Euler method as follow,

$$\bar{\mathbf{x}}_{2[k+1]} = \mathbf{x}_{2[k]} + h\mathbf{f}_2(t_k, \mathbf{x}_{1[k]}, \mathbf{x}_{2[k]}). \quad (15)$$

S3: Let $\tau_2 = t_k + \theta_2 h$.

S4: Compute $\mathbf{x}_{2[\hat{k}+1]}$:

$$\bar{\mathbf{x}}_{2[k]} = (1 - \theta_2)\mathbf{x}_{2[k]} + \theta_2\bar{\mathbf{x}}_{2[k+1]} \quad (16a)$$

$$\bar{\mathbf{a}}_{2[k]} = \frac{\bar{\mathbf{f}}_{2[k]}}{\|\bar{\mathbf{x}}_{2[k]}\|} = \frac{\mathbf{f}_2(\tau_2, \mathbf{x}_{1[k]}, \bar{\mathbf{x}}_{2[k]})}{\|\bar{\mathbf{x}}_{2[k]}\|} \quad (16b)$$

$$\bar{\mathbf{b}}_{2[k]} = \frac{\bar{\mathbf{x}}_{2[k]}}{\|\bar{\mathbf{x}}_{2[k]}\|} \quad (16c)$$

$$\bar{c}_{2[k]} = \bar{\mathbf{a}}_{2[k]} \cdot \bar{\mathbf{b}}_{2[k]} \quad (16d)$$

$$\bar{d}_{2[k]} = \mathbf{x}_{2[k]} \cdot \bar{\mathbf{b}}_{2[k]} \quad (16e)$$

$$\rho_{2[k]} = \rho(\bar{c}_{2[k]}, h) = \frac{e^{\bar{c}_{2[k]}h} - 1}{\bar{c}_{2[k]}} \quad (16f)$$

$$\hat{\mathbf{x}}_{2[k+1]} = [\mathbf{I} + \rho(\bar{c}_{2[k]}, h)\bar{\mathbf{a}}_{2[k]}\bar{\mathbf{b}}_{2[k]}^T]\mathbf{x}_{2[k]} = \mathbf{x}_{2[k]} + \rho(\bar{c}_{2[k]}, h)\bar{d}_{2[k]}\bar{\mathbf{a}}_{2[k]} \quad (16g)$$

S5: If $\hat{\mathbf{x}}_{2[k+1]}$ convergers according to a given stopping criterion

$$\|\hat{\mathbf{x}}_{2[k+1]} - \bar{\mathbf{x}}_{2[k+1]}\| < \varepsilon_2, \quad (17)$$

then let $\mathbf{x}_{2[k+1]} = \bar{\mathbf{x}}_{2[k+1]}$; otherwise, let $\bar{\mathbf{x}}_{2[k+1]} = \hat{\mathbf{x}}_{2[k+1]}$ and goto S4.

For the \mathbf{x}_1 -Loop:

S1: Give $0 \leq \theta_1 \leq 1$, $\mathbf{x}_{1[k]}$, $\mathbf{x}_{3[k]}$, $\mathbf{x}_{2[k+1]}$ resulting from the \mathbf{x}_2 -Loop.

S2: Estimate $\bar{\mathbf{x}}_{1[k+1]}$ via Euler method as follow,

$$\bar{\mathbf{x}}_{1[k+1]} = \mathbf{x}_{1[k]} + h\mathbf{f}_1(t_k, \mathbf{x}_{1[k]}, \mathbf{x}_{2[k+1]}, \mathbf{x}_{3[k]}). \quad (18)$$

S3: Let $\tau_1 = t_k + \theta_1 h$.

S4: Compute $\mathbf{x}_{1[\hat{k}+1]}$:

$$\bar{\mathbf{x}}_{1[k]} = (1 - \theta_1)\mathbf{x}_{1[k]} + \theta_1\bar{\mathbf{x}}_{1[k+1]} \quad (19a)$$

$$\bar{\mathbf{a}}_{1[k]} = \frac{\bar{\mathbf{f}}_{1[k]}}{\|\bar{\mathbf{x}}_{1[k]}\|} = \frac{\mathbf{f}_1(\tau_1, \bar{\mathbf{x}}_{1[k]}, \mathbf{x}_{2[k+1]}, \mathbf{x}_{3[k]})}{\|\bar{\mathbf{x}}_{1[k]}\|} \quad (19b)$$

$$\bar{\mathbf{b}}_{1[k]} = \frac{\bar{\mathbf{x}}_{1[k]}}{\|\bar{\mathbf{x}}_{1[k]}\|} \quad (19c)$$

$$\bar{c}_{1[k]} = \bar{\mathbf{a}}_{1[k]} \cdot \bar{\mathbf{b}}_{1[k]} \quad (19d)$$

$$\bar{d}_{1[k]} = \mathbf{x}_{1[k]} \cdot \bar{\mathbf{b}}_{1[k]} \quad (19e)$$

$$\rho_{1[k]} = \rho(\bar{c}_{1[k]}, h) = \frac{e^{\bar{c}_{1[k]}h} - 1}{\bar{c}_{1[k]}} \quad (19f)$$

$$\hat{\mathbf{x}}_{1[k+1]} = [\mathbf{I} + \rho(\bar{c}_{1[k]}, h)\bar{\mathbf{a}}_{1[k]}\bar{\mathbf{b}}_{1[k]}^T]\mathbf{x}_{1[k]} = \mathbf{x}_{1[k]} + \rho(\bar{c}_{1[k]}, h)\bar{d}_{1[k]}\bar{\mathbf{a}}_{1[k]} \quad (19g)$$

S5: If $\hat{\mathbf{x}}_{1[k+1]}$ convergers according to a given stopping criterion

$$\|\hat{\mathbf{x}}_{1[k+1]} - \bar{\mathbf{x}}_{1[k+1]}\| < \varepsilon_1, \quad (20)$$

then let $\mathbf{x}_{1[k+1]} = \bar{\mathbf{x}}_{1[k+1]}$; otherwise, let $\bar{\mathbf{x}}_{1[k+1]} = \hat{\mathbf{x}}_{1[k+1]}$ and goto S4.

For the \mathbf{x}_3 -Loop:

S1: Let $\tau_3 = t_k + \theta_3 h, \bar{\mathbf{x}}_{3[k+1]} = \mathbf{x}_{3[k]}$.

S2: Compute $\hat{\mathbf{x}}_{3[k+1]}$ via Newton iterative method:

$$\hat{\mathbf{x}}_{3[k+1]} = \bar{\mathbf{x}}_{3[k+1]} - \mathbf{J}^{-1} \mathbf{f}_3(t_{k+1}, \mathbf{x}_2([\mathbf{x}_1(\bar{\mathbf{x}}_{3[k+1]})])), \quad (21)$$

where $\tilde{\mathbf{x}}_{1[k]} = (1 - \theta_3)\mathbf{x}_{1[k]} + \theta_3\mathbf{x}_{1[k+1]}$,

$\tilde{\mathbf{x}}_{2[k]} = (1 - \theta_3)\mathbf{x}_{2[k]} + \theta_3\mathbf{x}_{2[k+1]}$, $\mathbf{J} = \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_3} = \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2} \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} \frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3} \Big|_{\mathbf{x}_1=\tilde{\mathbf{x}}_{1[k]}, \mathbf{x}_2=\tilde{\mathbf{x}}_{2[k]}, \mathbf{x}_3=\bar{\mathbf{x}}_{3[k+1]}}$. We could obtain

$$\frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3} \Big|_{\mathbf{x}_1=\tilde{\mathbf{x}}_{1[k]}, \mathbf{x}_2=\tilde{\mathbf{x}}_{2[k]}, \mathbf{x}_3=\bar{\mathbf{x}}_{3[k+1]}}$$

$$\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} \Big|_{\mathbf{x}_1=\tilde{\mathbf{x}}_{1[k]}, \mathbf{x}_2=\tilde{\mathbf{x}}_{2[k]}} \text{ from Eq. (14).}$$

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \tilde{d}_{12[k]} \{ \tilde{\mathbf{a}}_{12[k]} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\rho}_{2[k]} + \tilde{\rho}_{2[k]} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} \} \end{array} \right. \quad (22a)$$

$$\tilde{\mathbf{a}}_{12[k]} = \frac{\mathbf{f}_2(t, \mathbf{x}_1, \mathbf{x}_2)}{\|\mathbf{x}_2\|} \Big|_{t=\tau_3, \mathbf{x}_1=\tilde{\mathbf{x}}_{1[k]}, \mathbf{x}_2=\tilde{\mathbf{x}}_{2[k]}} \quad (22b)$$

$$\frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} = \frac{\frac{\partial}{\partial \mathbf{x}_1} \mathbf{f}_2(t, \mathbf{x}_1, \mathbf{x}_2)}{\|\mathbf{x}_2\|} \Big|_{t=\tau_3, \mathbf{x}_1=\tilde{\mathbf{x}}_{1[k]}, \mathbf{x}_2=\tilde{\mathbf{x}}_{2[k]}} \quad (22c)$$

$$\tilde{\mathbf{b}}_{12[k]} = \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} \Big|_{\mathbf{x}_2=\tilde{\mathbf{x}}_{2[k]}} \quad (22d)$$

$$\tilde{c}_{12[k]} = \tilde{\mathbf{a}}_{12[k]} \cdot \tilde{\mathbf{b}}_{12[k]} \quad (22e)$$

$$\frac{\partial}{\partial \mathbf{x}_1} \tilde{c}_{12[k]} = \tilde{\mathbf{b}}_{12[k]}^T \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} \quad (22f)$$

$$\tilde{d}_{12[k]} = \mathbf{x}_{2[k]} \cdot \tilde{\mathbf{b}}_{12[k]} \quad (22g)$$

$$\tilde{\rho}_{2[k]} = \rho(\tilde{c}_{12[k]}, h) = \frac{e^{ut} - 1}{u} \Big|_{u=\tilde{c}_{12[k]}, t=h} \quad (22h)$$

$$\frac{\partial}{\partial \mathbf{x}_1} \tilde{\rho}_{2[k]} = \left[\frac{(ut - 1) \exp(ut) + 1}{u^2} \Big|_{u=\tilde{c}_{12[k]}, t=h} \right] \frac{\partial}{\partial \mathbf{x}_1} \tilde{c}_{12[k]} \quad (22i)$$

$$\tilde{\mathbf{x}}_{2[k+1]} = \mathbf{x}_{2[k]} + \tilde{\rho}_{2[k]} \tilde{d}_{12[k]} \tilde{\mathbf{a}}_{12[k]} \quad (22j)$$

$$\left\{ \begin{array}{lcl} \frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3} & = & \tilde{d}_{11}[k] \{ \tilde{\mathbf{a}}_{11}[k] \frac{\partial}{\partial \mathbf{x}_3} \tilde{\rho}_{11}[k] + \tilde{\rho}_{11}[k] \frac{\partial}{\partial \mathbf{x}_3} \tilde{\mathbf{a}}_{11}[k] \} \quad (23a) \\ \tilde{\mathbf{a}}_{11}[k] & = & \frac{\mathbf{f}_1(t, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)}{\|\mathbf{x}_1\|} \Big|_{t=\tau_3, \mathbf{x}_1=\tilde{\mathbf{x}}_1[k], \mathbf{x}_2=\tilde{\mathbf{x}}_2[k], \mathbf{x}_3=\tilde{\mathbf{x}}_3[k+1]} \quad (23b) \\ \frac{\partial}{\partial \mathbf{x}_3} \tilde{\mathbf{a}}_{11}[k] & = & \frac{\frac{\partial}{\partial \mathbf{x}_3} \mathbf{f}_1(t, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)}{\|\mathbf{x}_1\|} \Big|_{t=\tau_3, \mathbf{x}_1=\tilde{\mathbf{x}}_1[k], \mathbf{x}_2=\tilde{\mathbf{x}}_2[k], \mathbf{x}_3=\tilde{\mathbf{x}}_3[k+1]} \quad (23c) \\ \tilde{\mathbf{b}}_{11}[k] & = & \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} \Big|_{\mathbf{x}_1=\tilde{\mathbf{x}}_1[k]} \quad (23d) \\ \tilde{c}_{11}[k] & = & \tilde{\mathbf{a}}_{11}[k] \cdot \tilde{\mathbf{b}}_{11}[k] \quad (23e) \\ \frac{\partial}{\partial \mathbf{x}_3} \tilde{c}_{11}[k] & = & \tilde{\mathbf{b}}_{11}^T[k] \frac{\partial}{\partial \mathbf{x}_3} \tilde{\mathbf{a}}_{11}[k] \quad (23f) \\ \tilde{d}_{11}[k] & = & \mathbf{x}_{1[k]} \cdot \tilde{\mathbf{b}}_{11}[k] \quad (23g) \\ \tilde{\rho}_{11}[k] & = & \rho(\tilde{c}_{11}[k], h) = \frac{e^{ut} - 1}{u} \Big|_{u=\tilde{c}_{11}[k], t=h} \quad (23h) \\ \frac{\partial}{\partial \mathbf{x}_3} \tilde{\rho}_{11}[k] & = & \left[\frac{(ut - 1) \exp(ut) + 1}{u^2} \Big|_{u=\tilde{c}_{11}[k], t=h} \right] \frac{\partial}{\partial \mathbf{x}_3} \tilde{c}_{11}[k] \quad (23i) \\ \tilde{\mathbf{x}}_{1[k+1]} & = & \mathbf{x}_{1[k]} + \tilde{\rho}_{11}[k] \tilde{d}_{11}[k] \tilde{\mathbf{a}}_{11}[k] \quad (23j) \end{array} \right.$$

S3: If $\hat{\mathbf{x}}_{3[k+1]}$ convergers according to a given stopping criterion

$$\|\hat{\mathbf{x}}_{3[k+1]} - \tilde{\mathbf{x}}_{3[k+1]}\| < \varepsilon_3, \quad (24)$$

then let $\mathbf{x}_{3[k+1]} = \tilde{\mathbf{x}}_{3[k+1]}$; otherwise, let

$$\tilde{\mathbf{x}}_{3[k+1]} = \hat{\mathbf{x}}_{3[k+1]},$$

$$\mathbf{x}_{2[k+1]} = \tilde{\mathbf{x}}_{2[k+1]},$$

$$\mathbf{x}_{1[k+1]} = \tilde{\mathbf{x}}_{1[k+1]},$$

and goto S2.

Lemma 1. ([23]) Let \mathbf{U} be an $n \times m$ matrix and \mathbf{V} be an $m \times n$ matrix. Then

$$\lambda^m |\lambda \mathbf{I}_n + \mathbf{UV}| = \lambda^n |\lambda \mathbf{I}_m + \mathbf{VU}|, \quad (25)$$

where λ is any scalar and \mathbf{I}_k denotes the $k \times k$ identity matrix.

Lemma 2. Let \mathbf{U} be an $n \times m$ matrix, \mathbf{V} be an $m \times n$ matrix, and \mathbf{a} and \mathbf{b} be non-zero $m \times 1$ vectors. Define the $m \times m$ matrix $\mathbf{E} = \mathbf{I}_m + h\mathbf{a}\mathbf{b}^T$ for $h > 0$. If \mathbf{UV} is nonsingular and $h < \frac{1}{\|\mathbf{Q}\|_2 \cdot \|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2}$, where $\mathbf{Q} = \mathbf{V}(\mathbf{UV})^{-1}\mathbf{U}$, then \mathbf{UEV} is nonsingular.

Proof. To prove that \mathbf{UEV} is nonsingular, we need to show that its determinant is nonzero.

To begin with, \mathbf{UV} is nonsingular, that is, $n \leq m$. And it is noted that $\mathbf{Q}^2 = \mathbf{Q}$, then $\|\mathbf{Q}\|_2 \geq 1$.

For case $n = m$, we obtain

$$\begin{aligned}\det(\mathbf{UEV}) &= \det(\mathbf{U}) \det(\mathbf{E}) \det(\mathbf{V}) \\ &= \det(\mathbf{U}) \det(\mathbf{V}) \det(\mathbf{E}) \\ &= \det(\mathbf{UV}) \det(\mathbf{E})\end{aligned}$$

and using Lemma 1, we have

$$\det(\mathbf{E}) = 1 + h\mathbf{b}^T \mathbf{a} > 0,$$

where $h < \frac{1}{\|\mathbf{Q}\|_2 \cdot \|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2}$.

Since both $\det(\mathbf{UV})$ and $\det(\mathbf{E})$ are nonzero, then $\det(\mathbf{UEV})$ is also nonzero. Therefore \mathbf{UEV} is nonsingular.

For case $n < m$, using Lemma 1, we obtain

$$\det(\mathbf{UEV}) = \det(\mathbf{UV} + h\mathbf{Uab}^T \mathbf{V}) \quad (26)$$

$$= \det(\mathbf{UV}) \det(\mathbf{I}_n + h(\mathbf{UV})^{-1} \mathbf{Uab}^T \mathbf{V}) \quad (27)$$

$$= \det(\mathbf{UV}) (1 + h\mathbf{b}^T \mathbf{V}(\mathbf{UV})^{-1} \mathbf{Ua}) \quad (28)$$

Moreover, $|\mathbf{b}^T \mathbf{Qa}| \leq \|\mathbf{b}\|_2 \cdot \|\mathbf{Qa}\|_2 \leq \|\mathbf{Q}\|_2 \cdot \|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2$, then $1 + h\mathbf{b}^T \mathbf{Qa} > 0$, which means that \mathbf{UEV} is nonsingular from equation (28). \square

Lemma 3. Let \mathbf{U}_1 be an $n_1 \times n_3$ matrix, \mathbf{U}_2 be an $n_2 \times n_1$ matrix, and \mathbf{U}_3 be an $n_3 \times n_2$ matrix. Let \mathbf{a}_1 and \mathbf{b}_1 be nonzero $n_1 \times 1$ vectors, and let \mathbf{a}_2 and \mathbf{b}_2 be nonzero $n_2 \times 1$ vectors. Define the matrices $\mathbf{E}_1 = \mathbf{I}_{n_1} + h\mathbf{a}_1 \mathbf{b}_1^T$ and $\mathbf{E}_2 = \mathbf{I}_{n_2} + h\mathbf{a}_2 \mathbf{b}_2^T$, where $h > 0$. Suppose that $\mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1$ is nonsingular and that h satisfies

$$h < \min\left\{\frac{1}{3 \cdot \|\mathbf{Q}_1\|_2 \cdot \|\mathbf{a}_1\|_2 \cdot \|\mathbf{b}_1\|_2}, \frac{1}{3 \cdot \|\mathbf{Q}_2\|_2 \cdot \|\mathbf{a}_2\|_2 \cdot \|\mathbf{b}_2\|_2}, \frac{1}{2\|\mathbf{M}\|_2}\right\},$$

where $\mathbf{Q}_1 = \mathbf{U}_1(\mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1)^{-1} \mathbf{U}_3 \mathbf{U}_2$, $\mathbf{Q}_2 = \mathbf{U}_2 \mathbf{U}_1(\mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1)^{-1} \mathbf{U}_3$, and $\mathbf{M} = \mathbf{U}_1(\mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1)^{-1} \mathbf{U}_3 \mathbf{a}_2 \mathbf{b}_2^T \mathbf{U}_2$. Then, the matrix $\mathbf{U}_3 \mathbf{E}_2 \mathbf{U}_2 \mathbf{E}_1 \mathbf{U}_1$ is nonsingular.

Proof. Firstly, $\mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1$ is nonsingular, that is $n_3 \leq n_1$ and $n_3 \leq n_2$.

Let

$$\mathbf{V}_2 = \mathbf{U}_2 \mathbf{U}_1,$$

that is $\mathbf{U}_3 \mathbf{V}_2$ is nonsingular. And $\mathbf{E}_2 = \mathbf{I}_{n_2} + h\mathbf{a}_2 \mathbf{b}_2^T$ and $h < \frac{1}{3\|\mathbf{Q}_2\|_2 \cdot \|\mathbf{a}_2\|_2 \cdot \|\mathbf{b}_2\|_2}$, where $\mathbf{Q}_2 = \mathbf{U}_2 \mathbf{U}_1(\mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1)^{-1} \mathbf{U}_3$. By Lemma 2, $\mathbf{U}_3 \mathbf{E}_2 \mathbf{V}_2$ is nonsingular, which means $\mathbf{U}_3 \mathbf{E}_2 \mathbf{U}_2 \mathbf{U}_1$ is nonsingular.

Then, let

$$\mathbf{U}' = \mathbf{U}_3 \mathbf{E}_2 \mathbf{U}_2,$$

and $\mathbf{U}' \mathbf{U}_1$ is nonsingular. In addition, $\mathbf{E}_1 = \mathbf{I}_{n_1} + h\mathbf{a}_1 \mathbf{b}_1^T$, if $h < \frac{1}{\|\mathbf{Q}'_1\|_2 \cdot \|\mathbf{a}_1\|_2 \cdot \|\mathbf{b}_1\|_2}$, where $\mathbf{Q}'_1 = \mathbf{U}_1(\mathbf{U}_3 \mathbf{E}_2 \mathbf{U}_2 \mathbf{U}_1)^{-1} \mathbf{U}_3 \mathbf{E}_2 \mathbf{U}_2$. By Lemma 2, $\mathbf{U}' \mathbf{E}_1 \mathbf{U}_1$ is nonsingular, which means $\mathbf{U}_3 \mathbf{E}_2 \mathbf{U}_2 \mathbf{E}_1 \mathbf{U}_1$ is nonsingular.

Now, we need proof

$$h\|\mathbf{Q}'_1\|_2 \cdot \|\mathbf{a}_1\|_2 \cdot \|\mathbf{b}_1\|_2 < 1. \quad (29)$$

It is noted that

$$\mathbf{Q}'_1 = (\mathbf{I}_{n_1} + h\mathbf{M})^{-1} \mathbf{Q}_1 (\mathbf{I}_{n_1} + h\mathbf{M}), \quad (30)$$

where $h < \frac{1}{2\|\mathbf{M}\|_2}$. Moreover, we are able to obtain $\|(\mathbf{I}_{n_1} + h\mathbf{M})^{-1}\|_2 \leq \frac{1}{1 - \|h\mathbf{M}\|_2} < 2$ and $\|\mathbf{I}_{n_1} + h\mathbf{M}\|_2 \leq \|\mathbf{I}_{n_1}\|_2 + \|h\mathbf{M}\|_2 < \frac{3}{2}$.

Then, we obtain

$$\|\mathbf{Q}'_1\|_2 = \|(\mathbf{I}_{n_1} + h\mathbf{M})^{-1}\mathbf{Q}_1(\mathbf{I}_{n_1} + h\mathbf{M})\|_2 \quad (31)$$

$$\leq \|(\mathbf{I}_{n_1} + h\mathbf{M})^{-1}\|_2 \cdot \|\mathbf{Q}_1\|_2 \cdot \|\mathbf{I}_{n_1} + h\mathbf{M}\|_2 \quad (32)$$

$$< 3\|\mathbf{Q}_1\|_2 \quad (33)$$

For $h < \frac{1}{3\|\mathbf{Q}_1\|_2 \cdot \|\mathbf{a}_1\|_2 \cdot \|\mathbf{b}_1\|_2}$, then we obtain the inequality (29). \square

Theorem 1. For the x_3 -Loop of Hessenberg DAEs with index 3, if h is sufficiently small, then the Jacobian matrix \mathbf{J} is nonsingular.

Proof. We first compute the $\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1}$ by Eq.(??).

$$\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \tilde{d}_{12[k]} \left\{ \tilde{\mathbf{a}}_{12[k]} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\rho}_{2[k]} + \tilde{\rho}_{2[k]} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} \right\} \quad (34)$$

$$= \tilde{d}_{12[k]} \left\{ \tilde{\mathbf{a}}_{12[k]} \left[\frac{(ut-1)\exp(ut)+1}{u^2} \right]_{u=\tilde{c}_{12[k]}, t=h} \frac{\partial}{\partial \mathbf{x}_1} \tilde{c}_{12[k]} + \tilde{\rho}_{2[k]} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} \right\} \quad (35)$$

$$= \tilde{d}_{12[k]} \left\{ \tilde{\mathbf{a}}_{12[k]} \frac{(\tilde{c}_{12[k]}h-1)\exp(\tilde{c}_{12[k]}h)+1}{\tilde{c}_{12[k]}^2} [\tilde{\mathbf{b}}_{12[k]}^T \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]}] + \tilde{\rho}_{2[k]} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} \right\} \quad (36)$$

$$= \tilde{d}_{12[k]} \left\{ \tilde{\rho}_{2[k]} \mathbf{I}_{n_2} + \frac{(\tilde{c}_{12[k]}h-1)\exp(\tilde{c}_{12[k]}h)+1}{\tilde{c}_{12[k]}^2} \tilde{\mathbf{a}}_{12[k]} \tilde{\mathbf{b}}_{12[k]}^T \right\} \frac{\partial}{\partial \mathbf{x}_1} \tilde{\mathbf{a}}_{12[k]} \quad (37)$$

$$= \frac{\tilde{d}_{12[k]}}{\|\tilde{\mathbf{x}}_{2[k]}\|} \left\{ \tilde{\rho}_{2[k]} \mathbf{I}_{n_2} + \frac{(\tilde{c}_{12[k]}h-1)\exp(\tilde{c}_{12[k]}h)+1}{\tilde{c}_{12[k]}^2} \tilde{\mathbf{a}}_{12[k]} \tilde{\mathbf{b}}_{12[k]}^T \right\} \frac{\partial \mathbf{f}_2(t, \mathbf{x}_1, \mathbf{x}_2)}{\partial \mathbf{x}_1}. \quad (38)$$

Similarly, we obtain the $\frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3}$ from Eq. (??).

$$\frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3} = \tilde{d}_{11[k]} \left\{ \tilde{\mathbf{a}}_{11[k]} \frac{\partial}{\partial \mathbf{x}_3} \tilde{\rho}_{1[k]} + \tilde{\rho}_{1[k]} \frac{\partial}{\partial \mathbf{x}_3} \tilde{\mathbf{a}}_{11[k]} \right\} \quad (39)$$

$$= \tilde{d}_{11[k]} \left\{ \tilde{\rho}_{1[k]} \mathbf{I}_{n_1} + \frac{(\tilde{c}_{11[k]}h-1)\exp(\tilde{c}_{11[k]}h)+1}{\tilde{c}_{11[k]}^2} \tilde{\mathbf{a}}_{11[k]} \tilde{\mathbf{b}}_{11[k]}^T \right\} \frac{\partial}{\partial \mathbf{x}_3} \tilde{\mathbf{a}}_{11[k]} \quad (40)$$

$$= \frac{\tilde{d}_{11[k]}}{\|\tilde{\mathbf{x}}_{1[k]}\|} \left\{ \tilde{\rho}_{1[k]} \mathbf{I}_{n_1} + \frac{(\tilde{c}_{11[k]}h-1)\exp(\tilde{c}_{11[k]}h)+1}{\tilde{c}_{11[k]}^2} \tilde{\mathbf{a}}_{11[k]} \tilde{\mathbf{b}}_{11[k]}^T \right\} \frac{\partial \mathbf{f}_1(t, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)}{\partial \mathbf{x}_3}. \quad (41)$$

Futhermore, we are able to get

$$\mathbf{J} = \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_3} = \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2} \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} \frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_3} \quad (42)$$

$$= \frac{\tilde{d}_{12[k]}}{\|\tilde{\mathbf{x}}_{2[k]}\|} \frac{\tilde{d}_{11[k]}}{\|\tilde{\mathbf{x}}_{1[k]}\|} \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2} \mathbf{E}_2 \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} \mathbf{E}_1 \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3}, \quad (43)$$

where $\mathbf{E}_{i[k]} = \tilde{\rho}_{i[k]} \mathbf{I}_{n_i} + \frac{(\tilde{c}_{1i[k]}h-1)\exp(\tilde{c}_{1i[k]}h)+1}{\tilde{c}_{1i[k]}^2} \tilde{\mathbf{a}}_{1i[k]} \tilde{\mathbf{b}}_{1i[k]}^T$ ($i = 1, 2$), and $\mathbf{F} = \mathbf{U}_3 \mathbf{U}_2 \mathbf{U}_1$ ($\mathbf{U}_3 =$

$\frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2}$, $\mathbf{U}_2 = \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1}$, $\mathbf{U}_1 = \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3}$) is nonsingular.

From Lemma 1, it is noted that

$$|\mathbf{E}_{i[k]}| = \bar{\rho}_{i[k]}^{n_i-1} \{ \bar{\rho}_{i[k]} + \frac{(\tilde{c}_{1i[k]}h - 1) \exp(\tilde{c}_{1i[k]}h) + 1}{\tilde{c}_{1i[k]}^2} \tilde{\mathbf{b}}_{1i[k]}^T \tilde{\mathbf{a}}_{1i[k]} \} = h \bar{\rho}_{i[k]}^{n_i-1} \exp(\tilde{c}_{1i[k]}h) \neq 0.$$

For small $h < H$, $\mathbf{E}_{i[k]} = h\mathbf{I}_{n_i} + h^2\tilde{\mathbf{a}}_{1i[k]}\tilde{\mathbf{b}}_{1i[k]}^T + O(h^3) \approx h\mathbf{I}_{n_i} + h^2\tilde{\mathbf{a}}_{1i[k]}\tilde{\mathbf{b}}_{1i[k]}^T$. Moreover, by using Lemma 3, if

$$h < \min_k \left\{ \frac{1}{3\|\mathbf{Q}_{1[k]}\|_2 \cdot \|\tilde{\mathbf{a}}_{11[k]}\|_2 \cdot \|\tilde{\mathbf{b}}_{11[k]}\|_2}, \frac{1}{3\|\mathbf{Q}_{2[k]}\|_2 \cdot \|\tilde{\mathbf{a}}_{12[k]}\|_2 \cdot \|\tilde{\mathbf{b}}_{12[k]}\|_2}, \frac{1}{2\|\mathbf{M}_k\|_2} \right\},$$

where $\mathbf{Q}_{1[k]} = \mathbf{U}_1(\mathbf{U}_3\mathbf{E}_2\mathbf{U}_2\mathbf{U}_1)^{-1}\mathbf{U}_3\mathbf{E}_2\mathbf{U}_2$, $\mathbf{Q}_{2[k]} = \mathbf{U}_2\mathbf{U}_1(\mathbf{U}_3\mathbf{U}_2\mathbf{U}_1)^{-1}\mathbf{U}_3$ and

$\mathbf{M}_k = \mathbf{U}_1(\mathbf{U}_3\mathbf{U}_2\mathbf{U}_1)^{-1}\mathbf{U}_3\tilde{\mathbf{a}}_{12[k]}\tilde{\mathbf{b}}_{12[k]}^T\mathbf{U}_2$, then $\mathbf{U}_3\mathbf{E}_2\mathbf{U}_2\mathbf{E}_1\mathbf{U}_1$ is nonsingular. Therefore, satisfying for small h with

$$h < \min_k \left\{ H, \frac{1}{3\|\mathbf{Q}_{1[k]}\|_2 \cdot \|\tilde{\mathbf{a}}_{11[k]}\|_2 \cdot \|\tilde{\mathbf{b}}_{11[k]}\|_2}, \frac{1}{3\|\mathbf{Q}_{2[k]}\|_2 \cdot \|\tilde{\mathbf{a}}_{12[k]}\|_2 \cdot \|\tilde{\mathbf{b}}_{12[k]}\|_2}, \frac{1}{2\|\mathbf{M}_k\|_2} \right\},$$

we are able to obtain \mathbf{J} is nonsingular. \square

4. Numerical Examples

In this section, we do numerical simulation experiments to illustrate the performance of the MELGDAE. All codes are written under Windows 10 system and run on a personal computer with Intel(R) Core(TM) i5-3570 CPU @ 3.40 GHz, 4.00 GB RAM and 64-bit operating system.

We consider a Hessenberg DAEs system with index 3 as follows, which can be found in [24].

$$\begin{cases} z_1'(t) = g_1 = (z_3(t)z_4(t) + z_1(t)z_2(t))z_5(t), & (44a) \\ z_2'(t) = g_2 = -z_3(t)z_4(t)^2z_2(t)^2z_5(t), & (44b) \\ z_3'(t) = g_3 = 2z_3(t)z_4(t)z_1(t)z_2(t), & (44c) \\ z_4'(t) = g_4 = -z_3(t)z_4(t)z_2(t)^2, & (44d) \\ 0 = g_5 = z_3(t)z_4(t)^2 - 1, & (44e) \end{cases}$$

where $t \in [0, 1]$, $z_1(0) = z_2(0) = z_3(0) = z_4(0) = z_5(0) = 1$ are the consistent initial values. And the exact solution is given by

$$z_1(t) = z_3(t) = e^{2t}, z_2(t) = z_4(t) = e^{-t}, z_5(t) = e^t.$$

Let $\mathbf{x}_1(t) = (z_1(t), z_2(t))^T$, $\mathbf{x}_2(t) = (z_3(t), z_4(t))^T$, $\mathbf{x}_3(t) = z_5(t)$, and $\mathbf{f}_1 = (g_1, g_2)^T$, $\mathbf{f}_2 = (g_3, g_4)^T$, $\mathbf{f}_3 = g_5$, then $\mathbf{F} = \frac{\partial \mathbf{f}_3}{\partial \mathbf{x}_2} \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_3} = 2z_2(t)(2z_2(t)^2z_3(t) + 1)$.

To obtain DAEs with index 2, we differentiate the constraint g_5 to obtain the following equation.

$$0 = g_6 = z_1(t)z_4(t) - z_2(t)z_3(t). \quad (45)$$

Then, we are able to obtain index 2 DAEs below.

$$\begin{cases} z_1'(t) = g_1 = (z_3(t)z_4(t) + z_1(t)z_2(t))z_5(t), & (46a) \\ z_2'(t) = g_2 = -z_3(t)z_4(t)^2z_2(t)^2z_5(t), & (46b) \\ z_3'(t) = g_3 = 2z_3(t)z_4(t)z_1(t)z_2(t), & (46c) \\ z_4'(t) = g_4 = -z_3(t)z_4(t)z_2(t)^2, & (46d) \\ 0 = g_6 = z_1(t)z_4(t) - z_2(t)z_3(t). & (46e) \end{cases}$$

We compared the numerical results of the MELGDAE method with those of RADAU, a code based on implicit Runge-Kutta methods, and MEBDFI, a code based on the MEBDF method that has better theoretical properties than BDF methods, to demonstrate the capability of the MELGDAE method. The comparison was made using the Hessenberg DAEs with index 2 or 3 mentioned above.

Without loss of generality, we assume the absolute error ($\epsilon_1, \epsilon_2, \epsilon_3$) is 10^{-8} , the relative error is 10^{-5} , the (initial) step size is $h = 10^{-3}$, $\theta_1 = \theta_2 = \theta_3 = 0.5$. Based on Figures 1 and 2, the MELGDAE method showed comparable accuracy with the RADAU method for the differential variables $z_1(t)$ and $z_3(t)$ for both index-2 and index-3 Hessenberg DAEs. In contrast, the MEBDFI method was less accurate for these variables. For the algebraic variable $z_5(t)$, both RADAU and MEBDFI methods provided more accurate results than the MELGDAE method. However, for the algebraic constraints $g_5(t)$ and $g_6(t)$, the MELGDAE method exhibited significantly higher accuracy compared to the RADAU and MEBDFI methods.

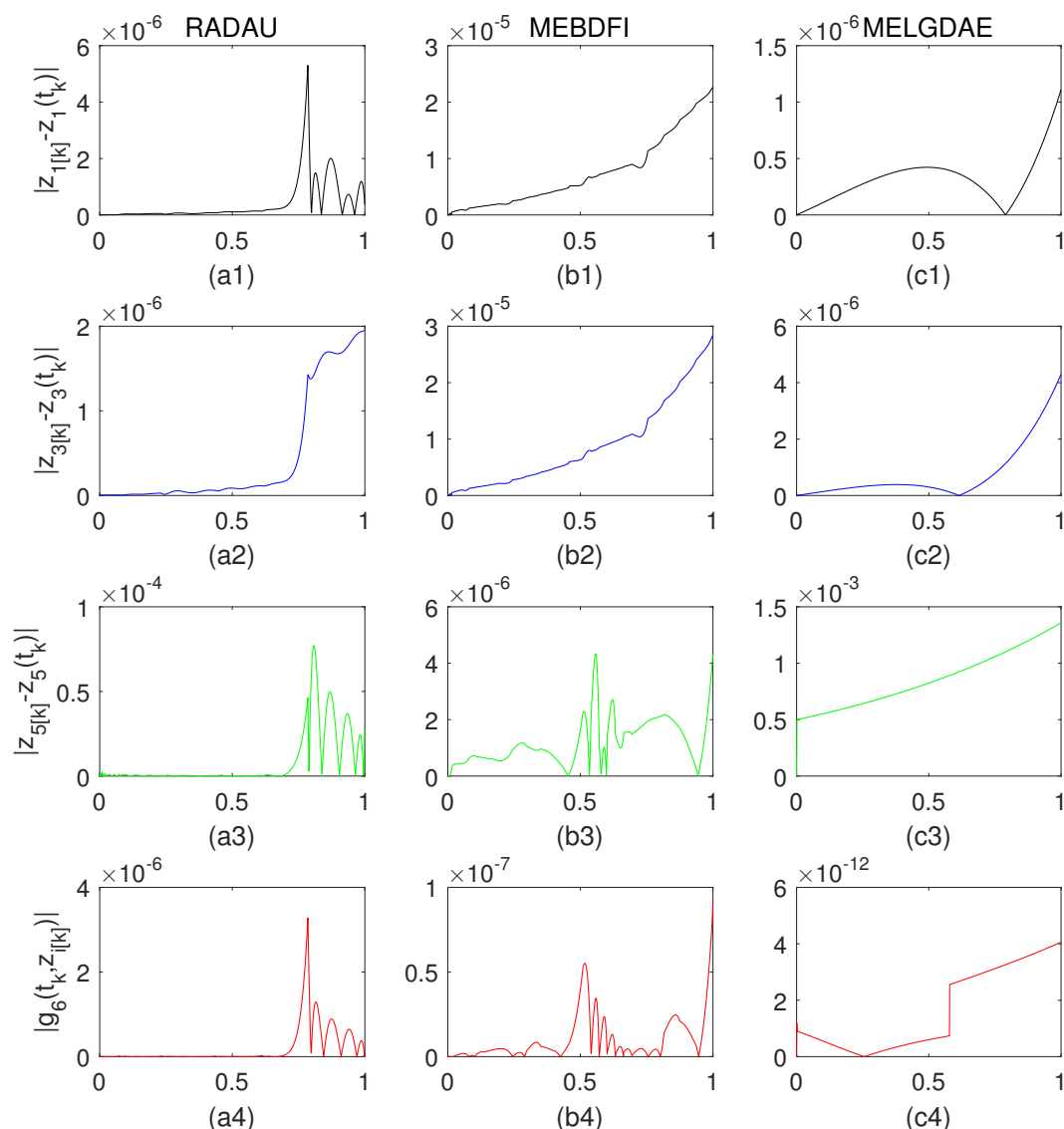


Figure 1. The absolute errors of three methods for index -2 DAEs. (a1)-(a4) are the absolute errors of $z_1(t)$, $z_3(t)$, $z_5(t)$ and $g_6(t)$ of RADAU respectively; (b1)-(b4) are the absolute errors of MEBDFI; (c1)-(c4) are the absolute errors of MELGDAE.

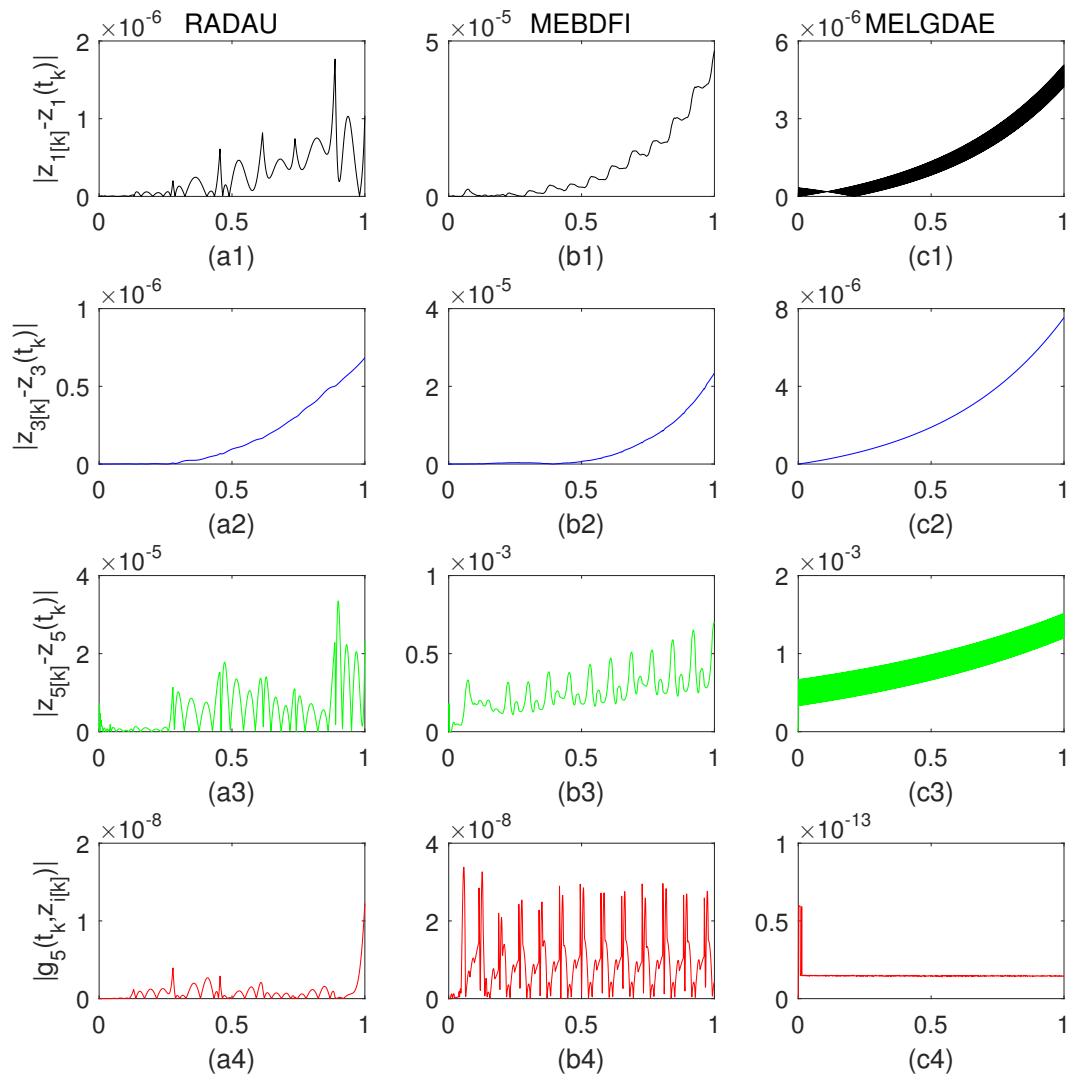


Figure 2. The absolute errors of three methods for Index 3 DAEs. (a1)-(a4) are the absolute errors of $z_1(t)$, $z_3(t)$, $z_5(t)$ and $g_5(t)$ of RADAU respectively; (b1)-(b4) are the absolute errors of MEBDFI; (c1)-(c4) are the absolute errors of MELGDAE.

Upon observing Figures 1 and 2, it is clear that all three methods exhibit a reduction in the absolute errors of $z_1(t)$, $z_3(t)$, and $z_5(t)$ as the index increases from 2 to 3. This reduction is not significant for the MELGDAE and RADAU methods. Therefore, we can conclude that the MELGDAE method has the potential to be extended for computing DAEs systems with an index higher than 3.

Furthermore, we conducted numerical experiments to verify the convergence accuracy of the MELGDAE method for index 3 DAEs. We assumed $h = \{2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}\}$ and calculated their constants in $-\log_2(\max_k |z_{i[k]} - z_i(t_k)|) = \nu(-\log_2 h) + \mu$ ($i = 1, 2, 3, 4, 5$) and $-\log_2(\max_k |g_5(t_k, z_{3[k]}, z_{4[k]})|) = \nu(-\log_2 h) + \mu$, which represents $\max_k |z_{i[k]} - z_i(t_k)| = O(h^\nu)$ and $\max_k |g_5(t_k, z_{3[k]}, z_{4[k]})| = O(h^\nu)$, using the standard least-squares method. The results are presented in Figure 3, which indicate that the MELGDAE method exhibits second-order convergence for $z_1(t)$, $z_2(t)$, $z_3(t)$, and $z_4(t)$ (all differential variables) and g_5 (the algebraic constraints), and first-order convergence for $z_5(t)$ (algebraic variable). Thus, this method may be extended to compute systems with an index of 4 or higher and is suitable for DAE systems that cannot be directly solved using other classical methods.

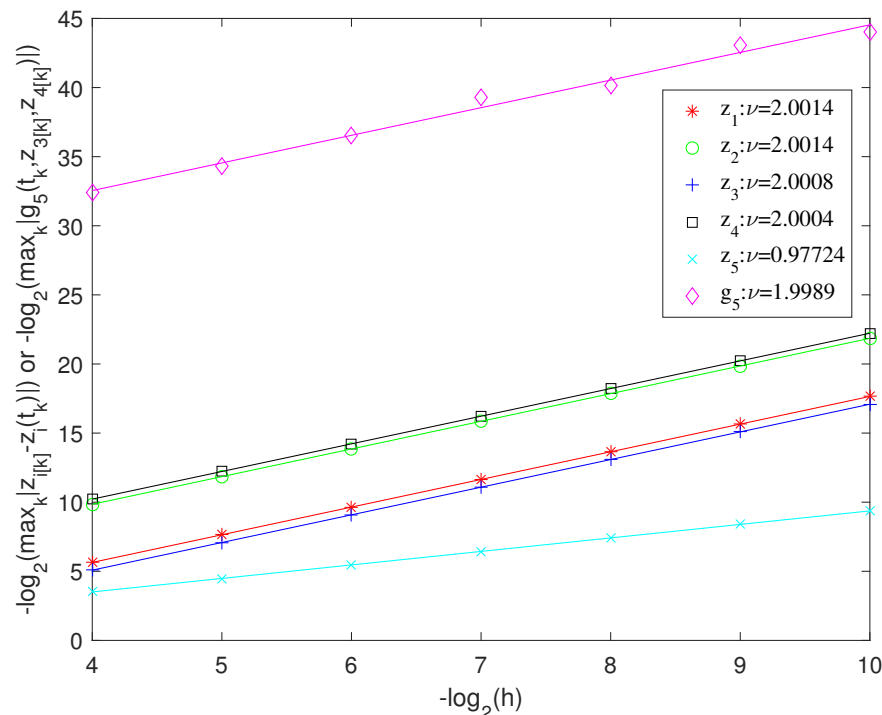


Figure 3. The convergence order $O(h^\nu)$ of MELGDAE for index 3 DAEs.

5. Conclusions

The Modified Extended Lie Group Differential Algebraic Equations (MELGDAE) method was developed in this paper to solve Hessenberg-DAEs with index 3. While the LGDAE method presented in [18] is only capable of handling index 2 systems, the MELGDAE method can solve higher index systems by taking into account the differential equation components with respect to the index in Hessenberg form. It is worth mentioning that the MELGDAE method demonstrates superior theoretical properties compared to the LGDAEs, as illustrated in Theorem 1.

We conducted a comparison of the MELGDAE method with the standard methods RADAU and MEBDFI on index 2 and 3 DAE systems. The MELGDAE method demonstrated promising results in terms of solution errors when compared to RADAU and MEBDFI, which are known to be powerful on different DAE systems. However, the MELGDAE method excelled in preserving the algebraic constraints. Moreover, all differential variables in index 3 Hessenberg-DAEs exhibited order 2 convergence with the MELGDAE method, whereas MEBDFI in [24] and RADAU in [7,25] demonstrated order reduction. Although the MELGDAE method has high stability and accuracy when solving strong nonlinear Hessenberg DAEs with index 3, there are still some theoretical gaps that need to be addressed in the future. Additionally, the MELGDAE method can be further extended to solve DAEs with an index of 4 or higher, which cannot be directly solved using standard methods.

Author Contributions: J.T. contributed to supervision, methodology, validation, funding acquisition and project administration. J.L. contributed to investigation, computations, funding acquisition and validation.

Funding: This work was partly supported by the NSF of China (No: 12201144), the Science and Technology Program of Guangzhou (No: 202002030138), the Guangdong Basic and Applied Basic Research Foundation (No: 2020A151110554), the Science and Technology Foundation of Guizhou Province (No: QKHJC-ZK[2021]YB015), and the University-Industry Collaborative Education Program of the Ministry of Education (No: BINTECH-KJZX-20220831-61), China.

Acknowledgments: The authors wish to thank the referee for his or her very helpful comments and useful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schiehlen, W. Multibody system dynamics: roots and perspectives. *Multibody System Dynamics* **1997**, *1*, 149–188.
2. Winkler, R. Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *Journal of computational and applied mathematics* **2003**, *157*(2), 477–505.
3. Coffey, T. S.; Kelley, C. T.; Keyes, D. E. Pseudotransient continuation and differential-algebraic equations. *SIAM Journal on Scientific Computing* **2003**, *25*(2), 553–569.
4. Peng, H. J.; Li, F.; Liu, J. G.; Ju, Z. J. A symplectic instantaneous optimal control for robot trajectory tracking with differential-algebraic equation models. *IEEE Transactions on Industrial Electronics* **2019**, *67*(5), 3819–3829.
5. Çelik, E.; Karaduman, E.; Bayram, M. Numerical solutions of chemical differential-algebraic equations. *Applied Mathematics and Computation* **2003**, *139*(2), 259–264.
6. Ascher, U. M.; Petzold, L. R. Projected implicit Runge–Kutta methods for differential-algebraic equations. *SIAM Journal on Numerical Analysis* **1991**, *28*(4), 1097–1120.
7. Hairer, E.; Wanner, G. Stiff differential equations solved by radau methods. *Journal of Computational and Applied Mathematics* **1999**, *111*, 93–111.
8. Skvortsov, L. M. Diagonally implicit Runge–Kutta methods for differential algebraic equations of indices two and three. *Computational Mathematics and Mathematical Physics* **2010**, *50*, 993–1005.
9. Cash, J.; Considine, S. An mebf code for stiff initial value problems. *ACM Trans. Math. Softw.* **1992**, 142–158.
10. Cash, J. R. Modified extended backward differentiation formulae for the numerical solution of stiff initial value problems in ODEs and DAEs. *Journal of Computational and Applied Mathematics* **2000**, *125*(1), 117–130.
11. Karta, M.; Celik, E. On the numerical solution of differential-algebraic equations with Hessenberg Index-3. *Discrete Dynamics in Nature and Society* **2012**, *2012*, 12 pages.
12. Benhammouda, B. A novel technique to solve nonlinear higher-index Hessenberg differential-algebraic equations by Adomian decomposition method. *SpringerPlus* **2016**, *5*(1), 1–14.
13. Celledoni, E.; Kometa, B. K. Semi-Lagrangian multistep exponential integrators for index 2 differential-algebraic systems. *Journal of Computational Physics* **2011**, *230*(9), 3413–3429.
14. Iserles, A.; Zanna, A. Preserving algebraic invariants with Runge–Kutta methods. *Journal of Computational and Applied Mathematics* **2000**, *125*(1), 69–81.
15. Wieloch, V.; Arnold, M. BDF integrators for constrained mechanical systems on Lie groups. *Journal of Computational and Applied Mathematics* **2021**, *387*, 112517.
16. Liu, C. S. Preserving constraints of differential equations by numerical methods based on integrating factors. *Computer Modeling in Engineering and Sciences* **2006**, *12*(2), 83–107.
17. Liu, C. S. Finding unknown heat source in a nonlinear Cauchy problem by the Lie-group differential algebraic equations method. *Engineering Analysis with Boundary Elements* **2015**, *50*, 148–156.
18. Liu, C. S.; Chen, L.; Liu, L. W. Solving mechanical systems with nonholonomic constraints by a Lie-group differential algebraic equations method. *Journal of Engineering Mechanics* **2017**, *143*(9), 04017097.
19. Liu, C. S. Lie-group differential algebraic equations method to recover heat source in a Cauchy problem with analytic continuation data. *International Journal of Heat and Mass Transfer* **2014**, *78*, 538–547.
20. Liu, C. S.; Kuo, C. L.; Chang, J. R. Recovering a heat source and initial value by a Lie-group differential algebraic equations method. *Numerical Heat Transfer, Part B: Fundamentals* **2015**, *67*(3), 231–254.
21. Liu, C. S. A new sliding control strategy for nonlinear system solved by the Lie-group differential algebraic equation method. *Communications in Nonlinear Science and Numerical Simulation* **2014**, *19*(6), 2012–2038.
22. Liu, C. S.; Chang, C. W. A real-time Lie-group differential algebraic equations method to solve the inverse nonlinear vibration problems. *Inverse Problems in Science and Engineering* **2016**, *24*(9), 1569–1586.
23. Horn, R. A.; Johnson, C. R. *Matrix Analysis*, 2nd ed.; Cambridge University Press: New York, USA, 2012; pp 66–67.
24. Jay, L. Convergence of Runge-Kutta methods for differential-algebraic systems of index 3. *Applied Numerical Mathematics* **1995**, *17*(2), 97–118.
25. Hairer, E.; Lubich, C.; Roche, M. *The numerical solution of differential-algebraic systems by Runge-Kutta methods*; Springer-Verlag: Berlin, Germany, 1989; pp 19–20.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.