Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 987905, 8 pages
http://dx.doi.org/10.1155/2013/987905

Research Article

Solving Nonlinear Differential Algebraic Equations by an Implicit $GL(n,\mathbb{R})$ Lie-Group Method

Chein-Shan Liu

Department of Civil Engineering, National Taiwan University, Taipei, Taiwan

Correspondence should be addressed to Chein-Shan Liu; liucs@ntu.edu.tw

Received 26 February 2013; Accepted 28 June 2013

Academic Editor: Alexander Timokha

We derive an implicit $GL(n,\mathbb{R})$ Lie-group algorithm together with the Newton iterative scheme to solve nonlinear differential algebraic equations. Four numerical examples are given to evaluate the efficiency and accuracy of the new method when comparing the computational results with the closed-form solutions.

1. Introduction

In this paper, we propose a novel method to solve nonlinear differential algebraic equations (DAEs), which govern the evolution of $n+m$ variables $x_i, i = 1,\ldots,n$ and $y_j, j = 1,\ldots,m$ with n nonlinear ordinary differential equations (ODEs) and m nonlinear algebraic equations (NAEs):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{y},t), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad t \in \mathbb{R},\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{y} \in \mathbb{R}^m, \tag{1}$$

$$\mathbf{F}(\mathbf{x},\mathbf{y},t) = \mathbf{0}, \quad \mathbf{F} \in \mathbb{R}^m. \tag{2}$$

Usually, n is larger than m. When $m = 0$, the DAEs reduce to the ODEs. There are many numerical methods used to solve ODEs, but only a few is used to solve DAEs [1–5]. A lot of engineering problems are modelled as a combination of ODEs and NAEs, which are abbreviated as differential algebraic equations (DAEs). The DAEs are both numerically and analytically difficult than the ODEs. Recently, there were some new methods to solve DAEs, for example, Adomian decomposition method [6, 7], variational iterative method [8], and pseudospectral method [9].

2. The $GL(n,\mathbb{R})$ Structure of Differential Equations System

The Lie-group is a differentiable manifold, which is endowed with a group structure that is compatible with the underlying topology of the manifold. The Lie-group method can provide a better algorithm that retains the orbit generated from numerical solution on the manifold which is associated with the Lie-group.

The general linear group is a Lie group, whose manifold is an open subset $GL(n,\mathbb{R}) := \{\mathbf{G} \in \mathbb{R}^{n\times n} \mid \det \mathbf{G} \neq 0\}$ of the linear space of all $n \times n$ nonsingular matrices. Thus, $GL(n,\mathbb{R})$ is an $n \times n$-dimensional manifold. The group composition is given by the matrix multiplication.

Here we give a new form of (1) from the $GL(n,\mathbb{R})$ Lie-group structure. The vector field $\mathbf{f}$ on the right-hand side of (1) can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \tag{3}$$

where

$$\mathbf{A} = \frac{\mathbf{f}}{\|\mathbf{x}\|} \otimes \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{4}$$

is the coefficient matrix. The symbol $\otimes$ in $\mathbf{u} \otimes \mathbf{y}$ denotes the dyadic operation of $\mathbf{u}$ and $\mathbf{y}$, that is, $(\mathbf{u} \otimes \mathbf{y})\mathbf{z} = \mathbf{y} \cdot \mathbf{z}\mathbf{u}$.

Because the coefficient matrix $\mathbf{A}$ is well defined, the Lie-group element $\mathbf{G}$ generated from the above dynamical system (3) with $\dot{\mathbf{G}} = \mathbf{A}\mathbf{G}$ satisfies $\det \mathbf{G}(t) \neq 0$, such that $\mathbf{G} \in GL(n,\mathbb{R})$.

3. An Implicit $GL(n,\mathbb{R})$ Lie-Group Scheme

Equation (3) is a new starting point for the development of the Lie-group $GL(n,\mathbb{R})$ algorithm. In order to develop

a numerical scheme from (3) and (4), we suppose that the coefficient matrix $\mathbf{A}$ is constant with

$$\mathbf{a} = \frac{\bar{\mathbf{f}}}{\|\bar{\mathbf{x}}\|}, \qquad \mathbf{b} = \frac{\bar{\mathbf{x}}}{\|\bar{\mathbf{x}}\|} \tag{5}$$

being two constant vectors, which can be obtained by taking the values of $\mathbf{f}$ and $\mathbf{x}$ at a suitable mid-point of $\bar{t} \in [t_0 = 0, t]$, where $t \le t_0 + h$ and h is a small time stepsize. The variable $\mathbf{y}$ is supposed to be a constant vector in this small time interval. Thus, from (3) and (4) we have

$$\dot{\mathbf{x}} = \mathbf{b} \cdot \mathbf{x}\mathbf{a}. \tag{6}$$

Let

$$w = \mathbf{b} \cdot \mathbf{x}, \tag{7}$$

and (6) becomes

$$\dot{\mathbf{x}} = w\mathbf{a}. \tag{8}$$

At the same time, from the above two equations we can derive

$$\dot{w} = cw, \tag{9}$$

where

$$c = \mathbf{a} \cdot \mathbf{b} \tag{10}$$

is viewed as a constant scalar. Thus, we have

$$w(t) = w_0 \exp(ct), \tag{11}$$

where $w_0 = \mathbf{b} \cdot \mathbf{x}_0$.

Inserting (11) for $w(t)$ into (8) and integrating the resultant equation, we can obtain

$$\mathbf{x}(t) = \left[\mathbf{I}_n + \eta(t)\,\mathbf{a}\mathbf{b}^T\right]\mathbf{x}_0, \tag{12}$$

where $\mathbf{x}_0$ is the initial value of $\mathbf{x}$ at an initial time $t = t_0 = 0$, and

$$\eta(t) = \frac{e^{ct} - 1}{c}. \tag{13}$$

Let $\mathbf{G}(t)$ be the coefficient matrix before $\mathbf{x}_0$ in (12), that is,

$$\mathbf{G} = \mathbf{I}_n + \eta(t)\,\mathbf{a}\mathbf{b}^T, \tag{14}$$

which is one sort of elementary matrices. According to [10, 11], one can prove

$$\det \mathbf{G} = e^{ct} > 0, \tag{15}$$

which means that $\mathbf{G}$ is a Lie-group element of $GL(n, \mathbb{R})$.

Within a small time step we can suppose that the variables y_j, $j = 1, \ldots, m$ are constant in the interval of $t_k < t < t_{k+1}$. As a consequence, we can develop the following implicit scheme for solving the ODEs (1) where $\mathbf{y}$ at the kth time step, denoted by $\mathbf{y}_k$, is viewed as a parameter.

(i) Give $0 \le \theta \le 1$.

(ii) Give an initial $\mathbf{x}_0$ at an initial time $t = t_0$ and a time stepsize h.

(iii) For $k = 0, 1, \ldots$, we repeat the following computations to a terminal time $t = t_f$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}_k, \tag{16}$$

where $\mathbf{f}_k := \mathbf{f}(\mathbf{x}_k, \mathbf{y}_k, t_k)$. With the above $\mathbf{x}_{k+1}$ generated from an Euler step as an initial guess, we can iteratively solve the new $\mathbf{x}_{k+1}$ by

$$\begin{aligned}
\bar{t}_k &= t_k + \theta h,\\
\bar{\mathbf{x}}_k &= (1-\theta)\,\mathbf{x}_k + \theta\mathbf{x}_{k+1},\\
\bar{\mathbf{f}}_k &= \mathbf{f}\left(\bar{\mathbf{x}}_k, \mathbf{y}_k, \bar{t}_k\right),\\
\mathbf{a}_k &= \frac{\bar{\mathbf{f}}_k}{\left\|\bar{\mathbf{x}}_k\right\|},\\
\mathbf{b}_k &= \frac{\bar{\mathbf{x}}_k}{\left\|\bar{\mathbf{x}}_k\right\|},\\
c_k &= \mathbf{a}_k \cdot \mathbf{b}_k,\\
d_k &= \mathbf{x}_k \cdot \mathbf{b}_k,\\
\eta_k &= \frac{\exp(c_k h) - 1}{c_k},\\
\mathbf{z}_{k+1} &= \mathbf{x}_k + \eta_k d_k \mathbf{a}_k.
\end{aligned} \tag{17}$$

If $\mathbf{z}_{k+1}$ converges according to a given stopping criterion, such that

$$\left\|\mathbf{z}_{k+1} - \mathbf{x}_{k+1}\right\| < \varepsilon_2, \tag{18}$$

then go to (iii) to the next time step; otherwise, let $\mathbf{x}_{k+1} = \mathbf{z}_{k+1}$ and go to the computations in (17) again. In all the computations given below we will use $\theta = 1/2$.

4. Newton Iterative Scheme for DAEs

Now, we turn our attention to the DAEs defined in (1) and (2). Within a small time step we can suppose that the variables y_j, $j = 1, \ldots, m$ are constant in that interval of $t_k < t < t_{k+1}$. We give an initial guess of y_j, $j = 1, \ldots, m$ and insert them into (1). Then, we apply the above implicit scheme to find the next $\mathbf{x}_{k+1}$, supposing that $\mathbf{x}_k$ is already obtained in the previous time step. When $\mathbf{x}_{k+1}$ are available we insert them into (2) and then apply the Newton iterative scheme to solve $\mathbf{y}_{k+1}$ by

$$\mathbf{y}_{k+1}^{\ell+1} = \mathbf{y}_{k+1}^{\ell} - \mathbf{B}^{-1}\mathbf{F}\left(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}^{\ell}, t_{k+1}\right), \tag{19}$$

until the following convergence criterion is satisfied:

$$\left\|\mathbf{y}_{k+1}^{\ell+1} - \mathbf{y}_{k+1}^{\ell}\right\| < \varepsilon_1. \tag{20}$$

The component B_{ij} of the Jacobian matrix $\mathbf{B}$ is given by $\partial F_i / \partial y_j$. Below we use some examples to demonstrate the numerical processes in Sections 3 and 4.

5. Numerical Examples of DAEs

In order to assess the performance of the newly developed scheme based on the Lie-group $GL(n,\mathbb{R})$, let us investigate the following four examples of DAEs.

Example 1. Using the on-off switching criteria, we can synthesize the flow model of perfect plasticity into a two-phase system [12]:

$$\dot{\mathbf{Q}} = k_e\dot{\mathbf{q}} - \frac{k_e\lambda}{Q_0}\mathbf{Q}, \tag{21}$$

where λ is subjected to

$$\begin{aligned} \lambda &= \frac{1}{Q_0}\mathbf{Q}\cdot\dot{\mathbf{q}} > 0 \quad \text{if } \|\mathbf{Q}\| = Q_0,\ \mathbf{Q}\cdot\dot{\mathbf{q}} > 0, \\ \lambda &= 0 \quad \text{if } \|\mathbf{Q}\| < Q_0 \text{ or } \mathbf{Q}\cdot\dot{\mathbf{q}} \le 0. \end{aligned} \tag{22}$$

While $k_e > 0$ is known as an elastic modulus, the constant $Q_0 > 0$ is a yield strength of material.

We can view the above equations in the plastic state, that is, $\lambda > 0$, as a system of DAEs:

$$\dot{\mathbf{Q}} = k_e\dot{\mathbf{q}} - \frac{k_e\lambda}{Q_0}\mathbf{Q} =: \mathbf{f}, \tag{23}$$

$$\|\mathbf{Q}\|^2 = Q_0^2. \tag{24}$$

Now we explain that (23) and (24) are index two DAEs. Taking the time differential of (24) and inserting (23) into the resultant equation we can solve λ by

$$\lambda = \frac{1}{Q_0}\mathbf{Q}\cdot\dot{\mathbf{q}}. \tag{25}$$

Inserting it into (23) we obtain a nonlinear ODEs system:

$$\dot{\mathbf{Q}} = k_e\dot{\mathbf{q}} - \frac{k_e}{Q_0^2}\mathbf{Q}\cdot\dot{\mathbf{q}}\mathbf{Q}. \tag{26}$$

A further differential of (25) and inserting (26) leads to a differential equation for λ. Usually, when one applies the general purpose numerical integration method to solve (26), it cannot guarantee that the yield condition in (24) can be automatically satisfied. Hong and Liu [12] have developed the exponential-based scheme from the Lorentz group $SO_o(n,1)$, which can automatically satisfy (24).

We apply the implicit $GL(n,\mathbb{R})$ scheme to solve $\mathbf{Q}$ through (23) and then iteratively solve the unknown function λ through (24) by the Newton iterative method. The numerical processes of this implicit Lie-group DAE (LGDAE) are given below.

(i) Give an initial guess of λ_0, for example, $\lambda_0 = 0$.

(ii) Give an initial condition $\mathbf{Q}_0$ at an initial time $t = t_0$ and a time stepsize h.

(iii) For $k = 0, 1, \ldots$, we repeat the following computations to a specified terminal time $t = t_f$:

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k + h\mathbf{f}_k. \tag{27}$$

With the above $\mathbf{Q}_{k+1}$ generated from an Euler step as an initial guess, we then iteratively solve the new $\mathbf{Q}_{k+1}$ by

$$\begin{aligned} \overline{\mathbf{Q}}_k &= (1-\theta)\,\mathbf{Q}_k + \theta\mathbf{Q}_{k+1}, \\ \mathbf{a}_k &= \frac{\overline{\mathbf{f}}_k}{\left\|\overline{\mathbf{Q}}_k\right\|}, \\ \mathbf{b}_k &= \frac{\overline{\mathbf{Q}}_k}{\left\|\overline{\mathbf{Q}}_k\right\|}, \\ c_k &= \mathbf{a}_k\cdot\mathbf{b}_k, \\ d_k &= \mathbf{Q}_k\cdot\mathbf{b}_k, \\ \eta_k &= \frac{\exp(c_k h) - 1}{c_k}, \\ \mathbf{z}_{k+1} &= \mathbf{Q}_k + \eta_k d_k \mathbf{a}_k. \end{aligned} \tag{28}$$

If $\mathbf{z}_{k+1}$ converges according to a given stopping criterion, such that

$$\left\|\mathbf{z}_{k+1} - \mathbf{Q}_{k+1}\right\| < \varepsilon_2, \tag{29}$$

then go to (iv); otherwise, let $\mathbf{Q}_{k+1} = \mathbf{z}_{k+1}$ and go to (28).

(iv) For $j = 0, 1, \ldots$, we repeat the following computations:

$$\lambda_{j+1} = \lambda_j - \frac{F_j}{F_j'}, \tag{30}$$

where the prime denotes the differential with respect to λ, and

$$\begin{aligned} \mathbf{a}_k' &= \frac{\overline{\mathbf{f}}_k'}{\left\|\overline{\mathbf{Q}}_k\right\|}, \\ c_k' &= \mathbf{b}_k\cdot\mathbf{a}_k', \\ \eta_k' &= \frac{c_k'\left[(hc_k - 1)\exp(c_k h) + 1\right]}{c_k^2}, \\ \mathbf{Q}_{k+1}' &= \eta_k' d_k \mathbf{a}_k + \eta_k d_k \mathbf{a}_k', \\ F_j &= \left\|\mathbf{Q}_{k+1}\right\|^2 - Q_0^2, \\ F_j' &= 2\mathbf{Q}_{k+1}\cdot\mathbf{Q}_{k+1}'. \end{aligned} \tag{31}$$

If λ_j converges according to

$$\left|\lambda_{j+1} - \lambda_j\right| < \varepsilon_1, \tag{32}$$

then go to (iii) with λ_j as an initial guess of λ for the next time step; otherwise, let $\lambda_j = \lambda_{j+1}$ and go to (28).

In order to assess the performance of the above numerical method, we consider a strain control case with the strain

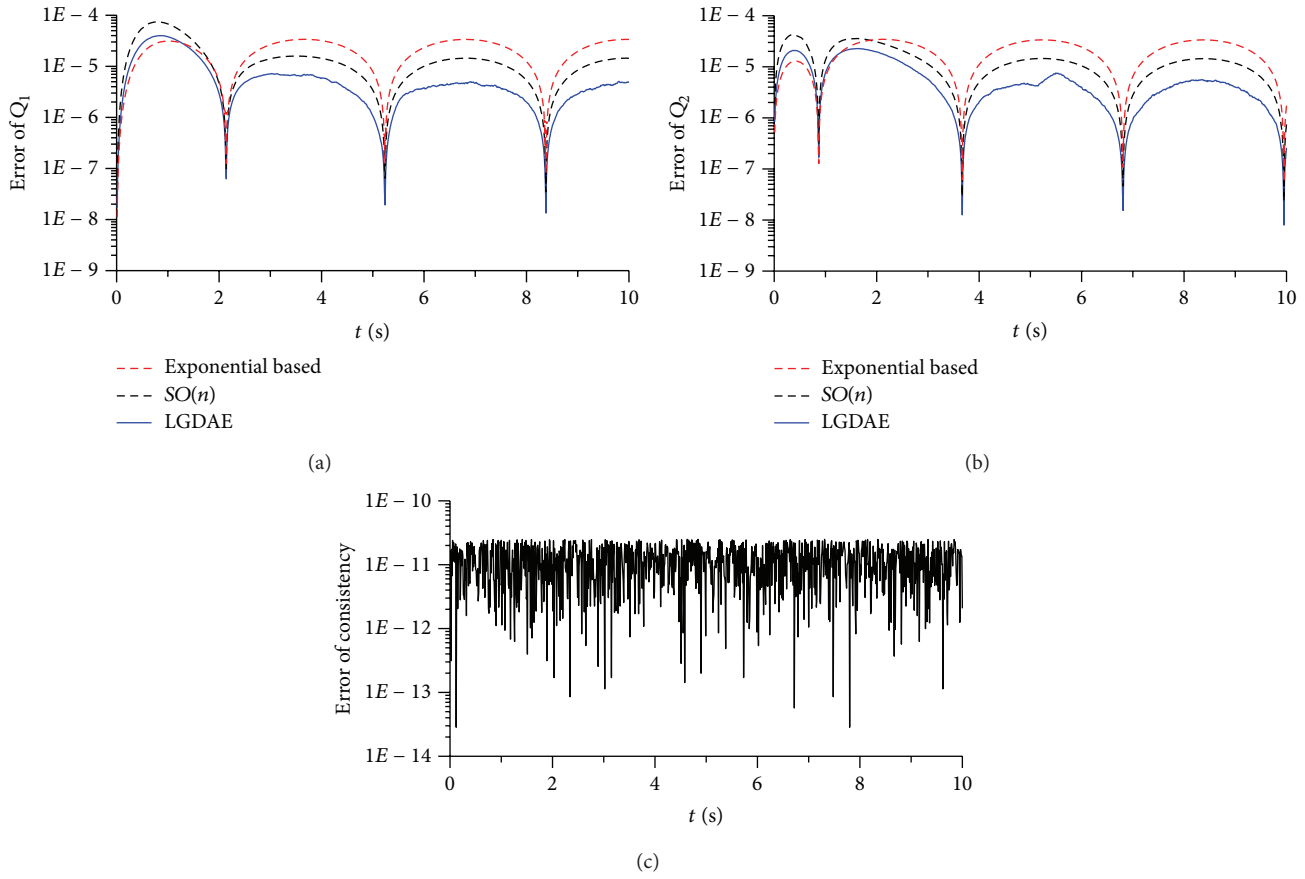

FIGURE 1: Example 1 of a plastic equation to compute the stresses, showing the numerical errors of (a) Q_1, (b) Q_2, and (c) consistency condition.

components being $q_1 = e_0 \cos(\omega t)$ and $q_2 = e_0 \sin(\omega t)$. Here we suppose that $\beta = k_e e_0 / Q_0 > 1$, and the initial stresses are on the yield surface with $Q_1 = Q_0 \cos\theta_0$ and $Q_2 = Q_0 \sin\theta_0$. As shown in Liu [13, 14], the responses of Q_1 and Q_2 have the following closed-form solutions:

$$\begin{aligned} \frac{Q_1}{Q_0} &= \frac{z + \beta\cos\theta_0 - 1}{\beta z}\cos(\omega t) - \frac{\dot{z}}{\beta\omega z}\sin(\omega t), \\ \frac{Q_2}{Q_0} &= \frac{\dot{z}}{\beta\omega z}\cos(\omega t) + \frac{z + \beta\cos\theta_0 - 1}{\beta z}\sin(\omega t), \end{aligned} \tag{33}$$

where

$$\begin{aligned} z(t) = 1 &+ \frac{\beta^2 - \beta\cos\theta_0}{m^2}\left[\cosh(m\omega t) - 1\right] \\ &+ \frac{\beta\sin\theta_0}{m}\sinh(m\omega t), \end{aligned} \tag{34}$$

in which $m = \sqrt{\beta^2 - 1}$.

In Figure 1, we plot the response errors of Q_1 and Q_2 and the error of the consistency condition, which is defined by $|\sqrt{Q_1^2 + Q_2^2} - Q_0|$, in a time range of $t \in [0, 10]$. We fix $k_e = 200{,}000$ MPa, $Q_0 = 200$ MPa, $e_0 = 0.002$, $\omega = 1$, and $\theta_0 = 0$, and the time stepsize used is $h = 0.001$. Under the convergence criteria $\varepsilon_2 = 10^{-8}$ for inner iterations and $\varepsilon_1 = 10^{-8}$ for outer iterations, we apply the LGDAE to solve the above problem. From Figure 1(c), we can observe that the LGDAE can retain the consistency condition very well. As shown in Figures 1(a) and 1(b), the accuracy of LGDAE is better than that obtained by the exponential-based scheme [12] and the $SO(n)$ scheme [15].

Example 2. We consider an index two nonlinear Hessenberg DAEs [7, 8]:

$$\begin{aligned} \dot{x}_1 &= t x_2^2 + \lambda + g_1(t), \\ \dot{x}_2 &= t\exp(x_1) + t\lambda + g_2(t), \\ x_1 &+ t x_2 + g_3(t) = 0, \end{aligned} \tag{35}$$

with $x_1(0) = x_2(0) = \lambda(0) = 0$, where

$$\begin{aligned} g_1(t) &= \frac{1 - t^2 - t^3}{(1+t)^2}, \\ g_2(t) &= \frac{1 - t - 4t^2 - 4t^3 - t^4}{(1+t)^2}, \\ g_3(t) &= -\ln(1+t) - \frac{t^2}{1+t}. \end{aligned} \tag{36}$$

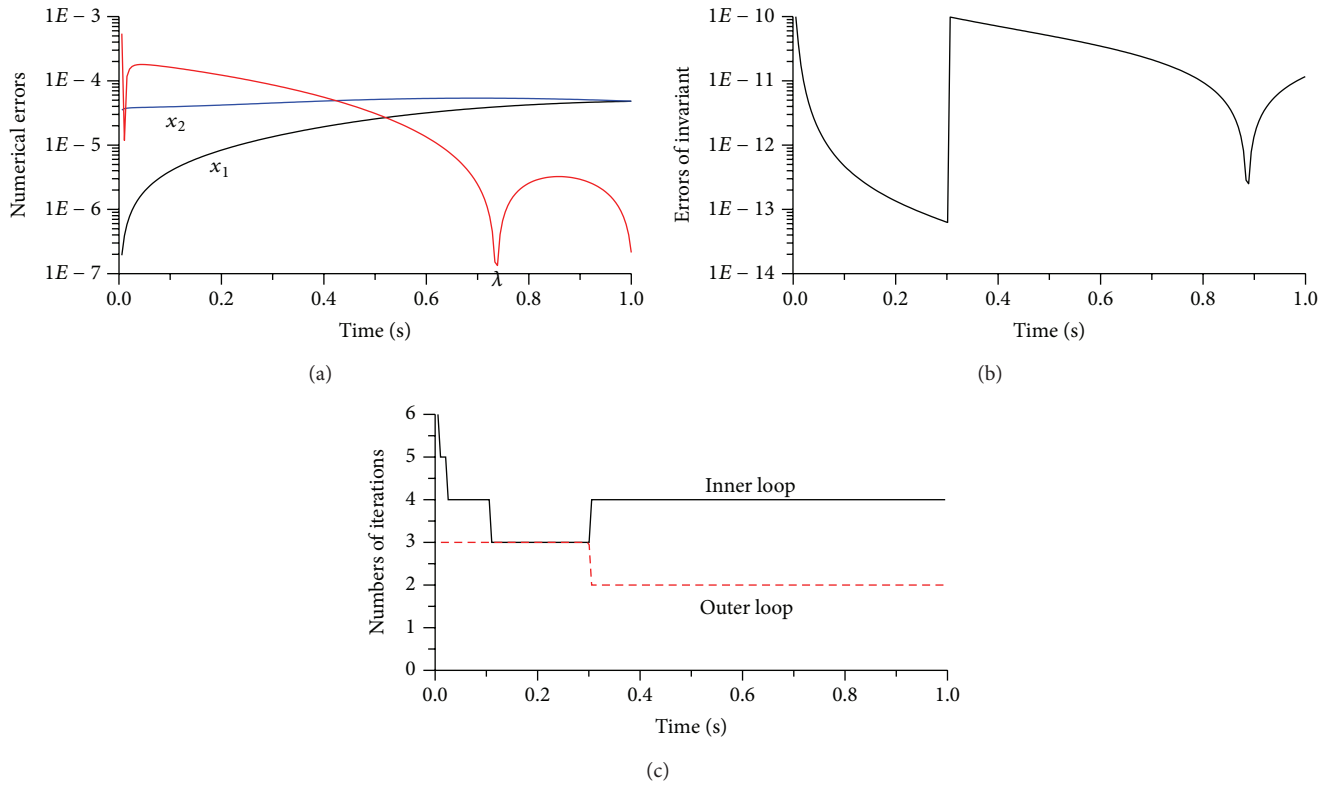

FIGURE 2: Example 2 showing (a) the numerical errors of solutions, (b) the error of invariant, and (c) the numbers of iterations.

The exact solutions of this problem are

$$x_1(t) = \ln(1+t), \qquad x_2(t) = \lambda(t) = \frac{t}{1+t}. \tag{37}$$

Under the convergence criteria $\varepsilon_2 = 10^{-15}$ for inner iterations and $\varepsilon_1 = 10^{-10}$ for outer iterations, we apply the LGDAE as that in (27)–(32) to solve this problem, where

$$\begin{aligned} F_j &= x_1^{k+1} + t_{k+1}x_2^{k+1} + g_3(t_{k+1}), \\ \left(a_k^1\right)' &= \frac{1}{\left\|\bar{\mathbf{x}}_k\right\|}, \\ \left(a_k^2\right)' &= \frac{\bar{t}_k}{\left\|\bar{\mathbf{x}}_k\right\|}, \\ \left(x_1^{k+1}\right)' &= a_k^1 d_k \eta_k' + \left(a_k^1\right)' d_k \eta_k, \\ \left(x_2^{k+1}\right)' &= a_k^2 d_k \eta_k' + \left(a_k^2\right)' d_k \eta_k, \\ F_j' &= \left(x_1^{k+1}\right)' + t_{k+1}\left(x_2^{k+1}\right)'. \end{aligned} \tag{38}$$

In the above, $\left(x_1^{k+1}\right)'$ and $\left(x_2^{k+1}\right)'$ denote, respectively, the differentials of x_1^{k+1} and x_2^{k+1} with respect to λ, and a_k^i denotes the ith component of $\mathbf{a}_k$.

We use $h = 10^{-3}$, and the problem is solved in a range of $t \le 1$. In Figure 2(a), we show the numerical errors of x_1, x_2 and λ, of which we can see that the numerical solutions are very accurate. In Figure 2(b), we show the error of $|x_1(t) + tx_2(t) + g_3(t)|$, which is almost zero with the order 10^{-11}. It can be seen that the numbers of iterations are few with three to six for inner iterations and two or three for outer iterations as shown in Figure 2(c).

Example 3. We consider an index three differential algebraic equations system given by Sand [4], which describes the position of a particle on a circular track:

$$\begin{aligned} \ddot{u}_1 &= 2u_2 + \lambda u_1, \\ \ddot{u}_2 &= -2u_1 + \lambda u_2, \\ u_1^2 + u_2^2 &= 1. \end{aligned} \tag{39}$$

For $(u_1(0), u_2(0)) = (0, 1)$, $\lambda(0) = 0$, the exact solution is $u_1(t) = \sin t^2$, $u_2(t) = \cos t^2$ and $\lambda(t) = -4t^2$. The above problem can be viewed as a mechanical control problem to select a suitable controller $\lambda(t)$ changing the system's stiffness such that the orbit of the mechanical system can really trace a circle in time.

We use this example to demonstrate how to transform the above DAEs to a full system of ODEs with the following strong form constraint:

$$I_1 = x_1^2 + x_3^2 - 1 = u_1^2 + u_2^2 - 1 = 0, \tag{40}$$

where we let $x_1 = u_1$, $x_2 = \dot{u}_1$, $x_3 = u_2$ and $x_4 = \dot{u}_2$. At the same time the above ODEs become

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= 2x_3 + \lambda x_1, \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= -2x_1 + \lambda x_3. \end{aligned} \tag{41}$$

Taking the time differential of (40), we can derive

$$x_1\dot{x}_1 + x_3\dot{x}_3 = x_1x_2 + x_3x_4 = 0, \tag{42}$$

due to $x_2 = \dot{x}_1$ and $x_4 = \dot{x}_3$. The above is a first differential form of the constraint in (40). However, it is still not available for the determination of the Lagrange multiplier λ. Thus, the second differential form of (42) is required:

$$x_1\dot{x}_2 + \dot{x}_1x_2 + x_3\dot{x}_4 + \dot{x}_3x_4 = 0. \tag{43}$$

Inserting (41) and using (40), we can derive

$$I_2 = \lambda + x_2^2 + x_4^2 = 0. \tag{44}$$

Then, we finally come to a system of ODEs:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= 2x_3 - x_1\left(x_2^2 + x_4^2\right), \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= -2x_1 - x_3\left(x_2^2 + x_4^2\right), \end{aligned} \tag{45}$$

and the Lagrange multiplier λ is calculated by

$$\lambda = -x_2^2 - x_4^2. \tag{46}$$

Finally, taking the differential of the above equation and inserting (45) for $\dot{x}_2$ and $\dot{x}_4$ we can obtain the differential equation for λ:

$$\dot{\lambda} = 2x_2\left[x_1\left(x_2^2 + x_4^2\right) - 2x_3\right] + 2x_4\left[2x_1 + x_3\left(x_2^2 + x_4^2\right)\right]. \tag{47}$$

In the above, we have transformed the DAEs in (39) into a set of ODEs system (45) and (47) through three times differentiations. This DAEs system is thus said to be of index three.

In general, we may hope the solution of the system (45) can automatically satisfy the following constraint on (x_1, x_3):

$$x_1^2 + x_3^2 = 1, \tag{48}$$

which is the last equation in (39). However, we should remind that there exists no general numerical integrator which can automatically satisfy (48). For this purpose, the numerical integrator must be particularly designed to meet that requirement. In the above, we have introduced two invariants I_1 and I_2; for this DAEs system the preservations of these two invariants are utmost important.

We apply the LGDAE to solve x_i, $i = 1, \ldots, 4$ by using (41) and then iteratively solve the weak form constraint in (42) by the Newton method to find λ. The processes are the same as that in (27)–(32), but now we have different F_j and F_j':

$$\begin{aligned} F_j &= x_1^{k+1}x_2^{k+1} + x_3^{k+1}x_4^{k+1}, \\ c_k' &= \frac{\overline{x}_1 b_k^2 + \overline{x}_3 b_k^4}{\left\|\overline{\mathbf{x}}_k\right\|}, \\ \eta_k' &= \frac{c_k'\left[(hc_k - 1)\exp(c_k h) + 1\right]}{c_k^2}, \\ \left(x_1^{k+1}\right)' &= a_k^1 d_k \eta_k', \\ \left(x_2^{k+1}\right)' &= a_k^2 d_k \eta_k' + d_k \eta_k \overline{x}_1, \\ \left(x_3^{k+1}\right)' &= a_k^3 d_k \eta_k', \\ \left(x_4^{k+1}\right)' &= a_k^4 d_k \eta_k' + d_k \eta_k \overline{x}_3, \\ F_j' &= x_1^{k+1}\left(x_2^{k+1}\right)' + \left(x_1^{k+1}\right)' x_2^{k+1} + \left(x_3^{k+1}\right)' x_4^{k+1} \\ &\quad + x_3^{k+1}\left(x_4^{k+1}\right)'. \end{aligned} \tag{49}$$

In the above, a_k^i and b_k^j are, respectively, the ith and jth components of the vectors $\mathbf{a}_k$ and $\mathbf{b}_k$.

Under the convergence criteria $\varepsilon_2 = 10^{-15}$ for inner iterations and $\varepsilon_1 = 10^{-6}$ for outer iterations, we apply the above numerical method with $h = 10^{-4}$ to solve (39). In Figure 3(a) we show the numerical errors of u_1, u_2, and λ, of which we can see that the numerical solutions are very accurate. In Figure 3(b), we show the errors of $|I_1|$ and $|I_2|$, which are almost zero with the order 10^{-10}. It can be seen that the accuracy of u_1 and u_2 is in the order of $h^{2.5}$, and the numbers of iterations are few with three for inner iterations and two for outer iterations as shown in Figure 3(c).

Example 4. Finally, we consider a more complex pendulum problem [16]:

$$\begin{aligned} \dot{x}_1 &= x_3 - \lambda_2 x_1, \\ \dot{x}_2 &= x_4 - \lambda_2 x_2, \\ \dot{x}_3 &= -\lambda_1 x_1, \\ \dot{x}_4 &= -\lambda_1 x_2 - 1, \\ I_1 &= x_1^2 + x_2^2 - 1 = 0, \\ I_2 &= x_1x_3 + x_2x_4 = 0. \end{aligned} \tag{50}$$

The exact solution of the above problem is not available, which has two Lagrange multipliers λ_1 and λ_2 and two constraints.

The initial values are $x_1(0) = 1$ and $x_2(0) = x_3(0) = x_4(0) = 0$. Under the convergence criteria $\varepsilon_2 = 10^{-15}$ for inner iterations and $\varepsilon_1 = 10^{-10}$ for outer iterations, we apply the LGDAE with $h = 10^{-4}$ to solve (50). In Figure 4(a), we show the numerical solutions. In Figure 4(b), we show the errors of $|I_1|$ and $|I_2|$, which are almost to be zero with the order 10^{-13} for $|I_1|$ and the order 10^{-17} for $|I_2|$. The numbers of iterations are few with three for inner iterations and two for outer iterations as shown in Figure 4(c).

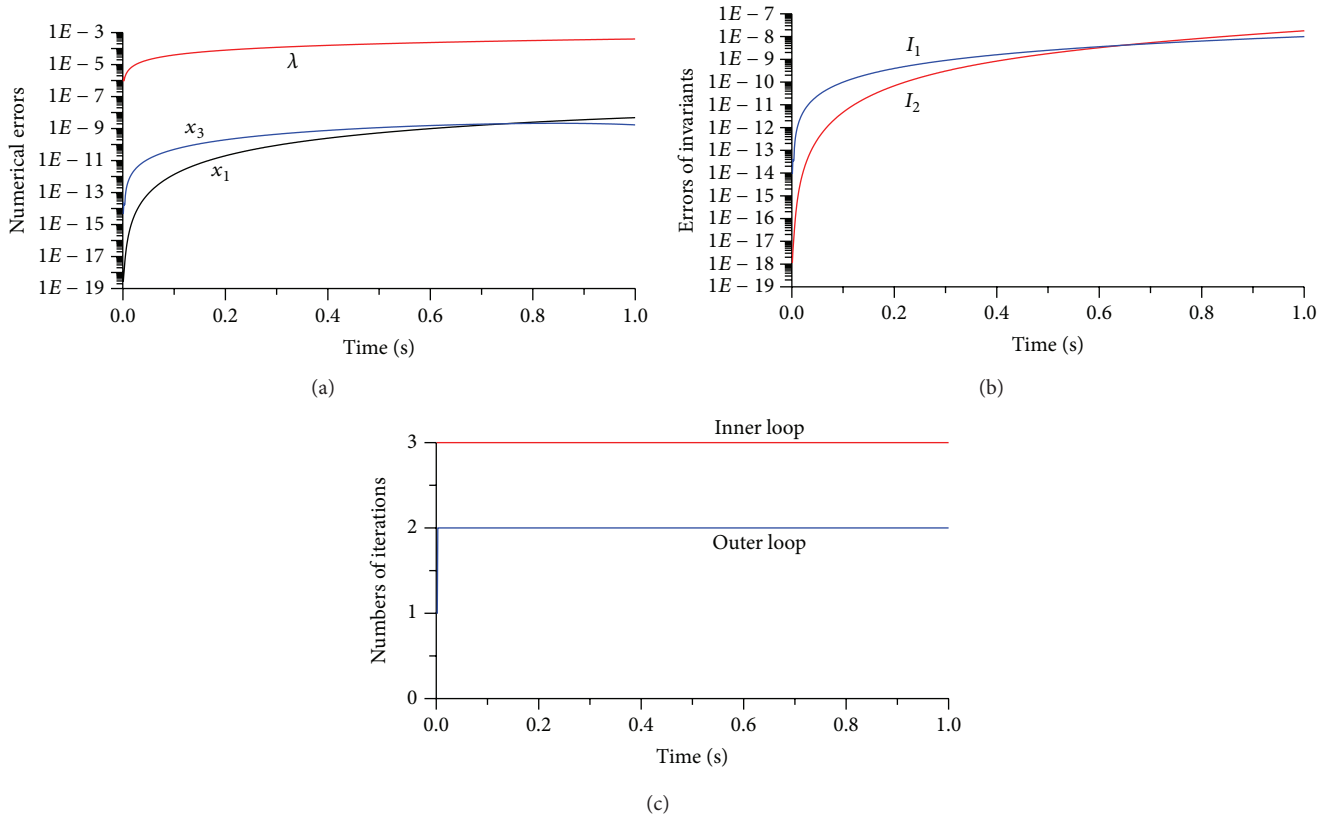

Figure 3: Example 3 showing (a) the numerical errors of solutions, (b) the errors of invariants, and (c) the numbers of iterations.

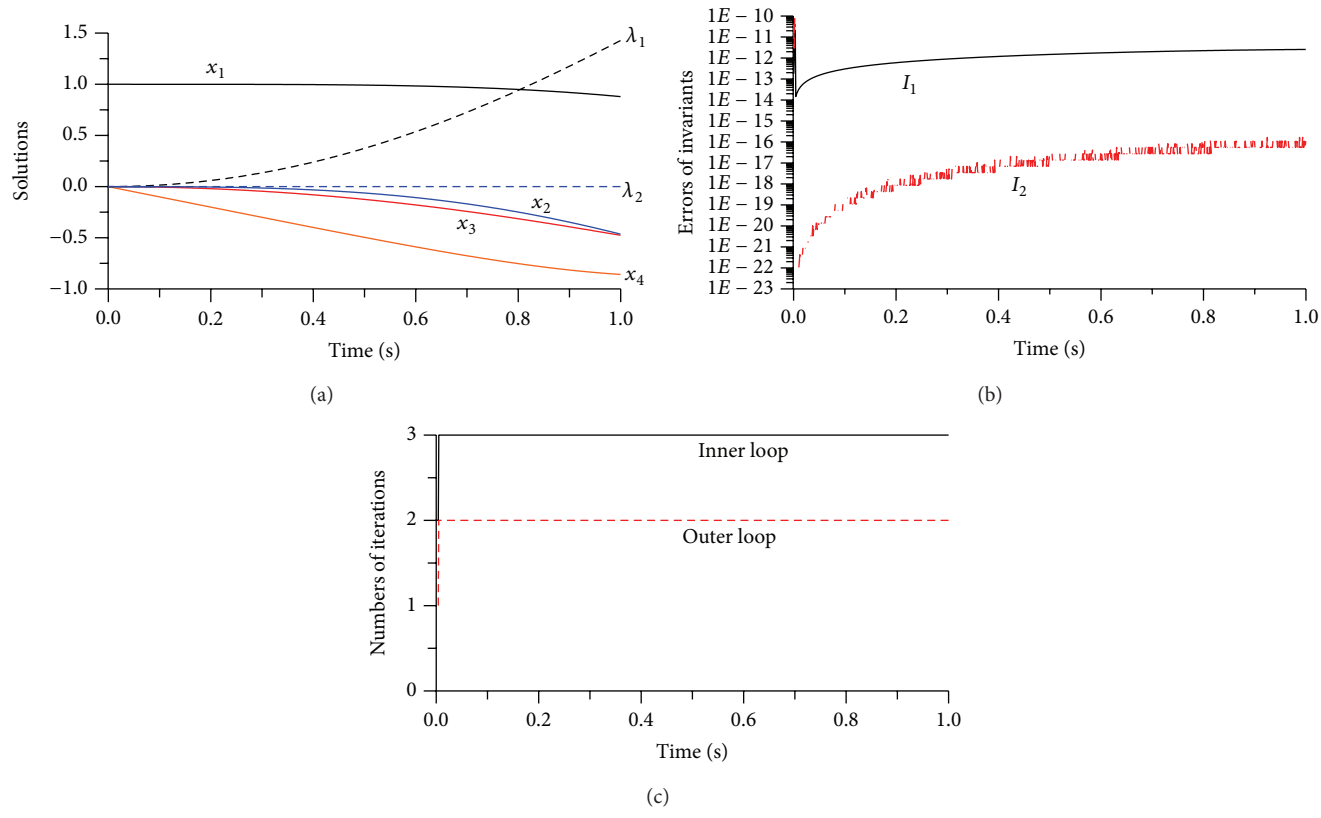

Figure 4: Example 4 showing (a) the numerical solutions, (b) the errors of invariants, and (c) the numbers of iterations.

6. Conclusions

By recasting the nonlinear ODEs: $\dot{\mathbf{x}} = \mathbf{f}$ into a quasi-linear Lie-form: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} = \mathbf{f}/\|\mathbf{x}\| \otimes \mathbf{x}/\|\mathbf{x}\|$, we have derived an implicit $GL(n, \mathbb{R})$ Lie-group algorithm together with the Newton iterative scheme, namely, the LGDAE, to solve the nonlinear differential algebraic equations. Four numerical examples were given to assess the performance of the novel method, which is easily implemented to solve the nonlinear differential algebraic equations with a high efficiency and accuracy.

Acknowledgments

The project NSC-102-2221-E-002-125-MY3 and the 2011 Outstanding Research Award from Taiwan's National Science Council and the 2011 Taiwan Research Front Award from Thomson Reuters, granted to the author, are highly appreciated. Also the author is a Lifetime Distinguished Professor of National Taiwan University, since 2013.

References

[1] C. Arévalo, S. L. Campbell, and M. Selva, "Unitary partitioning in general constraint preserving DAE integrators," *Mathematical and Computer Modelling*, vol. 40, no. 11-12, pp. 1273–1284, 2004.

[2] U. M. Ascher and L. R. Petzold, "Projected implicit Runge-Kutta methods for differential-algebraic equations," *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 1097–1120, 1991.

[3] R. P. K. Chan, P. Chartier, and A. Murua, "Post-projected Runge-Kutta methods for index-2 differential-algebraic equations," *Applied Numerical Mathematics*, vol. 42, no. 1-3, pp. 77–94, 2002.

[4] J. Sand, "On implicit Euler for high-order high-index DAEs," *Applied Numerical Mathematics*, vol. 42, no. 1–3, pp. 411–424, 2002.

[5] C.-S. Liu, "Preserving constraints of differential equations by numerical methods based on integrating factors," *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, no. 2, pp. 83–107, 2006.

[6] M. M. Hosseini, "Adomian decomposition method for solution of differential-algebraic equations," *Journal of Computational and Applied Mathematics*, vol. 197, no. 2, pp. 495–501, 2006.

[7] M. M. Hosseini, "Adomian decomposition method for solution of nonlinear differential algebraic equations," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1737–1744, 2006.

[8] F. Soltanian, S. M. Karbassi, and M. M. Hosseini, "Application of He's variational iteration method for solution of differential-algebraic equations," *Chaos, Solitons & Fractals*, vol. 41, no. 1, pp. 436–445, 2009.

[9] M. Saravi, E. Babolian, R. England, and M. Bromilow, "System of linear ordinary differential and differential-algebraic equations and pseudo-spectral method," *Computers & Mathematics with Applications*, vol. 59, no. 4, pp. 1524–1531, 2010.

[10] C.-S. Liu, "A method of Lie-symmetry $GL(n, \mathbb{R})$ for solving nonlinear dynamical systems," *International Journal of Non-Linear Mechanics*, vol. 52, pp. 85–95, 2013.

[11] C.-S. Liu, "A state feedback controller used to solve an ill-posed linear system by a $GL(n, \mathbb{R})$ iterative algorithm," *Communications in Numerical Analysis*, vol. 2013, Article ID cna-00181, 22 pages, 2013.

[12] H.-K. Hong and C.-S. Liu, "Internal symmetry in the constitutive model of perfect elastoplasticity," *International Journal of Non-Linear Mechanics*, vol. 35, no. 3, pp. 447–466, 2000.

[13] C.-S. Liu, "Intermittent transition to quasiperiodicity demonstrated via a circular differential equation," *International Journal of Non-Linear Mechanics*, vol. 35, no. 5, pp. 931–946, 2000.

[14] C.-S. Liu, "The steady-state behavior of the Prandtl-Reuss material bifurcated under nonproportional circular strain paths," *Journal of the Chinese Institute of Engineers*, vol. 26, no. 2, pp. 173–190, 2003.

[15] C.-S. Liu, "A Lie-group $DSO(n)$ method for nonlinear dynamical systems," *Applied Mathematics and Letters*, vol. 26, pp. 710–717, 2013.

[16] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, vol. 14 of *Classics in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1996.