Symbolic-Numeric Methods for Improving Structural Analysis of DAEs

Guangning Tan, Nedialko S. Nedialkov, and John D. Pryce

Abstract Systems of differential-algebraic equations (DAEs) are generated routinely by simulation and modeling environments, such as MapleSim and those based on the Modelica language. Before a simulation starts and a numerical method is applied, some kind of structural analysis is performed to determine which equations to be differentiated, and how many times. Both Pantelides's algorithm and Pryce's Σ-method are equivalent in the sense that, if one method succeeds in finding the correct index and producing a nonsingular Jacobian for a numerical solution procedure, then the other does also. Such a success occurs on many problems of interest, but these structural analysis methods can fail on simple solvable DAEs and give incorrect structural information including the index. This article investigates the Σ-method's failures, and presents two symbolic-numeric conversion methods for fixing these failures. These methods convert a DAE on which the Σ-method fails to a DAE on which this structural analysis may succeed.

1 Introduction

We consider DAEs of the general form

$$f_i(\,t, \text{the } x_j \text{ and derivatives of them}\,) = 0, \quad i = 1, \ldots, n, \tag{1}$$

G. Tan
School of Computational Science and Engineering, McMaster University,
1280 Main St. W., L8S 4L8, Hamilton, ON, Canada, e-mail: tang4@mcmaster.ca

N. S. Nedialkov
Department of Computing and Software, McMaster University,
1280 Main St. W., L8S 4L8, Hamilton, ON, Canada, e-mail: nedialk@mcmaster.ca

J. D. Pryce
School of Mathematics, Cardiff University,
Senghennydd Road, Cardiff CF24 4AG, Wales, UK., e-mail: prycejd1@cardiff.ac.uk

where the $x_j(t),\ j=1,\ldots,n$ are state variables that are functions of an independent variable t, usually regarded as time.

Pryce's structural analysis (SA), the Σ-method [9], determines for a DAE (1) its structural index, the number of degrees of freedom (DOF), variables and derivatives that need initial values, and constraints of this DAE. These SA results can help decide how to apply an index reduction algorithm [4], perform a regularization process [12], or design a solution scheme for a Taylor series method [1, 2, 7]. The Σ-method is equivalent to Pantelides's algorithm [8]: they both produce the same structural index [9, Theorem 5.8]. This index is an upper bound for the differentiation index, and often they are the same [9].

The Σ-method provably succeeds on many problems of practical interest, producing a nonsingular System Jacobian. However, this SA method can fail—hence Pantelides's algorithm can fail as well—on some simple solvable DAEs, producing an identically singular System Jacobian.

We investigate this SA's failures and present two symbolic-numeric conversion methods for fixing these failures. After each conversion, the value of the signature matrix is guaranteed to decrease, provided some conditions are satisfied. We conjecture that such a decrease should result in a better problem formulation of a DAE, so that the SA may succeed and hence produce a nonsingular System Jacobian.

Sect. 2 summarizes the Σ-method. Sect. 3 describes this SA's failure. Sect. 4 presents our two symbolic-numeric conversion methods, each of which is illustrated with an example therein. Sect. 5 gives conclusions and indicates the future work.

2 Summary of the Σ-method

This SA method [9] constructs for a DAE (1) an $n\times n$ *signature matrix* Σ, whose (i,j) entry σ_{ij} is either an integer ≥ 0, order of the highest derivative to which variable x_j occurs in equation f_i, or $-\infty$ if x_j does not occur in f_i.

A *highest-value transversal* (HVT) of Σ is a set T of n positions (i,j) with one entry in each row and each column, such that the sum of these entries is the largest possible. This sum is the *value of* Σ, written $\mathrm{Val}(\Sigma)$. Assume henceforth that $\mathrm{Val}(\Sigma)$ is finite, or equivalently, the DAE is *structurally well posed*. Then we find $2n$ *equation and variable offsets* $c_1,\ldots,c_n$ and $d_1,\ldots,d_n$, respectively, which are integers satisfying

$$c_i \geq 0 \quad \text{for all } i; \qquad d_j - c_i \geq \sigma_{ij} \quad \text{for all } i,j \text{ with equality on a HVT}\,. \tag{2}$$

Offsets satisfying (2) are called *valid*, but they are never unique; there exists an elementwise smallest solution of (2) termed the *canonical offsets*.

We associate with some valid offsets an $n\times n$ *System Jacobian* $\mathbf{J}$, defined as

$$\mathbf{J}_{ij} = \begin{cases} \partial f_i / \partial x_j^{(\sigma_{ij})} & \text{if } d_j - c_i = \sigma_{ij}, \text{ and} \\ 0 & \text{otherwise}\,. \end{cases} \tag{3}$$

If one $\mathbf{J}$ is nonsingular—and hence all $\mathbf{J}$'s are—at a *consistent point* as defined in [5], then we say that the Σ-method *succeeds* and there exists (locally) a unique solution *through* this point [9].

In the success case, the SA uses the canonical offsets to determine the *structural index* and the number of *DOF*:

$$\nu_S = \max_i c_i + \begin{cases} 1 & \text{if some } d_j = 0 \text{, and} \\ 0 & \text{otherwise ,} \end{cases}$$

$$\mathrm{DOF} = \mathrm{Val}(\Sigma) = \sum_{(i,j)\in T} \sigma_{ij} = \sum_j d_j - \sum_i c_i \ .$$

Example 1. We show[1] the above concepts for the simple pendulum (PEND), a DAE of differentiation index 3.

$$\begin{aligned} 0 &= f_1 = x'' + x\lambda \\ 0 &= f_2 = y'' + y\lambda - g \\ 0 &= f_3 = x^2 + y^2 - L^2 \end{aligned} \qquad \Sigma = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \begin{bmatrix} 2^\bullet & & 0^* \\ & 2^* & 0^\bullet \\ 0^* & 0^\bullet & \end{bmatrix} \\ \begin{array}{ccc} 2 & 2 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 2 \\ \\ \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \begin{bmatrix} 1 & & x \\ & 1 & y \\ 2x & 2y & \end{bmatrix} \end{array}$$

The state variables are x, y, and λ; G is gravity and $L > 0$ is the length of the pendulum. There are two HVTs of Σ, marked with $\bullet$ and $*$, respectively. A blank in Σ denotes $-\infty$, and a blank in $\mathbf{J}$ denotes 0. Since $\det(\mathbf{J}) = -2(x^2+y^2) = -2L^2 \neq 0$, the System Jacobian is nonsingular, and the SA succeeds. The structural index is hence $\nu_S = \min_i c_i + 1 = 2 + 1 = 3$ (because $d_3 = 0$), which equals the differentiation index. The number of DOF is $\mathrm{Val}(\Sigma) = \sum_j d_j - \sum_i c_i = 4 - 2 = 2$.

3 Structural Analysis's Failure

We say that the Σ-method *fails*, if a DAE (1) has a finite $\mathrm{Val}(\Sigma)$ and an identically singular System Jacobian $\mathbf{J}$. In the failure case, the SA reports $\mathrm{Val}(\Sigma)$ as an "apparent" DOF but not a meaningful one.

In this article, we only focus on the case where such an identically singular $\mathbf{J}$ is not structurally singular—that is, there exists a HVT T of Σ such that $\mathbf{J}_{ij}$ is generically nonzero for all $(i, j) \in T$. For a detailed description and discussion of SA's failure, we refer the reader to [13, §4].

Example 2. We illustrate a failure case with the following DAE[2] in [3, p. 23].

---

[1] When we give a DAE for example, we present with it its signature matrix Σ, offsets c_i and d_j, and the associated System Jacobian $\mathbf{J}$.

[2] In the original formulation, the driving functions are f_1, f_2. We change them to g_1, g_2.

$$
\begin{aligned}
0 = f_1 &= x' + ty' - g_1(t) \\
0 = f_2 &= x \;\; + ty \;\; - g_2(t)
\end{aligned}
\qquad
\Sigma = \begin{array}{c|cc|c} & x & y & c_i \\ \hline f_1 & 1^\bullet & 1 & 0 \\ f_2 & 0 & 0^\bullet & 1 \\ \hline d_j & 1 & 1 & \end{array}
\qquad
\mathbf{J} = \begin{array}{c|cc|} & x & y \\ \hline f_1 & 1 & t \\ f_2 & 1 & t \\ \hline \end{array}
$$

The SA fails since $\det(\mathbf{J}) = 0$. Now $\mathbf{J}$ is identically singular but not structurally singular: on a HVT marked by $\bullet$, each of $\mathbf{J}_{11} = 1$ and $\mathbf{J}_{22} = t$ is generically nonzero.

One simple fix is to replace f_1 by

$$0 = \overline{f}_1 = -f_1 + f_2' = y + g_1(t) - g_2'(t) \; ,$$

which results in an algebraic problem; cf. [4, Example 5].

$$
\begin{aligned}
0 = \overline{f}_1 &= y + g_1(t) - g_2'(t) \\
0 = f_2 &= x + ty - g_2(t)
\end{aligned}
\qquad
\overline{\Sigma} = \begin{array}{c|cc|c} & x & y & c_i \\ \hline \overline{f}_1 & & 0^\bullet & 0 \\ f_2 & 0^\bullet & 0 & 0 \\ \hline d_j & 0 & 0 & \end{array}
\qquad
\overline{\mathbf{J}} = \begin{array}{c|cc|} & x & y \\ \hline \overline{f}_1 & & 1 \\ f_2 & 1 & t \\ \hline \end{array}
$$

Now that $\det(\overline{\mathbf{J}}) = -1 \neq 0$, the SA succeeds. Notice $\mathrm{Val}(\overline{\Sigma}) = 0 < 1 = \mathrm{Val}(\Sigma)$.

Another simple fix is to introduce a new variable $z = x + ty$ and eliminate x in f_1 and f_2; note $z' = x' + ty' + y$.

$$
\begin{aligned}
0 = \overline{f}_1 &= -y + z' - g_1(t) \\
0 = \overline{f}_2 &= \qquad\;\; z \;\, - g_2(t)
\end{aligned}
\qquad
\overline{\Sigma} = \begin{array}{c|cc|c} & y & z & c_i \\ \hline \overline{f}_1 & 0^\bullet & 1 & 0 \\ \overline{f}_2 & & 0^\bullet & 1 \\ \hline d_j & 0 & 1 & \end{array}
\qquad
\overline{\mathbf{J}} = \begin{array}{c|cc|} & y & z \\ \hline \overline{f}_1 & -1 & 1 \\ \overline{f}_2 & & 1 \\ \hline \end{array}
$$

For this resulting DAE, $\det(\overline{\mathbf{J}}) = -1 \neq 0$, and the SA succeeds. After solving for y and z, we can obtain $x = z - ty$. This fix gives $\mathrm{Val}(\overline{\Sigma}) = 0 < 1 = \mathrm{Val}(\Sigma)$ also.

We shall note that each of the above fixes actually uses each of the methods described below.

4 Conversion Methods

We present two conversion methods for systematically fixing SA's failures. The first method is based on replacing an existing equation by a linear combination of some equations and derivatives of them. We call this method the linear combination (LC) method and describe it in Sect. 4.1. The second method is based on substituting newly introduced variables for some expressions and enlarging the system. We call this method the expression substitution (ES) method and describe it in Sect. 4.2.

We only present the main features of these methods; the reader is referred to [13] for more details and especially the proofs of Theorems 1 and 2 below.

Given a DAE (1), we assume henceforth that $\mathrm{Val}(\Sigma)$ is finite and that a System Jacobian $\mathbf{J}$ is identically singular but not structurally singular.

4.1 Linear Combination Method

Let u be a nonzero n-vector function in the cokernel of $\mathbf{J}$, that is, $u \in \text{coker}(\mathbf{J})$ and $\mathbf{J}^T u = \mathbf{0}$. Let u_i denote the ith component of u. Here, we consider $\mathbf{J}$ and u as functions of t, the x_j's and derivatives of them. For such a function ω, $\omega \neq 0$ means that ω is nonzero for all t in some time interval $\mathbb{I}$.

For our convenience, we let

$$\sigma(x_j, u) = \begin{cases} \text{order of the highest derivative to which } x_j \text{ occurs in } u\text{; or} \\ -\infty \ \text{ if } x_j \text{ does not occur in } u \ . \end{cases}$$

Denote

$$I = \{\, i \mid u_i \neq 0 \,\}, \quad \underline{c} = \min_{i \in I} c_i, \quad \text{and} \quad L = \left\{\, i \in I \mid c_i = \underline{c} \,\right\} . \tag{4}$$

The LC method is based on the following theorem.

Theorem 1. *If*

$$\sigma(x_j, u) < d_j - \underline{c} \qquad \textit{for all } j = 1, \ldots, n \, , \tag{5}$$

and we replace some f_l with $l \in L$ by

$$\overline{f} = \sum_{i \in I} u_i f_i^{(c_i - \underline{c})} \tag{6}$$

(denoted as $\overline{f}_l$), then $Val(\overline{\Sigma}) < Val(\Sigma)$, where $\overline{\Sigma}$ is the signature matrix of the resulting DAE.

We call (5) the condition for applying the LC method. The following example illustrates this method.

Example 3. Consider

$$\begin{aligned} 0 &= f_1 = -x_1' + x_3 & \qquad 0 &= f_3 = x_1 x_2 + g_1(t) \\ 0 &= f_2 = -x_2' + x_4 & \qquad 0 &= f_4 = x_1 x_4 + x_2 x_3 + x_1 + x_2 + g_2(t) \, , \end{aligned}$$

where g_1 and g_2 are given driving functions.

$$\Sigma = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \left[\begin{array}{cccc} 1^\bullet & & 0 & \\ & 1 & & 0^\bullet \\ 0 & 0^\bullet & & \\ \boxed{0} & \boxed{0} & 0^\bullet & 0 \end{array}\right] \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 0 \\ \\ \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \left[\begin{array}{cccc} -1 & & 1 & \\ & -1 & & 1 \\ x_2 & x_1 & & \\ & & x_2 & x_1 \end{array}\right] \end{array}$$

A shaded entry $\boxed{\sigma_{ij}}$ in Σ denotes a position (i, j) where $d_j - c_i > \sigma_{ij} \geq 0$ and hence $\mathbf{J}_{ij} = 0$ by (3). The SA fails since $\det(\mathbf{J}) = 0$, where $\mathbf{J}$ is identically (but not structurally) singular.

We use $u = \big(x_2, x_1, 1, -1\big)^T \in \operatorname{coker}(\mathbf{J})$. Then (4) becomes

$$I = \big\{\, i \mid u_i \neq 0 \,\big\} = \big\{\, 1,2,3,4 \big\}, \quad \underline{c} = \min_{i\in I} c_i = 0\,,$$
$$\text{and} \quad L = \big\{\, i \in I \mid c_i = \underline{c} = 0 \big\} = \big\{\, 1,2,4 \,\big\}\,.$$

The condition (5) holds since

$$\begin{aligned}
\sigma(x_1,u) &= \;\;0 \;\; < 1-0 = d_1 - \underline{c}, \quad & \sigma(x_2,u) &= \;\;0 \;\; < 1-0 = d_2 - \underline{c}\,,\\
\sigma(x_3,u) &= -\infty < 0-0 = d_3 - \underline{c}, \quad & \sigma(x_4,u) &= -\infty < 0-0 = d_4 - \underline{c}\,.
\end{aligned}$$

Choosing $l = 4 \in L = \big\{\, 1,2,4 \big\}$ for example, we replace f_4 by

$$\overline{f}_4 = \sum_{i\in I} u_i f_i^{(c_i-\underline{c})} = x_2 f_1 + x_1 f_2 + f_3' - f_4 = -x_1 - x_2 + g_1'(t) - g_2(t)\,.$$

The resulting DAE is $(f_1, f_2, f_3, \overline{f}_4) = \mathbf{0}$.

$$\overline{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ \overline{f}_4 \\ d_j \end{array}
\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1 & & 0^\bullet & \\ & 1 & & 0^\bullet \\ 0 & 0^\bullet & & \\ 0^\bullet & 0 & & \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array}
\begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 1 \\ \\ \end{array}
\qquad
\overline{\mathbf{J}} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ \overline{f}_4 \end{array}
\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ & -1 & & 1 \\ x_2 & x_1 & & \\ -1 & -1 & & \end{bmatrix} \end{array}$$

Now $\operatorname{Val}(\overline{\Sigma}) = 0 < 1 = \operatorname{Val}(\Sigma)$. The SA succeeds at all points where $\det(\overline{\mathbf{J}}) = x_2 - x_1 \neq 0$.

From (4) and (6), we can recover the replaced equation f_l by

$$f_l = \Big(\overline{f}_l - \textstyle\sum_{i\in I\setminus\{l\}} u_i f_i^{(c_i-\underline{c})}\Big) \big/ u_l\,.$$

Provided $u_l \neq 0$, it is not difficult to show that the original DAE and the resulting one have the same solution (if there exists one); see also [13, §5.3].

If the resulting DAE still has an identically (but not structurally) singular System Jacobian, we can iterate the LC method, provided the condition (5) is satisfied. Since each conversion reduces the value of the signature matrix by at least one, the number of iterations does not exceed the original $\operatorname{Val}(\Sigma)$.

4.2 Expression Substitution Method

Let v be a nonzero n-vector function in the kernel of $\mathbf{J}$, that is, $v \in \ker(\mathbf{J})$ and $\mathbf{J}v = \mathbf{0}$. Denote

$$\begin{aligned} &J = \left\{\, j \mid v_j \neq 0 \,\right\}, \quad s = |J| \,, \\ &M = \left\{\, i \mid d_j - c_i = \sigma_{ij} \text{ for some } j \in J \,\right\}, \quad \text{and} \quad \overline{c} = \max_{i \in M} c_i \,. \end{aligned} \tag{7}$$

We choose an $l \in J$, and introduce $s-1$ new variables

$$y_j = x_j^{(d_j - \overline{c})} - \frac{v_j}{v_l} x_l^{(d_l - \overline{c})} \qquad \text{for all } j \in J \setminus \left\{\, l \,\right\} . \tag{8}$$

In each f_i with $i \in M$, we

$$\begin{aligned} &\text{substitute} \quad \left(y_j + \frac{v_j}{v_l} x_l^{(d_l - \overline{c})} \right)^{(\overline{c} - c_i)} \\ &\text{for every} \quad x_j^{(\sigma_{ij})} \quad \text{with } \sigma_{ij} = d_j - c_i \text{ and } j \in J \setminus \left\{\, l \,\right\} . \end{aligned} \tag{9}$$

Denote by $\overline{f}_i$ the equations that result from these substitutions, and write $\overline{f}_i = f_i$ for $i \notin M$. By (8), we append to these $\overline{f}_i$'s the equations

$$0 = g_j = -y_j + x_j^{(d_j - \overline{c})} - \frac{v_j}{v_l} x_l^{(d_l - \overline{c})} \quad \text{for all } j \in J \setminus \left\{\, l \,\right\} \tag{10}$$

that prescribe the substitutions. Hence the resulting enlarged system consists of

$$\begin{array}{llcll} \text{equations} & \left(\overline{f}_1, \ldots, \overline{f}_n \right) = \mathbf{0} & \text{and} & g_j = 0 & \text{for all } j \in J \setminus \left\{\, l \,\right\} \\ \text{in variables} & x_1, \ldots, x_n & \text{and} & y_j & \text{for all } j \in J \setminus \left\{\, l \,\right\} . \end{array}$$

Critical to the ES method is the following theorem.

Theorem 2. *Let J, s, M, and $\overline{c}$ be as defined in (7). Assume*

$$\sigma\left(x_j, v\right) \leq \begin{cases} d_j - \overline{c} - 1 & \textit{if } j \in J \\ d_j - \overline{c} & \textit{otherwise} \,, \end{cases} \qquad \textit{and} \quad d_j - \overline{c} \geq 0 \quad \textit{for all } j \in J \,. \tag{11}$$

If we

- *introduce $s-1$ new variables defined in (8),*
- *perform substitutions in f_i for all $i \in M$ by (9), and*
- *append the equations g_j in (10),*

then $Val(\overline{\Sigma}) < Val(\Sigma)$, where $\overline{\Sigma}$ is the signature matrix of the resulting DAE.

We call the (11) the conditions for applying the ES method.

Example 4. We illustrate the ES method with the artificially constructed DAE below.

$$\begin{aligned} 0 &= f_1 = x_1 + e^{-x_1' - x_2 x_2''} + h_1(t) \\ 0 &= f_2 = x_1 + x_2 x_2' + x_2^2 + h_2(t) \end{aligned} \qquad \Sigma = \begin{array}{c|cc|c} & x_1 & x_2 & c_i \\ \hline f_1 & 1^\bullet & 2 & 0 \\ f_2 & 0 & 1^\bullet & 1 \\ \hline d_j & 1 & 2 & \end{array} \qquad \mathbf{J} = \begin{array}{c|cc} & x_1 & x_2 \\ \hline f_1 & -\alpha & -\alpha x_2 \\ f_2 & 1 & x_2 \end{array}$$

Here h_1 and h_2 are given driving functions, and $\alpha = e^{-x_1'-x_2x_2''}$. Obviously $\det(\mathbf{J}) = 0$ and the SA fails.

Suppose we choose $v = (x_2, -1)^T \in \ker(\mathbf{J})$. Then (7) becomes

$$J = \{1, 2\}, \quad s = |J| = 2, \quad M = \{1, 2\}, \quad \text{and} \quad \overline{c} = \max_{i\in M} c_i = 1 .$$

We can apply the ES method as conditions (11) read

$$\begin{aligned}
\sigma(x_1, v) &= -\infty \le -1 = 1-1-1 = d_1 - \overline{c} - 1, \quad d_1 - \overline{c} = 1-1 = 0 \ge 0 ,\\
\sigma(x_2, v) &= \ \ 0 \ \ \le \ \ 0 = 2-1-1 = d_2 - \overline{c} - 1, \quad d_2 - \overline{c} = 2-1 = 1 \ge 0 .
\end{aligned}$$

For example, we choose $l = 2 \in J$. Now $J \setminus \{l\} = \{1\}$. Using (8) and (10), we introduce a new variable

$$y_1 = x_1^{(d_1-\overline{c})} - \frac{v_1}{v_2} x_2^{(d_2-\overline{c})} = x_1^{(1-1)} - \frac{x_2}{-1} x_2^{(2-1)} = x_1 + x_2 x_2' ,$$

and append an equation $0 = g_1 = -y_1 + x_1 + x_2x_2'$. Then we substitute $(y_1 - x_2x_2')'$ for x_1' in f_1 to obtain $\overline{f}_1$, and substitute $y_1 - x_2x_2'$ for x_1 in f_2 to obtain $\overline{f}_2$. The resulting DAE and its SA results are shown below.

$$\begin{aligned}
0 = \overline{f}_1 &= x_1 + e^{-y_1'+x_2'^2} + h_1(t)\\
0 = \overline{f}_2 &= y_1 + x_2^2 + h_2(t)\\
0 = g_1 &= -y_1 + x_1 + x_2x_2'
\end{aligned}
\qquad
\overline{\Sigma} = \begin{array}{c|ccc|c}
 & x_1 & x_2 & y_1 & c_i \\
\hline
\overline{f}_1 & 0 & 1 & 1^\bullet & 0\\
\overline{f}_2 & & 0^\bullet & 0 & 1\\
g_1 & 0^\bullet & 1 & \boxed{0} & 0\\
\hline
d_j & 0 & 1 & 1 &
\end{array}
\qquad
\overline{\mathbf{J}} = \begin{array}{c|ccc|}
 & x_1 & x_2 & y_1 \\
\hline
\overline{f}_1 & 1 & 2x_2'\beta & -\beta\\
\overline{f}_2 & & 2x_2 & 1\\
g_1 & 1 & x_2 &
\end{array}$$

Here $\beta = e^{-y_1'+x_2'^2}$. Now $\mathrm{Val}(\overline{\Sigma}) = 1 < 2 = \mathrm{Val}(\Sigma)$. The SA succeeds at all points where $\det(\overline{\mathbf{J}}) = 2\beta(x_2 + x_2') - x_2 \ne 0$.

From the steps of applying the ES method, we can undo the expression substitutions to recover the original DAE. Similar to the LC method, the ES method also guarantees that, provided $v_l \ne 0$, the original DAE and the resulting one have the same solution (if there is one); this is shown in [13, §6.3].

From our experience, we usually attempt the LC method first, and when its condition (5) is violated, we try the ES method. Also, it is desirable to choose an $l \in L$ [resp. $l \in J$] in the LC [resp. ES] method, such that an u_l [resp. v_l] *never* becomes zero. For example, it can be a nonzero constant, $x_1^2 + 1$, or $2 + \cos x_2$. Such a choice of l guarantees that the resulting DAE is "equivalent" to the original one—that is, they *always* have the same solution, if there exists one.

We show below that the LC method does not work on the DAE in Example 4 because the condition (5) is not satisfied.

Choose $u = (1, \alpha)^T \in \mathrm{coker}(\mathbf{J})$, where $\alpha = e^{-x_1'-x_2x_2''}$. Then (4) becomes

$$I = \{\, i \mid u_i \ne 0 \,\} = \{1, 2\}, \quad \underline{c} = \min_{i\in I} c_i = 0, \ \text{and} \ L = \{\, i \in I \mid c_i = \underline{c} \,\} = \{1\} .$$

Since x_1 and x_2 occur of order 1 and 2 in u, respectively, we have

$$\sigma(x_1,u) = 1 \not< 1-0 = d_1 - \underline{c} \quad \text{and} \quad \sigma(x_2,u) = 2 \not< 2-0 = d_2 - \underline{c}\,.$$

Hence (5) is not satified. Choosing $l = 1 \in I$ for example and replacing f_1 by

$$\overline{f}_1 = u_1 f_1 + u_2 f_2' = x_1 + h_1(t) + \alpha\left(1 + x_1' + x_2 x_2'' + (x_2')^2 + 2x_2x_2' + h_2'(t)\right)$$

results in a DAE $\left(\overline{f}_1, f_2\right) = \mathbf{0}$.

$$\overline{\Sigma} = \begin{array}{c c c} & \begin{array}{cc} x_1 & x_2 \end{array} & c_i \\ \begin{array}{c} \overline{f}_1 \\ f_2 \end{array} & \left[\begin{array}{cc} 1^\bullet & 2 \\ 0 & 1^\bullet \end{array}\right] & \begin{array}{c} 0 \\ 1 \end{array} \\ d_j & \begin{array}{cc} 1 & 2 \end{array} & \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c c} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{array}{c} \overline{f}_1 \\ f_2 \end{array} & \left[\begin{array}{cc} -\gamma\alpha & -\gamma\alpha x_2 \\ 1 & x_2 \end{array}\right] \end{array}$$

Here $\gamma = x_1' + x_2 x_2'' + (x_2')^2 + 2x_2x_2' + h_2'(t)$. The SA fails still, since $\overline{\mathbf{J}}$ is identically (but not structurally) singular. Now $\mathrm{Val}(\overline{\Sigma}) = \mathrm{Val}(\Sigma) = 2$.

5 Conclusions and Future Work

We proposed two symbolic-numeric conversion methods for improving the Σ-method. They convert a DAE of finite $\mathrm{Val}(\Sigma)$ and an identically (but not structurally) singular System Jacobian to a DAE that may have a nonsingular System Jacobian. A conversion guarantees that both DAEs have (at least locally) the same solution if there exists one. The conditions for applying these methods can be checked automatically, and the main result of a conversion is $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$, where $\overline{\Sigma}$ is the signature matrix of the resulting DAE.

In our experiments, the linear combination (LC) method and the expression substitution (ES) method succeed in fixing many solvable DAEs on which the SA fails. We believe that these methods can help make the Σ-method more reliable. We also conjecture that reducing the value of the signature matrix tends to give a better formulation of a DAE in the SA's perspective and then a nonsingular System Jacobian.

An implementation of the conversion methods requires (a) computing a symbolic form of a System Jacobian $\mathbf{J}$, (b) finding a vector in $\mathrm{coker}(\mathbf{J})$ or $\ker(\mathbf{J})$, (c) checking the conditions, and (d) generating equations of the resulting DAE. These symbolic computations may seem expensive. Fortunately, we can combine block triangularization techniques [10, 11] with the conversion methods. When $\mathbf{J}$ is identically singular, we can locate the diagonal block of which the Jacobian is singular, and perform a conversion on this block. For DAEs that have a nontrivial block triangular form (BTF), that is, having more than one diagonal blocks, the symbolic computations can be reduced significantly.

For problems that do not have very complicated formulas, the aforementioned symbolic computations can be handled reasonably well by a symbolic tool, such as MATLAB's Symbolic Math Toolbox [14]. We combine this toolbox with our structural analysis software DAESA [6, 11], and build a prototype code that applies the conversion methods automatically. We aim to produce a solid implementation of these methods and incorporate this feature in a future version of DAESA.

Acknowledgements The authors wish to acknowledge with thanks the fundings for supporting their research: GT supported in part by the Ontario Research Fund (ORF, Canada), NSN supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and JDP supported in part by the Leverhulme Trust (UK). GT wishes to thank R. McKenzie for proofreading the draft of this article and providing valuable suggestions.

References

1. Barrio, R.: Performance of the Taylor series method for ODEs/DAEs. Appl. Math. Comp. **163**, 525–545 (2005)
2. Barrio, R.: Sensitivity analysis of ODES/DAES using the Taylor series method. SIAM J. Sci. Comput. **27**, 929–1947 (2006)
3. Brenan, K., Campbell, S., Petzold, L.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, second edn. SIAM, Philadelphia (1996)
4. Mattsson, S.E., Söderlind, G.: Index reduction in differential-algebraic equations using dummy derivatives. SIAM J. Sci. Comput. **14**(3), 677–692 (1993)
5. Nedialkov, N.S., Pryce, J.D.: Solving differential-algebraic equations by Taylor series (I): Computing Taylor coefficients. BIT Numerical Mathematics **45**, 561–591 (2005)
6. Nedialkov, N.S., Pryce, J.D., Tan, G.: Algorithm 948: DAESA: A Matlab tool for structural analysis of differential-algebraic equations: Software. ACM Trans. Math. Softw. **41**(2), 12:1–12:14 (2015). DOI 10.1145/2700586. URL http://doi.acm.org/10.1145/2700586
7. Nedialkov, N.S., Tan, G., Pryce, J.D.: Exploiting fine block triangularization and quasilinearity in differential-algebraic equation systems. arXiv:1411.4128 (2014). 18 pages, download link: http://arxiv.org/pdf/1411.4128v1
8. Pantelides, C.C.: The consistent initialization of differential-algebraic systems. SIAM. J. Sci. Stat. Comput. **9**, 213–231 (1988)
9. Pryce, J.D.: A simple structural analysis method for DAEs. BIT Numerical Mathematics **41**(2), 364–394 (2001)
10. Pryce, J.D., Nedialkov, N.S., Tan, G.: Graph theory, irreducibility, and structural analysis of differential-algebraic equation systems. arXiv:1411.4129 (2014). 24 pages, download link: http://arxiv.org/pdf/1411.4129v1
11. Pryce, J.D., Nedialkov, N.S., Tan, G.: DAESA: A Matlab tool for structural analysis of differential-algebraic equations: Theory. ACM Trans. Math. Softw. **41**(2), 9:1–9:20 (2015). DOI 10.1145/2689664. URL http://doi.acm.org/10.1145/2689664
12. Scholz, L., Steinbrecher, A.: A combined structural-algebraic approach for the regularization of coupled systems of DAEs. Tech. Rep. 30, Reihe des Instituts für Mathematik Technische Universität Berlin, Berlin, Germany (2013)
13. Tan, G., Nedialkov, N.S., Pryce, J.D.: Symbolic-numeric methods for improving structural analysis of differential-algebraic equation systems. arXiv:1505.03445 [cs.SC] (2015). 84 pages. Available at http://arxiv.org/pdf/1505.03445v1
14. The MathWorks, Inc.: Matlab Symbolic Math Toolbox (2015). URL https://www.mathworks.com/help/symbolic/index.html