BIT manuscript No.
(will be inserted by the editor)

Conversion Methods for Improving Structural Analysis of Differential-Algebraic Equation Systems

Guangning Tan · **Nedialko S. Nedialkov** ·
John D. Pryce

Received: date / Accepted: date

Abstract Structural analysis (SA) of a system of differential-algebraic equations (DAEs) is used to determine its index and which equations to be differentiated and how many times. Both Pantelides's algorithm and Pryce's Σ-method are equivalent: if one finds correct structural information, the other does also. Nonsingularity of the Jacobian produced by SA indicates success, which occurs on many problems of interest. However, these methods can fail on simple, solvable DAEs and give incorrect structural information including the index. This article investigates Σ-method's failures and presents two conversion methods for fixing them. Under certain conditions, both methods reformulate a DAE system on which the Σ-method fails into a locally equivalent problem on which SA is more likely to succeed. Aiming at achieving global equivalence between the original DAE system and the converted one, we provide a rationale for choosing a conversion from the applicable ones.

Keywords differential-algebraic equations · structural analysis · modeling · symbolic computation

Mathematics Subject Classification (2000) 34A09 · 65L80 · 41A58 · 68W30

1 Introduction.

G. Tan
School of Computational Science and Engineering, McMaster University,
1280 Main Street West, Hamilton, Ontario L8S 4K1, Canada, E-mail: tang4@mcmaster.ca

N. S. Nedialkov
Department of Computing and Software, McMaster University,
1280 Main Street West, , Hamilton, Ontario L8S 4K1, 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

Systems of differential-algebraic equation (DAEs) arise from a variety of engineering disciplines and are routinely generated by simulation and modeling software. Such systems can be large, sparse, nonlinear in highest derivatives, and of high differentiation index. Before a numerical solution method is applied, usually some structural analysis (SA) algorithm is used as a preprocessing tool to determine the (structural) index, number of degrees of freedom (DOF), constraints, and a set of variables and derivatives that need initial values. Such structural information can be useful for applying index reduction methods [8, 11] or regularization techniques [9, 23], so that we can call a standard DAE numerical code on a reduced DAE of differentiation index 1 or a regularized DAE, respectively. Some Taylor series methods [1, 2, 14, 15] are also built on SA.

Pantelides's SA algorithm [17] is widely used. Pryce's Σ-method [19] is equivalent to it, but can also handle high-order systems. Both SA methods produce the same structural index when applied to first-order systems [19, Theorem 5.8]. When SA *succeeds*, in the sense that it produces a nonsingular Jacobian, the structural index is an upper bound for the differentiation index, and often they are the same [19]. However, the structural index can be arbitrarily higher than the differentiation index, for example, on Reißig's family of DAEs of differentiation index 1 [21]. It has been shown in [26, §7.3] and [13, §5.2.5] that simple manipulations (similar to the linear combination techniques introduced in this article) on equations or variables can make the Σ-method report the correct (structural) index 1 on these DAEs.

Although the Σ-method succeeds on many problems of practical interest, it can fail—hence Pantelides's algorithm fails as well—on simple, solvable DAEs, producing an identically singular System Jacobian. Attempts to resolve SA's failures were made in existing literature. For example, Chowdhry *et al.* [6] propose the symbolic numeric index analysis, which handles first-order linear constant coefficient DAEs and some first-order DAEs where variables occur nonlinearly, but not all. Nor can their method detect complex variable substitutions or symbolic simplifications [6]. Scholz and Steinbrecher develop a structural-algebraic method to fix SA's failures on coupled systems [24]. During the remedy process where they take a linear combination of the algebraic equations, they also regularize the system so that the resulting DAE can be solved by a standard solver.

In this article, we investigate the Σ-method's failures and present two conversion methods that reformulate such a DAE in general form into an equivalent problem with the same solution (locally). After each conversion, provided some conditions are satisfied, the value of the signature matrix is guaranteed to decrease. We conjecture that this decrease usually leads to a better formulation of a problem, so that the SA may produce a (generically) nonsingular System Jacobian and hence succeed.

Compared to Scholz and Steinbrecher's approach in [24], our methods target a broader class of DAEs and hence are not limited to coupled systems. During a conversion, we also take into account the equations involving derivatives, not just the algebraic equations. Our expression substitution method can fix failure cases which taking a linear combination of equations cannot fix well; see Example 4.2 and §5.2. We also point out the key to remedying SA's failures is to reduce the value of a signature matrix.

The rest of this article is organized as follows. Section 2 summarizes the Σ-method theory and the notation we use throughout this article. Section 3 describes this SA's failures. Section 4 introduces the conversion methods and illustrates them with simple examples. Section 5 presents further two examples illustrating our methods and an example where neither method is applicable. Section 6 gives conclusions.

2 Summary of the Σ-method.

We consider DAEs in the general form

$$f_i(\,t, \text{the } x_j \text{ and derivatives of them}\,) = 0, \quad i = 1\!:\!n\,, \tag{2.1}$$

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

We let $\sigma(x_j, w)$ denote the order of the highest derivative to which variable x_j occurs in w, or $-\infty$ if neither x_j nor its derivatives[2] occur in w. Here w can be a scalar, a vector, or a matrix, depending on context.

The Σ-method constructs for a DAE (2.1) an $n \times n$ *signature matrix* Σ, whose (i, j) entry is $\sigma_{ij} := \sigma(x_j, f_i)$. 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 of Σ, such that the sum of these entries is maximized. This sum is the *value of* Σ, written $\mathrm{Val}(\Sigma)$. If $\mathrm{Val}(\Sigma)$ is finite, then the DAE is *structurally well posed* (SWP); otherwise, $\mathrm{Val}(\Sigma) = -\infty$ and the DAE is *structurally ill posed* (SIP). In the SIP case, there exists no one-to-one correspondence between equations and variables.

We henceforth consider the SWP case. Using a HVT, we find $2n$ integers $\mathbf{c} := (c_1, \ldots, c_n)$ and $\mathbf{d} := (d_1, \ldots, d_n)$ associated with the equations and variables of (2.1), respectively. These integers satisfy

$$c_i \ge 0 \quad \text{for all } i; \qquad d_j - c_i \ge \sigma_{ij} \quad \text{for all } i, j \text{ with equality on a HVT}\,. \tag{2.2}$$

We refer to such $\mathbf{c}$ and $\mathbf{d}$, written as a pair $(\mathbf{c};\mathbf{d})$, as a *valid offset pair*. It is not unique, but there exists a unique elementwise smallest solution $(\mathbf{c};\mathbf{d})$ of (2.2), which we refer to as the *canonical offset pair* [19].

Any valid $(\mathbf{c};\mathbf{d})$ can be used to prescribe a stage-by-stage *solution scheme* for solving DAEs by a Taylor series method. The derivatives of the solution are computed in stages $k = k_d, k_d + 1, \ldots, 0, 1, \ldots$ where $k_d := -\max_j d_j$. At each stage k, we solve a system comprising

$$0 = f_i^{(c_i+k)} \qquad \text{for all } i \text{ such that } c_i + k \ge 0 \tag{2.3}$$

for derivatives

$$x_j^{(d_j+k)} \qquad \text{for all } j \text{ such that } d_j + k \ge 0\,, \tag{2.4}$$

[1] The colon notation $p\!:\!q$ for integers p, q denotes either the unordered set or the enumerated list of integers i with $p \le i \le q$, depending on context.

[2] Throughout this article, "derivatives of x_j" include x_j itself as its 0th derivative: $x_j^{(l)} = x_j$ if $l = 0$.

using $x_j^{(<d_j+k)}$, $j = 1:n$, found in the previous stages. Here $z^{(<r)}$ is a short notation for $z, z', \ldots, z^{(r-1)}$, and $z^{(\le r)}$ includes $z^{(<r)}$ and $z^{(r)}$.

If the solution scheme (2.3–2.4) can be carried out for stages $k = k_d : 0$, and the derivatives $x_j^{(\le d_j)}$, $j = 1:n$, can be uniquely determined, then we say the solution scheme and the Σ-method *succeed*. Otherwise we say our SA *fails*, in the sense that the Jacobian used to solve (2.3) at some stage $k \in k_d : 0$ does not have full row rank.

The Jacobian used to solve (2.3) for stages $k \ge 0$ is called the *System Jacobian* of (2.1), an $n \times n$ matrix $\mathbf{J}(\mathbf{c};\mathbf{d}) := (J_{ij})$ defined by

$$J_{ij} := \frac{\partial f_i^{(c_i)}}{\partial x_j^{(d_j)}} = \frac{\partial f_i}{\partial x_j^{(d_j - c_i)}} = \begin{cases} \frac{\partial f_i}{\partial x_j^{(\sigma_{ij})}} & \text{if } d_j - c_i = \sigma_{ij}, \text{ and} \\ 0 & \text{otherwise ,} \end{cases} \tag{2.5}$$

with $i, j = 1:n$. The second "=" in (2.5) results from Griewank's Lemma [7] (see later Lemma 4.1), and the third "=" follows from (2.2).

Although a different $(\mathbf{c};\mathbf{d})$ produces a different solution scheme (2.3–2.4) and generally a different $\mathbf{J}(\mathbf{c};\mathbf{d})$, all $\mathbf{J}$'s nevertheless share the same determinant [14]. If one $\mathbf{J}$ is nonsingular and hence all $\mathbf{J}$'s are, then there exists (locally) a unique solution at a consistent point, as described in [19]. The SA now uses the canonical $(\mathbf{c};\mathbf{d})$ to determine the *structural index* ν_S; it is $\max_i c_i + 1$ if some $d_j = 0$, and $\max_i c_i$ otherwise. The number of degrees of freedom (DOF) is $\mathrm{Val}(\Sigma) = \sum_{(i,j)\in T} \sigma_{ij} = \sum_j d_j - \sum_i c_i$.

Example 2.1 We illustrate[3] the above concepts with the simple pendulum, a DAE of differentiation index 3.

$$\begin{aligned} 0 &= f_1 = x'' + x\lambda \\ 0 &= f_2 = y'' + y\lambda - G \\ 0 &= f_3 = x^2 + y^2 - \ell^2 \end{aligned} \qquad \Sigma = \begin{array}{c|ccc|c} & x & y & \lambda & c_i \\ \hline f_1 & 2^\bullet & & 0^\circ & 0 \\ f_2 & & 2^\circ & 0^\bullet & 0 \\ f_3 & 0^\circ & 0^\bullet & & 2 \\ \hline d_j & 2 & 2 & 0 & \end{array} \qquad \mathbf{J} = \begin{array}{c|ccc} & x'' & y'' & \lambda \\ \hline f_1 & 1 & & x \\ f_2 & & 1 & y \\ f_3'' & 2x & 2y & \end{array} \tag{2.6}$$

The state variables are x, y, and λ; G is gravity and $\ell > 0$ is the length of the pendulum. There are two HVTs of Σ, marked with $\bullet$ and $\circ$, respectively. A blank in Σ denotes $-\infty$, and a blank in $\mathbf{J}$ denotes 0. The row and column labels in $\mathbf{J}$ show equations and variables differentiated to order c_i and d_j, respectively.

Now that $\det(\mathbf{J}) = -2(x^2 + y^2) = -2\ell^2 \ne 0$, the SA succeeds. The structural index is $\nu_S = \min_i c_i + 1 = 3$, which equals the differentiation index. The pendulum has $\mathrm{Val}(\Sigma) = \sum_j d_j - \sum_i c_i = 2$ DOF.

3 Structural analysis's failure.

In the following two subsections, we identify respectively the two causes of SA's failures, which are not well distinguished in existing literature. One cause is due to not

[3] When we present a DAE example, we also present its signature matrix Σ, the canonical offset pair $(\mathbf{c};\mathbf{d})$, and the associated System Jacobian $\mathbf{J}$.

doing symbolic simplifications, and can be identified by a structurally singular Jacobian [14]. This failure is "easy" to fix, provided appropriate computer algebra operations can remove derivatives that occur of higher order than they should. The other cause of failures is more subtle and obscure, for the System Jacobian is identically singular but structurally nonsingular. Our methods fix the latter case in Section 4.

We use $u \not\equiv 0$ to mean that u is not identically zero for all values of the variables occurring in the expressions that define u. This u may be a scalar, a vector, or a matrix, depending on context. Similarly, we use $\det(\mathbf{A}) \not\equiv 0$ to mean that a matrix $\mathbf{A}$ is not identically singular, or *generically nonsingular*.

3.1 Symbolic cancellation may cause failure.

In the encoding of a DAE, an equation f_1 may be, for instance, $x_2 + (x_1x_2)' - x_1'x_2$ or $x_1 + x_2 + \cos^2 x_1' + \sin^2 x_1'$. We say a *symbolic cancellation* occurs in f_1, because these expressions simplify to $x_2 + x_1x_2'$ and $x_1 + x_2 + 1$, respectively. That is, f_1 does not *truly* depend on x_1'. We note that the problem of detecting such true dependence (which is equivalent to recognizing zero) in any expressions is unsolvable in general [22].

Codes like DAETS [15] and DAESA [16, 20], which are implemented through operator overloading and do not perform symbolic simplifications, compute a *formal* $\widetilde{\sigma}_{ij}$ instead of a *true* one when constructing the signature matrix. For example, both codes would find for f_1 above the formal $\widetilde{\sigma}_{11} = 1$ instead of the true $\sigma_{11} = 0$. By a formal $\widetilde{\sigma}_{ij}$, we mean that $x_j^{(\widetilde{\sigma}_{ij})}$ appears as a highest-order derivative (HOD) in the encoding of an equation f_i, while a true σ_{ij} means that f_i is not constant with respect to a HOD $x_j^{(\sigma_{ij})}$ and thus truly depends on it—equivalently $\partial f_i / \partial x_j^{(\sigma_{ij})} \not\equiv 0$. Obviously $\widetilde{\sigma}_{ij} \geq \sigma_{ij}$.

For a formally computed $\widetilde{\Sigma} = (\widetilde{\sigma}_{ij})$, also a valid offset pair $(\widetilde{\mathbf{c}}, \widetilde{\mathbf{d}})$ is found and a System Jacobian $\widetilde{\mathbf{J}}$ is derived from $(\widetilde{\mathbf{c}}, \widetilde{\mathbf{d}})$ and $\widetilde{\Sigma}$ by (2.5). Suppose symbolic cancellations happen in some f_i and make $\widetilde{\sigma}_{ij} > \sigma_{ij}$. Then f_i does not truly depend on $x_j^{(\widetilde{\sigma}_{ij})}$, and $\widetilde{J}_{ij}$ is identically zero by (2.5), whether $\widetilde{d}_j - \widetilde{c}_i = \widetilde{\sigma}_{ij}$ holds or not. In this case, $\widetilde{\mathbf{J}}$ has more identically zero entries than does a $\mathbf{J}$ based on the true Σ and $(\mathbf{c}; \mathbf{d})$, hence being more likely structurally singular.

Overestimating some σ_{ij} of Σ may seem dangerous to the SA's success. Fortunately, modern modeling environments usually perform simplifications on problem formulation [5, 10, 25]. They can reduce the occurrence of a structurally singular $\mathbf{J}$, when SA is applied. Theorems 5.1 and 5.2 in [14] also ensure that, if $\mathrm{Val}(\widetilde{\Sigma}) = \mathrm{Val}(\Sigma)$ and $\det(\mathbf{J}) \not\equiv 0$, then an offset pair $(\widetilde{\mathbf{c}}, \widetilde{\mathbf{d}})$ of the formal $\widetilde{\Sigma}$ is also valid for Σ, and $\det(\widetilde{\mathbf{J}}) = \det(\mathbf{J}) \not\equiv 0$. In this case, such an overestimation would treat some identically zero entries of $\mathbf{J}$ as nonzeros and simply make the solution scheme slightly less efficient; see [14, Examples 5.1 and 5.2]. By the same theorems, in the case $\mathrm{Val}(\widetilde{\Sigma}) > \mathrm{Val}(\Sigma)$, $\widetilde{\mathbf{J}}$ *must be* structurally singular.

3.2 SA can fail when $\mathbf{J}$ is structurally nonsingular.

Hereafter we focus on the case where an identically singular System Jacobian $\mathbf{J}$ is structurally nonsingular—that is, there exists a HVT T of Σ such that $J_{ij} \not\equiv 0$ for all $(i,j) \in T$. We shall simply say "identically singular" to refer to this case.

When $\mathbf{J}$ is identically singular, the DAE may be still solvable, but the way its equations are written may not properly reflect its structure. For example, if the pendulum DAE (2.6) $\mathbf{f} = 0$ is equivalently formulated as $\mathbf{Mf} = 0$ with $\mathbf{M}$ being a random nonsingular constant 3×3 matrix, then each row of Σ is $[2,2,0]$, the canonical offset pair is $(\mathbf{c};\mathbf{d}) = (0,0,0;2,2,0)$, and the resulting $\mathbf{J}$ is identically singular [13, §5.2.3].

Example 3.1 We illustrate a failure case with the following DAE of differentiation index 2 [3, p. 23]. Throughout this article we shall use $h_i(t)$ for driving functions.

$$
\begin{aligned}
0 &= f_1 = x' + ty' + h_1(t)\\
0 &= f_2 = x + ty + h_2(t)
\end{aligned}
\qquad
\Sigma = \begin{array}{c|cc|c}
 & x & y & c_i \\
f_1 & 1^\bullet & 1 & 0\\
f_2 & 0 & 0^\bullet & 1\\
d_j & 1 & 1 &
\end{array}
\qquad
\mathbf{J} = \begin{array}{c|cc|}
 & x' & y' \\
f_1 & 1 & t\\
f_2' & 1 & t
\end{array}
$$

The SA fails since $\mathbf{J}$ is identically singular but not structurally singular.

One simple fix is to replace f_1 by $\overline{f}_1 = -f_1 + f_2'$, which results in the algebraic system (hence of differentiation index 1) below; cf. [11, Example 5].

$$
\begin{aligned}
0 &= \overline{f}_1 = y - h_1(t) + h_2'(t)\\
0 &= f_2 = x + ty + h_2(t)
\end{aligned}
\qquad
\overline{\Sigma} = \begin{array}{c|cc|c}
 & x & y & c_i \\
\overline{f}_1 & & 0^\bullet & 0\\
f_2 & 0^\bullet & 0 & 0\\
d_j & 0 & 0 &
\end{array}
\qquad
\overline{\mathbf{J}} = \begin{array}{c|cc|}
 & x & y \\
\overline{f}_1 & & 1\\
f_2 & 1 & t
\end{array}
$$

The SA succeeds and we notice $\mathrm{Val}(\overline{\Sigma}) = 0 < 1 = \mathrm{Val}(\Sigma)$. This is a simple illustration of our linear combination method in §4.1.

Another simple fix is to introduce a variable $z = x + ty$ and to eliminate x in f_1 and f_2, leading to a nonsingular $\overline{\mathbf{J}}$.

$$
\begin{aligned}
0 &= \overline{f}_1 = -y + z' + h_1(t)\\
0 &= \overline{f}_2 = \qquad\quad z + h_2(t)
\end{aligned}
\qquad
\overline{\Sigma} = \begin{array}{c|cc|c}
 & y & z & c_i \\
\overline{f}_1 & 0^\bullet & 1 & 0\\
\overline{f}_2 & & 0^\bullet & 1\\
d_j & 0 & 1 &
\end{array}
\qquad
\overline{\mathbf{J}} = \begin{array}{c|cc|}
 & y & z' \\
\overline{f}_1 & -1 & 1\\
\overline{f}_2' & & 1
\end{array}
$$

This fix also gives $\mathrm{Val}(\overline{\Sigma}) = 0 < 1 = \mathrm{Val}(\Sigma)$, and is a simple illustration of our expression substitution method in §4.2.

A conjecture in [13, §5.2.3] attributed the SA's failure to a DAE "being not sparse enough to reflect its underlying mathematical structure" (sparsity refers to occurrence of only a few derivatives in each equation). However, as we shall see later, decreasing $\mathrm{Val}(\Sigma)$ may be the key to deriving a better problem formulation of a DAE. Our conversion methods aim to do so, and are the main contribution of this article.

4 Conversion methods.

We present two conversion methods that attempt to fix SA's failures in a systematic way. 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 §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 §4.2.

Given a DAE (2.1), we assume henceforth that $\mathrm{Val}(\Sigma)$ is finite and that the associated System Jacobian $\mathbf{J}$ is identically singular but structurally nonsingular. We also assume that the equations and variables in (2.1) are sufficiently differentiable, so that our methods fit into the Σ-method theory; see Theorem 4.2 in [19] and §3 in [14].

After a conversion, we denote the corresponding signature matrix as $\overline{\Sigma}$ and System Jacobian as $\overline{\mathbf{J}}$. If $\mathrm{Val}(\overline{\Sigma})$ is finite and $\overline{\mathbf{J}}$ is identically singular still, then we can perform another conversion, using either of the methods, provided the corresponding conditions are satisfied.

Suppose a sequence of conversions produces a solvable DAE with $\mathrm{Val}(\overline{\Sigma}) \geq 0$ and a generically nonsingular $\overline{\mathbf{J}}$. Since each conversion reduces the value of the signature matrix by at least one, the total number of conversions does not exceed the value of the original signature matrix. If the resulting system is SIP after a conversion, that is, $\mathrm{Val}(\overline{\Sigma}) = -\infty$, then we say the original DAE is *ill posed*.

4.1 Linear combination method.

Let $\mathbf{u} := [u_1, \ldots, u_n]^T \not\equiv \mathbf{0}$ be a nonzero n-vector function in the cokernel of $\mathbf{J}$, that is, $\mathbf{u} \in \mathrm{coker}(\mathbf{J})$ or equivalently $\mathbf{J}^T\mathbf{u} = \mathbf{0}$. We consider $\mathbf{J}$ and $\mathbf{u}$ as expressions comprising t and derivatives of the $x_j(t)$'s, although in fact they are generally functions evolving with t.

Lemma 4.1 ***(Griewank's Lemma)*** *[7] Let w be a function of t, the $x_j(t)$, $j = 1\!:\!n$, and derivatives of them. Denote $w^{(p)} = d^p w/dt^p$, where $p \geq 0$. If $\sigma(x_j, w) \leq q$, then*

$$\frac{\partial w}{\partial x_j^{(q)}} = \frac{\partial w'}{\partial x_j^{(q+1)}} = \cdots = \frac{\partial w^{(p)}}{\partial x_j^{(q+p)}} \,. \tag{4.1}$$

Denote

$$I := \{\, i \mid u_i \not\equiv 0 \,\}, \quad \underline{c} := \min_{i \in I} c_i, \quad \text{and} \quad L := \left\{\, i \in I \mid c_i = \underline{c} \,\right\}. \tag{4.2}$$

We give two lemmas and use them to prove Theorem 4.1, on which the LC method is based.

Lemma 4.2 *Assume that $\mathbf{u} \in \mathrm{coker}(\mathbf{J})$ and $\mathbf{u} \not\equiv \mathbf{0}$. If*

$$\sigma(x_j, \mathbf{u}) < d_j - \underline{c}, \quad \textit{for all } j = 1\!:\!n\,, \tag{4.3}$$

then

$$\sigma\left(x_j,\overline{f}\right) < d_j - \underline{c}, \quad \textit{for all } j = 1:n\,, \tag{4.4}$$

where

$$\overline{f} := \sum_{i\in I} u_i f_i^{(c_i-\underline{c})}\,. \tag{4.5}$$

Proof The formula for $\underline{c}$ gives $c_i - \underline{c} \geq 0$ for all $i \in I$. By (2.2), $\sigma(x_j, f_i) = \sigma_{ij} \leq d_j - c_i$. Applying Griewank's Lemma (4.1) to (2.5) with $w = f_i$ and $q = c_i - \underline{c}$ yields

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j^{(d_j-c_i)}} = \frac{\partial f_i^{(c_i-\underline{c})}}{\partial x_j^{(d_j-c_i+c_i-\underline{c})}} = \frac{\partial f_i^{(c_i-\underline{c})}}{\partial x_j^{(d_j-\underline{c})}} \qquad \text{for } i \in I \text{ and all } j = 1:n\,. \tag{4.6}$$

This shows that such an $f_i^{(c_i-\underline{c})}$ depends on $x_j^{(\leq d_j-\underline{c})}$ only. Then for all $j = 1:n$,

$$\begin{aligned}
\frac{\partial \overline{f}}{\partial x_j^{(d_j-\underline{c})}} &= \frac{\partial\left(\sum_{i\in I} u_i f_i^{(c_i-\underline{c})}\right)}{\partial x_j^{(d_j-\underline{c})}} && \text{by the definition of } \overline{f} \text{ in (4.5)}\\
&= \sum_{i\in I} u_i \frac{\partial f_i^{(c_i-\underline{c})}}{\partial x_j^{(d_j-\underline{c})}} = \sum_{i\in I} u_i \mathbf{J}_{ij} && \text{by (4.3) and then (4.6)}\\
&= (\mathbf{J}^T\mathbf{u})_j = 0 && \text{since } \mathbf{u} \in \operatorname{coker}(\mathbf{J})\ .
\end{aligned}$$

Hence $\overline{f}$ depends on $x_j^{(<d_j-\underline{c})}$ only, for all j—this results in the inequality in (4.4). $\square$

The following lemma is straightforward to prove.

Lemma 4.3 *Assume that an $n\times n$ signature matrix Σ has a finite* $\mathrm{Val}(\Sigma)$ *and a valid offset pair* $(\mathbf{c};\mathbf{d})$. *Given a row of index l, if we replace in row l all entries σ_{lj} by $\overline{\sigma}_{lj} < d_j - c_l$, then the resulting signature matrix $\overline{\Sigma}$ satisfies* $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$.

Theorem 4.1 *Assume that a DAE has a finite* $\mathrm{Val}(\Sigma)$, *a valid offset pair* $(\mathbf{c};\mathbf{d})$, *and an identically singular* $\mathbf{J}$. *Assume a nonzero vector* $\mathbf{u} \in \operatorname{coker}(\mathbf{J})$. *Let I, $\underline{c}$, and L be as defined in (4.2). If* $\mathbf{u}$ *satisfies (4.3) and we replace f_l by $\overline{f}_l = \overline{f}$ in (4.5) for a given $l \in L$, then the resulting signature matrix $\overline{\Sigma}$ satisfies* $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$, *and the converted DAE and the original one have the same solution (if any) provided $u_l \neq 0$.*

We call (4.3) the condition for the LC method. The strict decrease $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$ results from Lemmas 4.2 and 4.3. The last claim can be shown by using (4.2) and (4.5): we can recover the replaced equation $f_l = \left(\overline{f}_l - \sum_{i\in I\setminus\{l\}} u_i f_i^{(c_i-\underline{c})}\right)/u_l(t)$ if $u_l(t) \neq 0$ at t. Since $f_l = 0$ if and only if $\overline{f}_l = 0$, both DAEs have the same solution at t, and we say they are (locally) *equivalent*. The reader is referred to [26, §4.1] for details on the equivalence of DAEs.

Example 4.1 We illustrate the LC method with the following simple example:

$$
\begin{aligned}
0 &= f_1 = -x_1' + x_3 & \qquad 0 &= f_3 = x_1 x_2 + h_1(t) \\
0 &= f_2 = -x_2' + x_4 & \qquad 0 &= f_4 = x_1 x_4 + x_2 x_3 + x_1 + x_2 + h_2(t)\,.
\end{aligned}
$$

$$
\Sigma = \begin{array}{c|cccc|c}
 & x_1 & x_2 & x_3 & x_4 & c_i \\ \hline
f_1 & 1^\bullet & & 0 & & 0 \\
f_2 & & 1 & & 0^\bullet & 0 \\
f_3 & 0 & 0^\bullet & & & 1 \\
f_4 & \colorbox{lightgray}{0} & \colorbox{lightgray}{0} & 0^\bullet & 0 & 0 \\ \hline
d_j & 1 & 1 & 0 & 0 &
\end{array}
\qquad
\mathbf{J} = \begin{array}{c|cccc|}
 & x_1' & x_2' & x_3 & x_4 \\ \hline
f_1 & -1 & & 1 & \\
f_2 & & -1 & & 1 \\
f_3' & x_2 & x_1 & & \\
f_4 & & & x_2 & x_1 \\ \hline
\end{array}
$$

A shaded entry $\colorbox{lightgray}{$\sigma_{ij}$}$ in Σ denotes a position (i,j) where $d_j - c_i > \sigma_{ij} \geq 0$ and hence $J_{ij} \equiv 0$ by the formula (2.5) for $\mathbf{J}$. The SA fails here since $\det(\mathbf{J}) \equiv 0$.

We choose $\mathbf{u} = [x_2, x_1, 1, -1]^T \in \operatorname{coker}(\mathbf{J})$. Then (4.2) becomes

$$
I = \{\, i \mid u_i \not\equiv 0 \,\} = \{\, 1{:}4 \,\}, \quad \underline{c} = \min_{i \in I} c_i = 0, \quad L = \{\, i \in I \mid c_i = \underline{c} \,\} = \{\, 1,2,4 \,\}\,.
$$

Checking the condition (4.3) is not difficult; for example, $\sigma(x_1, \mathbf{u}) = 0 < 1 = d_1 - \underline{c}$.

We pick $l = 4 \in L$ (we shall reason why this choice is desirable) and replace f_4 by

$$
\overline{f}_4 = \sum_{i \in I} u_i f_i^{(c_i - \underline{c})} = x_2 f_1 + x_1 f_2 + f_3' - f_4 = -x_1 - x_2 + h_1'(t) - h_2(t)\,.
$$

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

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

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

As the value of $u_l(t)$ may also evolve with t during integration, it would be desirable to select a u_l such that the equivalence between both the original and converted DAEs is *global*, in the sense that they *always* have the same solution (if any). In this way we can stick to solving the converted system. Hence, we wish to select, whenever possible, an $l \in L$ such that u_l would be an expression *never* becoming zero, e.g., a nonzero constant, $x_1^2 + 1$, or $2 + \cos x_2$.

Since determining whether an expression is identically zero is unsolvable in general [22], we consider a (nonzero) constant u_l as the most preferable choice among all $l \in L$, and derive a set $\overline{L} := \{\, l \in L \mid u_l \text{ is constant} \,\}$ that contains all l for such u_l.

We summarize the steps of the LC method.

Step 1. Obtain a symbolic form of $\mathbf{J}$.
Step 2. Find a vector $\mathbf{u} \in \operatorname{coker}(\mathbf{J})$ and derive I, $\underline{c}$, and L as defined in (4.2).

Step 3. Check condition (4.3). If it is not satisfied, then set $L \leftarrow \emptyset$ to mean that the LC method is not applicable; otherwise proceed to Step 4.
Step 4. $\overline{L} \leftarrow \{\, l \in L \mid u_l \text{ is constant} \,\}$. If $\overline{L} \neq \emptyset$, then choose $l \in \overline{L}$; otherwise $l \in L$.
Step 5. Replace f_l by $\overline{f}_l = \overline{f}$ as defined in (4.5).

We use L and $\overline{L}$ to decide a desirable conversion method; see Table 4.1 in §4.3. We show below that the LC method cannot fix the following (artificially constructed) DAE (4.7) because the condition (4.3) is not satisfied.

Example 4.2 Consider $0 = (f_1, f_2)$, where

$$\begin{aligned} f_1 &= x_1 + e^{-x_1'-x_2x_2''} + h_1(t) \\ f_2 &= x_1 + x_2x_2' + x_2^2 + h_2(t) \end{aligned} \qquad \Sigma = \begin{array}{c|cc|c} & x_1 & x_2 & c_i \\ \hline f_1 & 1^\bullet & 2 & 0 \\ f_2 & 0 & 1^\bullet & 1 \\ \hline d_j & 1 & 2 & \end{array} \qquad \mathbf{J} = \begin{array}{c|cc} & x_1' & x_2'' \\ \hline f_1 & -\alpha & -\alpha x_2 \\ f_2' & 1 & x_2 \end{array}, \tag{4.7}$$

and $\alpha = e^{-x_1'-x_2x_2''}$. Obviously SA fails.

Take $\mathbf{u} = [\alpha^{-1}, 1]^T = \left[e^{x_1'+x_2x_2''}, 1\right]^T \in \operatorname{coker}(\mathbf{J})$. Using (4.2) gives $I = \{1,2\}$, $\underline{c} = 0$, and $L = \{1\}$. The LC condition (4.3) is violated since $\sigma(x_j, \mathbf{u}) = d_j - \underline{c}$ for $j = 1,2$. If we choose $l = 1 \in L$ and replace f_1 by

$$\overline{f}_1 = u_1 f_1 + u_2 f_2' = \beta + x_1' + x_2x_2'' + (x_2')^2 + 2x_2x_2' + h_2'(t)\,,$$

where $\beta = e^{x_1'+x_2x_2''}(x_1 + h_1(t)) + 1$, then SA fails still on the resulting DAE $0 = \left(\overline{f}_1, f_2\right)$ with $\operatorname{Val}(\overline{\Sigma}) = \operatorname{Val}(\Sigma) = 2$ and $\det(\overline{\mathbf{J}}) \equiv 0$.

$$\overline{\Sigma} = \begin{array}{c|cc|c} & x_1 & x_2 & c_i \\ \hline \overline{f}_1 & 1^\bullet & 2 & 0 \\ f_2 & 0 & 1^\bullet & 1 \\ \hline d_j & 1 & 2 & \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c|cc} & x_1' & x_2'' \\ \hline \overline{f}_1 & \beta & \beta x_2 \\ f_2' & 1 & x_2 \end{array}$$

We shall show in Example 4.3 that the ES method can reduce $\operatorname{Val}(\Sigma)$ and fix (4.7).

4.2 Expression substitution method.

Let $\mathbf{v} := [v_1, \ldots, v_n]^T \not\equiv \mathbf{0}$ be a nonzero n-vector function in the kernel of $\mathbf{J}$, that is, $\mathbf{v} \in \ker(\mathbf{J})$, or equivalently $\mathbf{J}\mathbf{v} = \mathbf{0}$. Denote

$$\begin{aligned} &J := \{\, j \mid v_j \not\equiv 0 \,\}, \quad s := |J|\,, \\ &M := \{\, i \mid d_j - c_i = \sigma_{ij} \text{ for some } j \in J \,\}, \quad \text{and} \quad \overline{c} := \max_{i \in M} c_i\,. \end{aligned} \tag{4.8}$$

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

$$y_j := x_j^{(d_j - \overline{c})} - \frac{v_j}{v_l} \cdot x_l^{(d_l - \overline{c})} \qquad \text{for all } j \in J \setminus \{\, l \,\}\,. \tag{4.9}$$

In each f_i, we

$$\text{replace every } x_j^{(\sigma_{ij})} = x_j^{(d_j - c_i)} \text{ with } j \in J \setminus \{\, l \,\} \text{ by } \left(y_j + \frac{v_j}{v_l} \cdot x_l^{(d_l - \overline{c})}\right)^{(\overline{c} - c_i)}. \tag{4.10}$$

From the formula (4.8) for M, these replacements (or substitutions) occur only in f_i's with $i \in M$, because at least one equality $d_j - c_i = \sigma_{ij}$ must hold for some $j \in J$.

After the replacements, denote each equation by $\overline{f}_i$ (for all $i \notin M$, $\overline{f}_i$ and f_i are the same). Equivalent to (4.9) are $s-1$ equations

$$0 = g_j := -y_j + x_j^{(d_j - \overline{c})} - \frac{v_j}{v_l} \cdot x_l^{(d_l - \overline{c})} \quad \text{for all } j \in J \setminus \{\, l \,\} \tag{4.11}$$

that prescribe the substitutions in (4.10). Appending (4.11) to the $\overline{f}_i$'s results in an enlarged DAE consisting of

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

The ES method is based on the following theorem.

Theorem 4.2 *Assume that a DAE has a finite* $\mathrm{Val}(\Sigma)$*, a valid offset pair* $(\mathbf{c};\mathbf{d})$*, and an identically singular* $\mathbf{J}$*. Assume that a nonzero vector* $\mathbf{v} \in \ker(\mathbf{J})$*. Let* J*,* s*, and* $\overline{c}$ *be as defined in (4.8). Assume that*

$$\sigma(x_j, \mathbf{v}) \begin{cases} < d_j - \overline{c} & \text{if } j \in J \\ \le d_j - \overline{c} & \text{otherwise}\,, \end{cases} \quad \text{and} \quad d_j - \overline{c} \ge 0 \quad \text{for all } j \in J\,. \tag{4.12}$$

For a given $l \in J$*, if we*

1) append $s-1$ *equations* g_j*, for all* $j \in J \setminus \{\, l \,\}$*, as defined in (4.11) and*
2) perform substitutions in f_i*, for all* $i = 1:n$*, as described by (4.10),*

then the resulting signature matrix $\overline{\Sigma}$ *satisfies* $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$*, and the converted DAE and the original one have the same solution (if any) provided* $v_l \neq 0$*.*

We call (4.12) the conditions for the ES method.

Example 4.3 We illustrate the ES method on the DAE (4.7). Suppose we choose $\mathbf{v} = [x_2, -1]^T \in \ker(\mathbf{J})$. Then (4.8) becomes

$$J = \{\, 1, 2 \,\}, \quad s = |J| = 2, \quad M = \{\, 1, 2 \,\}, \quad \text{and} \quad \overline{c} = \max_{i \in M} c_i = c_2 = 1\,.$$

We can apply the ES method as the conditions (4.12) hold:

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

We choose $l = 2 \in J$. Now $J \setminus \{\, l \,\} = \{\, 1 \,\}$. Using (4.11), we append the equation

$$0 = g_1 = -y_1 + x_1^{(d_1 - \overline{c})} - \frac{v_1}{v_2} \cdot x_2^{(d_2 - \overline{c})} = -y_1 + x_1 + x_2 x_2'\,,$$

which meanwhile defines the newly introduced variable y_1 corresponding to x_1. Then we replace x_1' by $(y_1 - x_2x_2')'$ in f_1 to obtain $\overline{f}_1$, and replace x_1 by $y_1 - x_2x_2'$ in f_2 to obtain $\overline{f}_2$. The resulting DAE $0 = (\overline{f}_1, \overline{f}_2, g_1)$ and its SA results are shown below.

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

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

We prove a lemma related to Theorem 4.2, using the following assumptions.

(a) Without loss of generality, we assume that the entries $v_j \not\equiv 0$ are in the first s positions of $\mathbf{v}$, that is, $\mathbf{v} = [v_1, \ldots, v_s, 0, \ldots, 0]^T$. Then $J = \{1, \ldots, s\}$ in (4.8).

(b) We introduce one more variable $y_l = x_l^{(d_l - \overline{c})}$ for the chosen $l \in J$, and append correspondingly one more equation $0 = g_l = -y_l + x_l^{(d_l - \overline{c})}$.

Lemma 4.4 *Let $(\mathbf{c};\mathbf{d}) = (c_1, \ldots, c_n; d_1, \ldots, d_n)$ be a valid offset pair of Σ. Let $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$ be the two $(n+s)$-vectors defined as*

$$\widetilde{d}_j := \begin{cases} d_j & \text{if } j = 1:n \\ \overline{c} & \text{if } j = n+1:n+s \end{cases} \quad \text{and} \quad \widetilde{c}_i := \begin{cases} c_i & \text{if } i = 1:n \\ \overline{c} & \text{if } i = n+1:n+s\,, \end{cases} \tag{4.13}$$

where $\overline{c}$ is as defined (4.8). Then the signature matrix $\overline{\Sigma}$ of the resulting DAE from the ES method has the form in Figure 4.1.

The proof of this lemma is rather technical; we present it in Appendix A. Using Lemma 4.4, we prove Theorem 4.2.

Proof We prove first the strict decrease $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$. Let $\overline{T}$ be a HVT of $\overline{\Sigma}$. By Lemma 4.4,

$$\begin{aligned} \mathrm{Val}(\overline{\Sigma}) &= \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} \le \sum_{(i,j)\in\overline{T}} (\widetilde{d}_j - \widetilde{c}_i) && \text{since } \widetilde{d}_j - \widetilde{c}_i \ge \overline{\sigma}_{ij} \text{ for all } i,j \\ &= \sum_{j=1}^{n+s} \widetilde{d}_j - \sum_{i=1}^{n+s} \widetilde{c}_i = \sum_{j=1}^{n} d_j - \sum_{i=1}^{n} c_i = \mathrm{Val}(\Sigma) && \text{by (4.13)}\,. \end{aligned}$$

We assert $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$, and show that an equality leads to a contradiction.

Assume that $\mathrm{Val}(\overline{\Sigma}) = \mathrm{Val}(\Sigma)$. Then there exists a transversal $\overline{T}$ of $\overline{\Sigma}$ such that

$$\widetilde{d}_j - \widetilde{c}_i = \overline{\sigma}_{ij} > -\infty \qquad \text{for all } (i,j) \in \overline{T}\,. \tag{4.14}$$

Consider $(i_1, 1), \ldots, (i_s, s) \in \overline{T}$ for the first s columns. Since the y_l column has only one finite entry $\overline{\sigma}_{n+l,n+l} = 0$, position $(n+l, n+l)$ is in $\overline{T}$, and thus only $s-1$ numbers of $i_1, \ldots, i_s$ are greater than n, leaving at least one of them in $1:n$. In other words,

$$
\begin{array}{c|ccc|c|ccc|c}
 & x_1 \cdots x_{l-1} & x_l & x_{l+1} \cdots x_s & x_{s+1} \cdots x_n & y_1 \cdots y_{l-1} & y_l & y_{l+1} \cdots y_s & \widetilde{c}_i \\
\hline
\overline{f}_1 & & & & & & -\infty & & c_1 \\
\vdots & & < & & \leq & \leq & \vdots & \leq & \vdots \\
\overline{f}_n & & & & & & -\infty & & c_n \\
\hline
g_1 & = \quad < & = & & \leq & 0 & & & \overline{c} \\
\vdots & \ddots & \vdots & < & & \ddots & & -\infty & \vdots \\
g_l & & = & & -\infty \cdots -\infty & & 0 & & \overline{c} \\
\vdots & < & \vdots & < \quad \ddots & \leq & -\infty & & \ddots & \vdots \\
g_s & & = & = & & & & 0 & \overline{c} \\
\hline
\widetilde{d}_j & d_1 \cdots d_{l-1} & d_l & d_{l+1} \cdots d_s & d_{s+1} \cdots d_n & \overline{c} \cdots \overline{c} & \overline{c} & \overline{c} \cdots \overline{c} &
\end{array}
$$

Fig. 4.1: The form of $\overline{\Sigma}$ for the resulting DAE by the ES method, assuming $J = \{1,\ldots,s\}$ in (4.8). The $<$, $\leq$, and $=$ mean the relations between $\overline{\sigma}_{ij}$ and $\widetilde{d}_j - \widetilde{c}_i$. For instance, every $\overline{\sigma}_{ij}$ whose (i,j) position is in the region marked with "$\leq$" is $\leq \widetilde{d}_j - \widetilde{c}_i$.

there exists a position $(r,j) \in \overline{T}$ with $1 \leq r \leq n$ and $1 \leq j \leq s$ in the "$<$" region in Figure 4.1. Hence $\widetilde{d}_j - \widetilde{c}_r > \overline{\sigma}_{rj}$, which yields a contradiction of (4.14). Therefore $\mathrm{Val}(\overline{\Sigma}) < \mathrm{Val}(\Sigma)$. Finally we remove the y_l column and its matched row g_l. The resulting signature matrix still has $\mathrm{Val}(\overline{\Sigma})$, since $(n+l, n+l) \in \overline{T}$ and $\overline{\sigma}_{n+l,n+l} = 0$.

If $v_l \neq 0$, then y_j in (4.9) is well defined. Both the converted DAE and the original one have the same solution in that we can recover the latter by reverting all expression substitutions occurring in $\overline{f}_i$ and removing all introduced variables y_j and equations g_j. $\square$

Choosing a v_l in the ES method is similar to choosing a u_l in the LC method. We can introduce well-defined y_j in (4.9) and perform the conversion process for $l \in J$ only if $v_l(t) \neq 0$ at t, whence the original and converted DAEs are locally equivalent; see details in [26, §4.2]. Therefore, it is again more desirable to choose a variable index $l \in J$ for which v_l is a (nonzero) constant, so that global equivalence is achieved. We hence define a set $\overline{J} := \{\, l \in J \mid v_l \text{ is constant} \,\}$, and whenever it is nonempty, we choose an l in it. We summarize the steps of the ES method below.

Step 1. Obtain a symbolic form of $\mathbf{J}$.
Step 2. Find a vector $\mathbf{v} \in \ker(\mathbf{J})$ and derive J, s, M, $\overline{c}$ as defined in (4.8).
Step 3. Check conditions (4.12). If any of them is not satisfied, then set $J \leftarrow \emptyset$ to mean that the ES method is not applicable; otherwise proceed to Step 4.
Step 4. $\overline{J} \leftarrow \{\, l \in J \mid v_l \text{ is constant} \,\}$. If $\overline{J} \neq \emptyset$, then choose an $l \in \overline{J}$; otherwise an $l \in J$.
Step 5. For each $j \in J \setminus \{\, l \,\}$, append the corresponding equation g_j defined in (4.11).
Step 6. Replace each $x_j^{(d_j - c_i)}$ in f_i by $\big(y_j + (v_j/v_l) \cdot x_l^{(d_l - \overline{c})}\big)^{(\overline{c} - c_i)}$, for all $i \in M$ and all $j \in J \setminus \{\, l \,\}$.

Step 7. (Optional) For consistence, rename variables y_j, $j \in J \setminus \{l\}$, to $x_{n+1}, \dots, x_{n+s-1}$, and rename equations g_j, $j \in J \setminus \{l\}$, to $f_{n+1}, \dots, f_{n+s-1}$.

The sets J and $\overline{J}$ are used to decide a desirable conversion method; see §4.3 below.

4.3 Which method to choose?

We present our rationale for choosing a conversion method in Table 4.1 and base our choice on the following observations. If both methods are applicable, then we consider as priority the equivalence between the original and the converted DAEs, and hence wish to perform a conversion that ensures global equivalence. This is done by choosing a nonzero constant u_l for the LC method or v_l for the ES method; recall discussions in §4.1 and §4.2. In the case where both methods guarantee global equivalence or neither of them does, we choose the LC method, since it is simpler to perform and maintains the problem size.

		ES method		
		$\overline{J} \neq \emptyset$	$\overline{J} = \emptyset$ and $J \neq \emptyset$	$J = \emptyset$
	$\overline{L} \neq \emptyset$	LC	LC	LC
LC method	$\overline{L} = \emptyset$ and $L \neq \emptyset$	ES	LC	LC
	$L = \emptyset$	ES	ES	–

Table 4.1: Rationale for choosing a conversion method.

5 More examples.

We show in §5.1 how to iterate the LC method on a linear constant coefficient DAE, illustrate in §5.2 the ES method with a modified pendulum problem by a linear transformation of the state variables, and present in §5.3 a DAE where neither of conversion methods is applicable, while a conversion can be easily found by observation.

5.1 A linear constant coefficient DAE.

Consider a linear constant coefficient DAE [24, Example 3][4], on which SA fails.

$$
\begin{aligned}
0 = f_1 &= -x_1' + x_3 + h_1(t) \qquad & 0 = f_3 &= \quad x_2 + x_3 + x_4 + h_3(t) \\
0 = f_2 &= -x_2' + x_4 + h_2(t) \qquad & 0 = f_4 &= -x_1 + x_3 + x_4 + h_4(t)\,.
\end{aligned}
$$

[4] We consider it with parameters $\beta = \varepsilon = 1$, $\alpha_1 = \alpha_2 = \delta = 1$, and $\gamma = -1$, and we use subscripts for parameter indices. The equations g_1, g_2 are renamed f_3, f_4 and the variables y_1, y_2 are renamed x_3, x_4.

$$
\Sigma_0 = \begin{array}{c|cccc|c}
 & x_1 & x_2 & x_3 & x_4 & c_i \\ \hline
f_1 & 1^\bullet & & 0 & & 0 \\
f_2 & & 1^\bullet & & 0 & 0 \\
f_3 & & 0 & 0^\bullet & 0 & 0 \\
f_4 & 0 & & 0 & 0^\bullet & 0 \\ \hline
d_j & 1 & 1 & 0 & 0 &
\end{array}
\qquad
\mathbf{J}_0 = \begin{array}{c|cccc|}
 & x_1' & x_2' & x_3 & x_4 \\ \hline
f_1 & -1 & & 1 & \\
f_2 & & -1 & & 1 \\
f_3 & & & 1 & 1 \\
f_4 & & & 1 & 1 \\ \hline
\end{array}
$$

We use a subscript in Σ_0 and $\mathbf{J}_0$ to mean an iteration number.

We first find $\mathbf{u} = [0,0,-1,1]^T \in \operatorname{coker}(\mathbf{J}_0)$ and derive $L = \overline{L} = \{3,4\}$. Obviously the LC condition (4.3) is satisfied. We choose $l = 3$ and replace f_3 by $\overline{f}_3 = -f_3 + f_4$.

$$
\Sigma_1 = \begin{array}{c|cccc|c}
 & x_1 & x_2 & x_3 & x_4 & c_i \\ \hline
f_1 & 1^\bullet & & 0 & & 0 \\
f_2 & & 1 & & 0^\bullet & 0 \\
\overline{f}_3 & 0 & 0^\bullet & & & 1 \\
f_4 & 0 & & 0^\bullet & 0 & 0 \\ \hline
d_j & 1 & 1 & 0 & 0 &
\end{array}
\qquad
\mathbf{J}_1 = \begin{array}{c|cccc|}
 & x_1' & x_2' & x_3 & x_4 \\ \hline
f_1 & -1 & & 1 & \\
f_2 & & -1 & & 1 \\
\overline{f}_3' & -1 & -1 & & \\
f_4 & & & 1 & 1 \\ \hline
\end{array}
$$

The SA fails still, so we iterate the LC method: find $\mathbf{u} = [-1,-1,1,1]^T \in \operatorname{coker}(\mathbf{J}_1)$, derive $L = \overline{L} = \{1,2,4\}$, and replace f_1 by $\overline{f}_1 = -f_1 - f_2 + \overline{f}_3' + f_4$.

$$
\Sigma_2 = \begin{array}{c|cccc|c}
 & x_1 & x_2 & x_3 & x_4 & \tilde{c}_i \\ \hline
\overline{f}_1 & 0^\bullet & & & & 1 \\
f_2 & & 1 & & 0^\bullet & 0 \\
\overline{f}_3 & 0 & 0^\bullet & & & 1 \\
f_4 & 0 & & 0^\bullet & 0 & 0 \\ \hline
\tilde{d}_j & 1 & 1 & 0 & 0 &
\end{array}
\qquad
\mathbf{J}_2 = \begin{array}{c|cccc|}
 & x_1' & x_2' & x_3 & x_4 \\ \hline
\overline{f}_1' & -1 & & & \\
f_2 & & -1 & & 1 \\
\overline{f}_3' & -1 & -1 & & \\
f_4 & & & 1 & 1 \\ \hline
\end{array}
$$

The SA succeeds since $\det(\mathbf{J}_2) = 1$. Note $\operatorname{Val}(\Sigma_2) = 0 < \operatorname{Val}(\Sigma_1) = 1 < \operatorname{Val}(\Sigma_0) = 2$.

5.2 Modified pendulum by change of variables.

For the pendulum DAE (2.6), if we perform a linear transformation on x, y, λ:

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ \lambda \end{bmatrix}, \tag{5.1}
$$

then the SA fails on the resulting problem.

$$
\begin{aligned}
0 = f_1 &= x_1'' + x_2'' + (x_1 + x_2)(x_3 + x_1) \\
0 = f_2 &= x_2'' + x_3'' + (x_2 + x_3)(x_3 + x_1) - G \\
0 = f_3 &= (x_1 + x_2)^2 + (x_2 + x_3)^2 - \ell^2 .
\end{aligned}
$$

$$\Sigma = \begin{array}{cc} & \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} & \left[\begin{array}{ccc} 2 & 2 & \boxed{0} \\ \boxed{0} & 2 & 2 \\ 0 & 0 & 0 \end{array}\right] \\ d_j & \begin{array}{ccc} 2 & 2 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 2 \\ \end{array} \qquad \mathbf{J} = \begin{array}{cc} & \begin{array}{ccc} x_1'' & x_2'' & x_3'' \end{array} \\ \begin{array}{c} f_1 \\ f_2 \\ f_3'' \end{array} & \left[\begin{array}{ccc} 1 & 1 & \\ & 1 & 1 \\ 2(x_1+x_2) & 2(x_1+2x_2+x_3) & 2(x_2+x_3) \end{array}\right] \end{array}$$

We first attempt the LC method: find $\mathbf{u} = [2(x_1+x_2), 2(x_2+x_3), -1]^T \in \operatorname{coker}(\mathbf{J})$ and derive $L = \{1,2\}$ by (4.2). For all $l \in L$, u_l is not a constant, so $L \neq \emptyset$ and $\overline{L} = \emptyset$. Then we try the ES method to seek a conversion that guarantees global equivalence.

We show below how the ES method reveals the linear transformation (5.1) without having the knowledge about the equations. Compute $\mathbf{v} = [1,-1,1]^T \in \ker(\mathbf{J})$ and find $J = \overline{J} = \{1,2,3\}$, $s = |J| = 3$, $M = \{1,2,3\}$, and $\overline{c} = 2$ using (4.8). Obviously the ES conditions (4.12) are satisfied, and the method guarantees global equivalence because $\overline{J} \neq \emptyset$. We show the conversion for $l = 1 \in \overline{J}$. As $J \setminus \{l\} = \{2,3\}$, we append the equations $0 = g_2 = -y_2 + x_2 + x_1$ and $0 = g_3 = -y_3 + x_3 - x_1$, which meanwhile define the newly introduced variables y_2, y_3 corresponding to x_2, x_3, respectively. Then we perform the expression substitutions in the below table.

substitute	for	in
$y_2'' - x_1''$	x_2''	f_1, f_2
$y_3'' + x_1''$	x_3''	f_2
$y_2 - x_1$	x_2	f_3
$y_3 + x_1$	x_3	f_3

After the substitutions, we rename y_2, y_3 to x_4, x_5 and g_2, g_3 to $\overline{f}_4, \overline{f}_5$. The SA succeeds on the resulting DAE with $\det(\overline{\mathbf{J}}) = -4\ell^2 \neq 0$.

$$\begin{aligned} 0 &= \overline{f}_1 = x_4'' + (x_1+x_2)(x_3+x_1) \\ 0 &= \overline{f}_2 = x_4'' + x_5'' + (x_2+x_3)(x_3+x_1) - G \\ 0 &= \overline{f}_3 = x_4^2 + (x_4+x_5)^2 - \ell^2 \\ 0 &= \overline{f}_4 = -x_4 + x_2 + x_1 \\ 0 &= \overline{f}_5 = -x_5 + x_3 - x_1 \end{aligned}$$

$$\overline{\Sigma} = \begin{array}{cc} & \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{array}{c} \overline{f}_1 \\ \overline{f}_2 \\ \overline{f}_3 \\ \overline{f}_4 \\ \overline{f}_5 \end{array} & \left[\begin{array}{ccccc} 0 & 0 & 0 & 2^\bullet & \\ 0^\bullet & 0 & 0 & 2 & 2 \\ & & & 0 & 0^\bullet \\ 0 & 0^\bullet & & \boxed{0} & \\ 0 & & 0^\bullet & & \boxed{0} \end{array}\right] \\ d_j & \begin{array}{ccccc} 0 & 0 & 0 & 2 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ \end{array}$$

$$\overline{\mathbf{J}} = \begin{array}{cc} & \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4'' & x_5'' \end{array} \\ \begin{array}{c} \overline{f}_1 \\ \overline{f}_2 \\ \overline{f}_3'' \\ \overline{f}_4 \\ \overline{f}_5 \end{array} & \left[\begin{array}{ccccc} 2x_1+x_2+x_3 & x_3+x_1 & x_1+x_2 & 1 & \\ x_2+x_3 & x_3+x_1 & x_1+x_2+2x_3 & 1 & 1 \\ & & & 2(2x_4+x_5) & 2(x_4+x_5) \\ 1 & 1 & & & \\ -1 & & 1 & & \end{array}\right] \end{array}$$

5.3 An example where both methods are not applicable.

Consider $0 = f_1 = x_1' x_2' - 2\cos^2 t$ and $0 = f_2 = (x_1' x_2')^2 + x_1 + x_2 - 4\cos^4 t - 3\sin t - 2$, with initial values $x_1(0) = x_1'(0) = 2$, $x_2(0) = 0$, $x_2'(0) = 1$. The solution is $x_1(t) = 2\sin t + 2,\ x_2(t) = \sin t$. The SA gives $\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ with $\mathbf{c} = [0,0],\ \mathbf{d} = [1,1]$ and singular $\mathbf{J} = \begin{bmatrix} x_2' & x_1' \\ 2x_1'(x_2')^2 & 2x_2'(x_1')^2 \end{bmatrix}$. A straightforward fix of this failure is to introduce x_3 and replace $x_1' x_2'$ by it, resulting $\mathrm{Val}(\overline{\Sigma}) = 1 < 2 = \mathrm{Val}(\Sigma)$ and $\det(\overline{\mathbf{J}}) = x_1' - x_2' \neq 0$.

However, neither of our conversion methods is applicable. In the LC method, we compute $\mathbf{u} = [2x_1' x_2', 1]^T \in \mathrm{coker}(\mathbf{J})$ and find $I = \{1,2\}$, $\underline{c} = 0$, and $L = \{1,2\}$. Since x_1' and x_2' occur in $\mathbf{u}$, the LC condition (4.3) is violated. Similarly, in the ES method we compute $\mathbf{v} = [x_1', x_2']^T \in \ker(\mathbf{J})$ and find $J = \{1,2\}$, $s = 2$, $M = \{1,2\}$, and $\overline{c} = 0$, and the first ES condition in (4.12) is violated. The algorithms described above for both methods will return $L = J = \emptyset$. Performing a conversion by either method gives $\mathrm{Val}(\overline{\Sigma}) = \mathrm{Val}(\Sigma) = 2$ and $\det(\overline{\mathbf{J}}) \equiv 0$ still.

The incapability of our methods here is due to a nonlinear operation on the common subexpression $x_1' x_2'$ that is already nonlinear in the derivatives of highest order. This situation is not usual in practice, so should have minimal effect on the applicability and usefulness of our methods.

6 Conclusions and related work.

We proposed two conversion methods aimed at improving the Σ-method, which handles DAEs in the general form (2.1). Our methods convert a DAE with finite $\mathrm{Val}(\Sigma)$ and an identically (but not structurally) singular System Jacobian to another DAE that is more likely to have a nonsingular System Jacobian. A conversion guarantees that both DAEs have (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.

We show in [27] a combination of our conversion methods with block triangularization of DAEs [20]. We use these block conversion methods to improve the efficiency of finding a useful conversion that reduces $\mathrm{Val}(\Sigma)$, and to remedy SA's failures in existing literature. For instance, on the Campbell-Griepentrog robot arm DAE [4] of differentiation index 5, the SA reports structural index 3 and $\mathrm{Val}(\Sigma) = 2$. After applying either block LC or block ES method, we obtain structural index 5 and $\mathrm{Val}(\overline{\Sigma}) = 0$, and the resulting DAEs are globally equivalent to the original formulation. On the transistor amplifier and ring modulator DAEs [12], our block conversion methods give $\mathrm{Val}(\overline{\Sigma}) = 5 < 8 = \mathrm{Val}(\Sigma)$ and $\mathrm{Val}(\overline{\Sigma}) = 10 < 11 = \mathrm{Val}(\Sigma)$, respectively. We refer the reader to the first author's PhD thesis [26] for details.

All of our conversion methods can be implemented in a computer algebra system. The computational cost of a conversion depends on the size of the DAE, its sparsity, and intricacy of the equations. Determining the cost in advance is undecidable in the sense of Richardson [22]. For example, fixing $\mathbf{Mf} = 0$ can be arbitrarily difficult, where $\mathbf{f}=0$ is a solvable DAE and $\mathbf{M}$ is a nonsingular dense matrix of expressions comprising t and any derivatives of the x_j's, typically lower than the d_jth.

Integrating our structural analysis software DAESA [16] with MATLAB's Symbolic Math Toolbox [28], we have built a prototype code that automates the conversion process. We have applied our methods on DAEs on which the Σ-method fails; they are either constructed to be SA-failure cases for our investigations, or borrowed from the existing literature. Our code can successfully fix these solvable DAEs, though incapable of dealing with the case in §5.3. We believe that our assumptions and conditions are reasonable for practical problems, and that these methods can help make the Σ-method more reliable.

Lastly we pose our main conjecture regarding SA's failures. When we successfully fix them by performing symbolic simplifications or using our conversion methods, the value of a signature matrix always decreases. As the third author pointed out in [18], the solvability of a DAE may lie within its inherent nature, not the way it is formulated or analyzed. Hence we conjecture that a DAE formulation friendly to SA should have the right $\mathrm{Val}(\Sigma)$ that can be interpreted as number of degrees of freedom (DOF) of the underlying mathematical problem. However, based on our current knowledge, it appears difficult to show why overestimating DOF can lead to an identically singular System Jacobian.

A Preliminary results and proof of Lemma 4.4.

Let the notation be as at the start of §4.2. We prove a lemma first and then Lemma 4.4.

Lemma A.1 *Let* $r\in J\setminus\{l\}$, $w_1=y_r+(v_r/v_l)\cdot x_l^{(d_l-\bar c)}$, *and*

$$w_2=w_1^{(\bar c-c_i)}=\left(y_r+(v_r/v_l)\cdot x_l^{(d_l-\bar c)}\right)^{(\bar c-c_i)}\,. \tag{A.1}$$

Then

$$\sigma\left(x_j,w_2\right)=\begin{cases}<d_j-c_i & \textit{if } j\in J\setminus\{l\}\\ \le d_j-c_i & \textit{otherwise}\,.\end{cases} \tag{A.2}$$

Proof Obviously $\sigma(x_l,w_1)=d_l-\bar c$ when $j=l\in J$. Now consider the case $j\neq l$. Since x_j can occur only in v_r and v_l in w_1, we have $\sigma\left(x_j,w_1\right)\le\sigma\left(x_j,\mathbf{v}\right)\le d_j-\bar c$.

Noting that $\bar c=\max_{i\in M}c_i$, we have $\bar c-c_i\ge 0$ for all $i\in M$. Then (A.2) results from connecting $\sigma\left(x_j,w_2\right)=\sigma\left(x_j,w_1\right)+(\bar c-c_i)$ with (4.12) and the results in the previous paragraph.

The proof of Lemma 4.4 uses the two assumptions preceding it.

Proof Write $\overline{\Sigma}$ in Figure 4.1 into the following 2×3 block form:

$$\overline{\Sigma}=\left[\begin{array}{c|c|c}\overline{\Sigma}_{11}&\overline{\Sigma}_{12}&\overline{\Sigma}_{13}\\\hline\overline{\Sigma}_{21}&\overline{\Sigma}_{22}&\overline{\Sigma}_{23}\end{array}\right].$$

We aim to verify below the relations between $\overline{\sigma}_{ij}$ and $\widetilde{d}_j-\widetilde{c}_i$ in each block.

(1) $\overline{\Sigma}_{11}$. Consider $j,r\in J\setminus\{l\}$. By (4.10), we substitute w_2 in (A.1) for every $x_r^{(d_r-c_i)}$ in f_i for all $i=1:n$. By (A.2), $\sigma\left(x_j,w_2\right)<d_j-c_i$ for all $i\in M$. So these expression substitutions do not introduce $x_r^{(d_r-c_i)}$ in $\overline{f}_i$, where $r\in J\setminus\{l\}$. Given M in (4.8), we have $d_j-c_i>\sigma_{ij}$ for all $i\notin M$ and $j\in J$. Hence

$$\sigma\left(x_j,\overline{f}_i\right)<d_j-c_i\qquad\text{for } j\in J\setminus\{l\},\ i=1:n\,. \tag{A.3}$$

What remains to show is the case $j=l$. From (4.9), $x_r^{(d_r-\bar{c})}=y_r+(v_r/v_l)\cdot x_l^{(d_l-\bar{c})}$. Taking the partial derivatives of both sides with respect to $x_l^{(d_l-\bar{c})}$ and applying Griewank's Lemma (4.1) with $w=x_r^{(d_r-\bar{c})}$ and $q=\bar{c}-c_i\geq 0$ for all $i\in M$, we have

$$\frac{v_r}{v_l}=\frac{\partial x_r^{(d_r-\bar{c})}}{\partial x_l^{(d_l-\bar{c})}}=\frac{\partial x_r^{(d_r-\bar{c}+\bar{c}-c_i)}}{\partial x_l^{(d_l-\bar{c}+\bar{c}-c_i)}}=\frac{\partial x_r^{(d_r-c_i)}}{\partial x_l^{(d_l-c_i)}}\,. \tag{A.4}$$

Then

$$\begin{aligned}\frac{\partial \overline{f}_i}{\partial x_l^{(d_l-c_i)}}&=\frac{\partial f_i}{\partial x_l^{(d_l-c_i)}}+\sum_{r\in J\setminus\{l\}}\frac{\partial f_i}{\partial x_r^{(d_r-c_i)}}\cdot\frac{\partial x_r^{(d_r-c_i)}}{\partial x_l^{(d_l-c_i)}} && \text{by the chain rule}\\ &=J_{il}+\sum_{r\in J\setminus\{l\}}J_{ir}\cdot\frac{v_r}{v_l}=\frac{1}{v_l}\sum_{r\in J}J_{ir}v_r=\frac{1}{v_l}(\mathbf{J}\mathbf{v})_i=0 && \text{by (A.4) and } \mathbf{J}\mathbf{v}=\mathbf{0}\,.\end{aligned}$$

This gives $\sigma\left(x_l,\overline{f}_i\right)<d_l-c_i$ for all $i=1:n$. Together with (A.3) we have proved the "$<$" part in $\overline{\Sigma}_{11}$.

(2) $\overline{\Sigma}_{12}$. The substitutions do not affect x_j, for all $j\notin L$. By (A.2), such an x_j occurs in every w_2 of order $\leq d_j-c_i$, where $i\in M$. Hence also $\sigma\left(x_j,\overline{f}_i\right)\leq d_j-c_i$ for all $i=1:n$ and $j\notin L$.

(3) $\overline{\Sigma}_{13}$. Consider $r\in J\setminus\{\,l\,\}$. For an $i\in M$, y_r occurs of order $\bar{c}-c_i$ in w_2 in (A.1). For all $i=1:n$, if a substitution occurs for an $x_r^{(d_r-c_i)}$ in f_i, then $\sigma\left(y_r,\overline{f}_i\right)=\bar{c}-c_i$; otherwise $\sigma\left(y_r,\overline{f}_i\right)=-\infty$. In either case $\sigma\left(y_r,\overline{f}_i\right)\leq\bar{c}-c_i$.

(4) $\overline{\Sigma}_{21}$. Equalities hold on the diagonal and in the lth column, as $y_r^{(d_r-\bar{c})}$ and $y_l^{(d_l-\bar{c})}$ occur in g_l, where $r\in J$. What remains to show is the "$<$" part. Assume that $j,r,l\in J$ are distinct. Then by (4.9) and (4.12),

$$\sigma\left(x_j,g_r\right)=\sigma\left(x_j,y_r-x_r^{(d_r-\bar{c})}+\frac{v_r}{v_l}\cdot x_l^{(d_l-\bar{c})}\right)\leq\sigma\left(x_j,\mathbf{v}\right)<d_j-\bar{c}\,. \tag{A.5}$$

(5) $\overline{\Sigma}_{22}$. Assume again that j,r,l are distinct, where $r\in J$ and $j=s+1:n$. Then replacing the "$<$" in (A.5) by "$\leq$" proves the "$\leq$" part in $\overline{\Sigma}_{22}$.

(6) $\overline{\Sigma}_{23}$. Consider $r,j\in J$. By $0=g_l=-y_l+x_l^{(d_l-\bar{c})}$ and (4.9), y_j occurs in g_r only if $j=r$, and $\sigma\left(y_j,g_j\right)=0$. Hence, on the diagonal lie zeros, and everywhere else is filled with $-\infty$.

Also worth noting is that in the y_l column is only one finite entry $\sigma_{n+l,n+l}=0$, and that in the g_l row are only two finite entries $\sigma_{n+l,n+l}=0$ and $\sigma_{n+l,l}=d_l-\bar{c}$.

Recalling (4.13) for the formulas of $\widetilde{c}_i$ and $\widetilde{d}_j$ of $\overline{\Sigma}$, we can summarize that the above items (1)–(6) verify the relations between $\overline{\sigma}_{ij}$ and $\widetilde{d}_j-\widetilde{c}_i$ in $\overline{\Sigma}$ for all $i,j=1:n+s$; see Figure 4.1. □

Acknowledgements The authors acknowledge with thanks the financial support for this research: GT was supported in part by the McMaster Centre for Software Certification through the Ontario Research Fund, Canada, NSN was supported in part by the Natural Sciences and Engineering Research Council of Canada, and JDP was supported in part by the Leverhulme Trust, the UK. The authors thank the anonymous reviewers for providing valuable suggestions on improving this article.

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.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, second edn. SIAM, Philadelphia (1996)
4. Campbell, S.L., Griepentrog, E.: Solvability of general differential-algebraic equations. SIAM Journal on Scientific Computing **16**(2), 257–270 (1995)

5. Carpanzano, E., Maffezzoni, C.: Symbolic manipulation techniques for model simplification in object-oriented modelling of large scale continuous systems. Mathematics and Computers in Simulation **48**(2), 133 – 150 (1998)
6. Chowdhry, S., Krendl, H., Linninger, A.A.: Symbolic numeric index analysis algorithm for differential-algebraic equations. Industrial & engineering chemistry research **43**(14), 3886–3894 (2004)
7. Griewank, A.: On automatic differentiation. In: M. Iri, K. Tanabe (eds.) Mathematical Programming: Recent Developments and Applications, pp. 83–108. Kluwer Academic Publishers, Dordrecht (1989)
8. Kunkel, P., Mehrmann, V.: Index reduction for differential-algebraic equations by minimal extension. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **84**(9), 579–597 (2004)
9. Kunkel, P., Mehrmann, V.L.: Differential-algebraic equations: analysis and numerical solution. European Mathematical Society, Zürich, Switzerland (2006)
10. MapleSim: Technological superiority in multi-domain physical modeling and simulation (2012). http://www.maplesoft.com/view.aspx?sf=7032
11. Mattsson, S.E., Söderlind, G.: Index reduction in differential-algebraic equations using dummy derivatives. SIAM J. Sci. Comput. **14**(3), 677–692 (1993)
12. Mazzia, F., Iavernaro, F.: Test set for initial value problem solvers. Tech. Rep. 40, Department of Mathematics, University of Bari, Italy (2003). http://pitagora.dm.uniba.it/~testset/
13. Nedialkov, N., Pryce, J.D.: DAETS user guide. Tech. Rep. CAS 08-08-NN, Department of Computing and Software, McMaster University, Hamilton, ON, Canada (2013). 68 pages, DAETS is available at http://www.cas.mcmaster.ca/~nedialk/daets
14. Nedialkov, N.S., Pryce, J.D.: Solving differential-algebraic equations by Taylor series (I): Computing Taylor coefficients. BIT Numerical Mathematics **45**(3), 561–591 (2005)
15. Nedialkov, N.S., Pryce, J.D.: Solving differential-algebraic equations by Taylor series (III): the DAETS code. JNAIAM J. Numer. Anal. Indust. Appl. Math **3**, 61–80 (2008)
16. 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)
17. Pantelides, C.C.: The consistent initialization of differential-algebraic systems. SIAM J. Sci. Stat. Comput. **9**, 213–231 (1988)
18. Pryce, J.D.: Solving high-index DAEs by Taylor Series. Numerical Algorithms **19**, 195–211 (1998)
19. Pryce, J.D.: A simple structural analysis method for DAEs. BIT Numerical Mathematics **41**(2), 364–394 (2001)
20. 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)
21. Reissig, G., Martinson, W.S., Barton, P.I.: Differential–algebraic equations of index 1 may have an arbitrarily high structural index. SIAM J. Sci. Comput. **21**(6), 1987–1990 (1999)
22. Richardson, D.: Some undecidable problems involving elementary functions of a real variable. J. Symbolic Logic **33**(4), 514–520 (1968)
23. Scholz, L., Steinbrecher, A.: Regularization of DAEs based on the signature method. BIT Numerical Mathematics **56**(1), 319–340 (2016)
24. Scholz, L., Steinbrecher, A.: Structural-algebraic regularization for coupled systems of DAEs. BIT Numerical Mathematics **56**(2), 777–804 (2016)
25. Sjölund, M., Fritzson, P.: Debugging symbolic transformations in equation systems. In: Proceedings of Equation-based Object-Oriented Modeling Languages and Tools (EOOLT), pp. 67–74 (2011)
26. Tan, G.: Conversion methods for improving structural analysis of differential-algebraic equation systems. Ph.D. thesis, School of Computational Science and Engineering, McMaster University, 1280 Main St W, Hamilton, Ontario, L8S 1A8, Canada (2016). 168 pages
27. Tan, G., Nedialkov, N.S., Pryce, J.D.: Conversion methods, block triangularization, and structural analysis of differential-algebraic equation systems (2016). Tech. Rep. CAS 16-04-NN, Department of Computing and Software, McMaster University, Hamilton, ON, Canada (2016). 26 pages, http://www.cas.mcmaster.ca/cas/0reports/CAS-16-04-NN.pdf
28. The MathWorks, Inc.: Matlab Symbolic Math Toolbox (2016). http://www.mathworks.com/products/symbolic/