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

CONVERSION METHODS FOR IMPROVING STRUCTURAL ANALYSIS OF DIFFERENTIAL-ALGEBRAIC EQUATION SYSTEMS

BY

GUANGNING TAN

A THESIS

SUBMITTED TO THE SCHOOL OF COMPUTATIONAL SCIENCE AND ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

PHD OF COMPUTATIONAL SCIENCE AND ENGINEERING

PhD of Computational Science and Engineering (2016) McMaster University
(School of Computational Science and Engineering) Hamilton, Ontario, Canada

TITLE:	Conversion Methods for Improving Structural Analysis of Differential-Algebraic Equation Systems
AUTHOR:	Guangning Tan School of Computational Science and Engineering McMaster University, Hamilton, Ontario, Canada
SUPERVISOR:	Nedialko S. Nedialkov
NUMBER OF PAGES:	viii, 170

Abstract

Systems of differential-algebraic equations (DAEs) arise in many areas including chemical engineering, electrical circuit simulation, and robotics. Such systems are routinely generated by simulation and modeling environments, like MapleSim, Matlab/Simulink, and those based on the Modelica language. Before a simulation starts and a numerical solution method is applied, some kind of structural analysis (SA) is performed to determine the structure and the index of a DAE system.

Structural analysis methods serve as a necessary preprocessing stage, and among them, Pantelides's graph-theory-based algorithm is widely used in industry. Recently, Pryce's Σ-method is becoming increasingly popular, owing to its straightforward approach and capability of analyzing high-order systems. Both methods are equivalent in the sense that (a) when one succeeds, producing a nonsingular Jacobian, the other also succeeds, and that (b) the two give the same structural index in the case of either success or failure. When SA succeeds, the structural results can be used to perform an index reduction process, or to devise a stage-by-stage solution scheme for computing derivatives or Taylor coefficients up to some order.

Although such a success occurs on fairly many problems of interest, SA can fail on some simple, solvable DAEs with an identically singular Jacobian, and give incorrect structural information that usually includes the index. In this thesis, we focus on the

Σ-method and investigate its failures. Aiming at making this SA more reliable, we develop two conversion methods for fixing SA failures. These methods reformulate a DAE on which the Σ-method fails into an equivalent problem on which SA is more likely to succeed with a nonsingular Jacobian. The implementation of our methods requires symbolic computations.

We also combine our conversion methods with block triangularization of a DAE. Using a block triangular form of a Jacobian sparsity pattern, we identify which diagonal block(s) of the Jacobian is identically singular, and then perform a conversion on each singular block. This approach can reduce the computational cost and improve the efficiency of finding a suitable conversion for fixing SA's failures.

Acknowledgements

I thank my supervisor Professor Nedialko (Ned) Stoyanov Nedialkov, who has been encouraging me to pursue this research topic that I found interesting. Without such freedom, I could not have fully dedicated myself to this topic and further developed it into a PhD thesis. He offered me invaluable suggestions and resources to improve my technical writing and presentation skills. He spent incredibly many hours and efforts on reading and commenting on my writing and slides, and instructed me how to make a work solid, concise, and understandable to many. He shared with me his experience in computer science and scientific programming and computing. During my hard times, he kept encouraging me to stay strong and move on.

I also thank Professor John Derwent Pryce. He had great insights in my research topics, and also gave helpful advice and suggestions on my writing, especially on my proofs. His mathematical way of thinking and his MATLAB code inspired me much. I have learned so much from his technical writing, and will never have learned enough from it.

I wish to thank my parents (Yitong Tan, Xiaowei Zhou), two of my grandmothers (Binping Zhou, Cuilian Wu), and my godparents (Shijin Li, Lun Wu). Without *you*, I could not have come to Canada and pursued my graduate studies abroad, not to mention making a PhD. All of you have given me unconditional love and sacrifice

to make this Doctorate degree happen, and I owe this success to you. I also thank my other family members for the help during all these years.

I wish to thank my fiancée Zhaocheng (Elly) Zeng, her parents (Wencai Zeng, Huizhen Cheng), and her other family members, in particular her uncle Colin Chunhui Cheng, aunt-in-law Jane Weijun Feng, and her grandparents, for continuous support for my life, studies, and work.

I thank my supervisory committee members for their continuous advice, help, and support. They are Professors Christopher Swartz (Chemical Engineering), Timothy Davidson (Electrical & Computer Engineering), Sanzheng Qiao (Computing & Software), and Nicholas Kevlahan (Mathematics & Statistics). I thank Professor Alan Wassyng for volunteering to chair my PhD defense. I thank Professor Robert Corless (Department of Applied Mathematics, Western University) for being my external examiner, recognizing my PhD research, and generously sharing with me his profound knowledge about computer algebra. I enjoyed very much discussing with him and his students. Finally I express my gratitude to Professor Bartosz Protas (Mathematics & Statistics), who has been advising me during my stay in the CSE programs.

I am grateful for the financial support by the McMaster Centre for Software Certification, who supported large part of my studies, NSERC, the McMaster International Excellence Scholarship, and the Dalley Fellowship (McMaster).

Finally, I thank my colleagues and friends Ross McKenzie (Applied Mathematics, Cardiff University), Ian Washington (Chemical Engineering), John Ernsthausen and Xiao (Shawn) Li (Computational Science & Engineering). We conduct research together, and I learn so much from your expertise and experience.

Contents

Chapter 1

Introduction

Differential-algebraic equation systems (DAEs) are generated routinely by modeling and simulation environments, such as MapleSim [25], Matlab/Simulink [55, 57], SimulationX [11], and those built on the Modelica language [20, 39, 58]. These DAE systems arise from disciplines such as electrical circuits, chemical engineering, and rigid body mechanical systems.

To simulate the dynamic behaviour of a DAE system, a variety of algorithms are applied in the steps from creating a mathematical model to constructing a numerically solvable system of equations. In the modeling process, dynamical systems are generated by selecting components in different libraries and integrating these components into subsystems. Such a subsystem can have its own physical dynamics, and all these subsystems together can be further interconnected to each other via interface or coupling formulas. This approach of modeling can result in a large, sparse, and nonlinear DAE system, which is typically structured: the coupling between components is stronger within a subsystem, but is weaker between subsystems. Moreover, such a DAE can be of high index.

Understanding the solution process of a DAE is nontrivial. To solve numerically a DAE, typically derivatives of some of its equations need to be appended to the original formulation, forming an augmented overdetermined system. We then wish to reduce this enlarged system to an implicit ODE or index-1 DAE system, so that a standard numerical code can be used for integration. However, in general, it is not easy to find which equations are to be differentiated and, especially, how many times. If the numerical method is not chosen properly for a high-index DAE, then the integration can lead to instabilities and non-convergence of the method [3,17,22].

Hence, before a numerical solution method is applied to a DAE, some kind of structural analysis (SA) algorithm is applied to determine some characteristics of the DAE, such as index, number of degrees of freedom, and variables and derivatives that need consistent initial values. These structural analysis methods serve as a preprocessing stage, providing more insights into the underlying structure of a DAE and indicating which numerical solution method can be applied.

1.1 Overview of DAEs

Throughout this thesis, we discuss an initial value problem DAE of the general form[1]:

$$f_i(\, t, \text{ the } x_j \text{ and derivatives of them }) = 0, \quad i = 1:n, \tag{1.1}$$

where the $x_j(t)$, $j = 1:n$, are n state variables, and t is the independent variable, usually regarded as time. The formulation (1.1) includes high-order systems and systems that are jointly nonlinear in leading derivatives. Furthermore, (1.1) includes

[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.

ODEs and purely algebraic systems.

An important characteristic of a DAE is its *index*. Various definitions of index exist in the literature: differentiation index [4, 13, 14], geometric index [48, 50], structural index [10, 40, 42], perturbation index [17], tractability index [15], and strangeness index [22]. Among these definitions, the differentiation index is the most popular one; see its definition below. Generally speaking, the index measures how different a DAE is from an ODE, and how difficult it is to solve a DAE.

We let $\mathbf{x}(t)$ denote a vector of functions $x_1(t), x_2(t), \ldots, x_n(t)$. The following definition for differentiation index is from [1, p. 236].

Definition 1.1 ***(Differentiation index)*** *Consider a general form of a first-order DAE*

$$\mathbf{F}(t, \mathbf{x}, \mathbf{x}') = \mathbf{0}, \tag{1.2}$$

where $\partial\mathbf{F}/\partial\mathbf{x}'$ may be singular. The differentiation index (written also d-index or ν_d) along a solution $\mathbf{x}(t)$ is the minimum number of differentiations of the system that would be required to solve $\mathbf{x}'$ uniquely in terms of $\mathbf{x}$ and t, that is, to define an ODE for $\mathbf{x}$. Thus this index is defined in terms of the overdetermined system

$$\begin{aligned} \mathbf{F}\left(t, \mathbf{x}, \mathbf{x}'\right) &= \mathbf{0}, \\ \frac{\mathrm{d}\mathbf{F}}{\mathrm{d}t}\left(t, \mathbf{x}, \mathbf{x}', \mathbf{x}''\right) &= \mathbf{0}, \\ &\vdots \\ \frac{\mathrm{d}^p\mathbf{F}}{\mathrm{d}t^p}\left(t, \mathbf{x}, \mathbf{x}', \cdots, \mathbf{x}^{(p+1)}\right) &= \mathbf{0} \end{aligned} \tag{1.3}$$

to be the smallest integer p so that $\mathbf{x}'$ in (1.2) can be solved for in terms of $\mathbf{x}$ and t.

If a DAE (1.1) is of high-order, then one can introduce additional variables to reduce it to first-order and fit into the form (1.2).

We give a definition for a solution of a DAE.

Definition 1.2 ***(Solution of a DAE)*** *[42] An n-vector function $\mathbf{x}(t)$, defined over a time interval $\mathbb{I} \subset \mathbb{R}$, is a solution of (1.1), if $(t, \mathbf{x}(t))$ satisfies $f_i = 0$, $i = 1:n$, pointwise for all $t \in \mathbb{I}$—that is, functions f_i vanish on $\mathbb{I}$.*

Remark 1.3 If $\mathbf{x}(t)$ is a solution of (1.1) and is sufficiently differentiable, then functions f_i and derivatives of them vanish for all $t \in \mathbb{I}$. That is,

$$0 = f_i^{(m)} \quad \text{for all } i = 1:n \text{ and } m \geq 0.$$

Since the general form (1.1) we deal with has 0 on the right-hand side, by "an equation f_i" we shall mean its corresponding equation $f_i = 0$, omitting the verbose "$= 0$" part.

If a DAE is of index-1, then we say it is of *low index*. To solve this DAE, a standard index-1 DAE solver can be used, for example, DASSL [3], IDA of SUNDIALS [18], or MATLAB's `ode15s` and `ode23t`. If a DAE is of index ≥ 2, then the DAE is of *high index* and we need a high-index DAE solver, such as RADAU5 for DAEs of index ≤ 3 [17], or DAETS for DAEs of any index [36]. Index reduction methods [21, 26] and regularization techniques [22, 52] exist, and can be used to reduce a high-index DAE to a DAE of index-1 or a regularized (and thus regular) DAE, respectively. Recent works by Pryce and McKenzie focus on the dummy derivatives (DDs) index reduction method [28, 29, 31, 32, 43, 44].

1.2 Overview of structural analysis methods

To compute the differentiation index from its formal definition in Definition 1.1, we may use linear algebra to investigate how the first-order derivatives $\mathbf{x}'$ can be determined by t and $\mathbf{x}$. That is, we construct a system of some or all equations in (1.3) in t, $\mathbf{x}$, $\mathbf{x}'$, and their higher derivatives. Then we ensure that the Jacobian matrix of these equations with respect to the relevant variables/derivatives is nonsingular, so that $\mathbf{x}'$ can be computed. See the derivative array equations approach in [5] and the tractability index approach in [23], which also use linear algebra.

For DAEs of large size and/or high index, the size of the matrix to be analyzed can be much larger than the original problem size n, so checking its matrix nonsingularity can involve heavy linear algebra [45,47]. Therefore, we wish to find the index of a DAE by its sparsity-based structural information, namely which variables and derivatives of them occur in each equation. *Structural analysis* (SA) methods are designed to do this job, so serve as a preprocessing stage to determine the index before a numerical solution method is applied. Among them is the Pantelides's method [40], a graph-based algorithm that finds how many times each equation needs to be differentiated. Pryce's SA, the **SIG**nature **MA**trix method or the Σ-*method*, involves a combinatorial optimization problem from discrete mathematics and optimization theory. The Σ-method is equivalent to Pantelides's algorithm, and both algorithms always compute the same *structural index* [42]. Pantelides's algorithm can handle first-order systems (1.2) only, while the Σ-method can be applied to (1.1) of any order. This allows one to avoid writing less interesting equations like $y' = z$, and hence to formulate many problems in a more compact and concise way; see Example 6.3 for instance. We believe that the Σ-method is more direct and easier to apply to some extent.

Throughout this thesis, we shall use the Σ-method and refer to it as *our SA*. Other SA methods can be found in, for example, [54] and [59].

The Σ-method determines structural index, which is often the same as the differentiation index, the number of degrees of freedom (DOF), the variables and derivatives that need to be initialized, the constraints of the DAE, and a solution scheme for a Taylor series method. We give the definition for the structural index in §2.

Nedialkov and Pryce have developed DAETS (solving **DAE**s by **T**aylor **S**eries), a C++ package that integrates an initial value problem (IVP) for a DAE system of arbitrarily high index and order using a Taylor series method [34, 35, 36]. DAETS uses the Σ-method to analyze a DAE of the form (1.1), and the SA result prescribes a stage by stage solution scheme. This scheme indicates at each stage which equations need to be solved and for which Taylor coefficients (TCs) [resp. derivatives] for the solution; see §2.1 and more details in [38]. DAETS computes these TCs up to some order and then performs an integration step.

Tan, Nedialkov, and Pryce have developed DAESA, **DAE**s **S**tructural **A**nalyzer, a MATLAB package that performs SA of DAEs [37, 46]. It incorporates recent development of our SA theory, and contains more sophisticated SA features compared to DAETS. DAESA allows convenient translation of a DAE into MATLAB and provides a set of easy-to-use functions (17 functions in Version 1.0) for determining SA results. It also allows a rapid investigation of the structure of a DAE by visualizing its block triangular forms (BTFs). DAESA performs quasilinearity analysis (QLA) on each diagonal block, and derives the minimum set of initial values [46] and a blockwise solution scheme [38]. For usages of this package we refer the reader to the DAESA user guide [30].

In [2, 16, 19], the Σ-method is also applied to perform SA of DAEs, and the SA results are used for numerical simulations. In [8], the Σ-method is used to prove that the computational complexity of solving the initial value problem for a DAE of arbitrarily high index is polynomial in the number of bits of accuracy needed. The Σ-method also guides one to carry out a DDs-based index reduction procedure and to design software code for automating this procedure [28, 29, 31, 32, 43, 44].

1.3 Limitations of structural analysis methods

Although the Σ-method provably gives correct structural information (including index) on many DAEs of practical interest [42], it can fail—whence Pantelides's algorithm fails as well—on simple, solvable DAEs, producing an identically singular System Jacobian. (See §2 for the definition of System Jacobian and that of SA's success.) We shall refer to these solvable DAEs as *SA-failure cases* or *SA-unfriendly DAEs*. The DAEs on which SA succeeds are said to be *SA-friendly*.

In [42], Pryce shows that, if the Σ-method succeeds, then the structural index ν_S is always an upper bound on the differentiation index ν_d. This implies that, if the structural index is smaller than ν_d, then the Σ-method *must* fail; otherwise we would have a counter statement to the above Definition 1.1.

The simplest way SA can fail is by hidden symbolic cancellations (see definition in §3.2). These cancellations can cause more structural zeros in the System Jacobian, making it more likely to be structurally singular; see later discussions in §3.2. However, SA can fail in a subtle and obscure way. In this case, it was difficult to understand the causes of such failures and to give systematic ways for fixing these problem

formulations—the last paragraph in [36, §7] and [33, §5.2.3] admit this difficulty. Although such deficiencies occur rarely, they make SA less reliable and hence can pose limitations to its applications. As SA-based methods are applied more widely, SA's failures are likely to become more common and hence should be carefully dealt with.

The following works investigate SA's failures and attempts to tackle them.

Pryce shows in [41] that the Σ-method fails on the index-5 Campbell-Griepentrog Robot Arm DAE—the SA produces an identically singular Jacobian; our Example 6.3 shall discuss this. He then provides a remedy: identify the common subexpressions in the DAE, introduce four extra variables, and substitute them for those subexpressions. The resulting equivalent problem is an enlarged one, on which the Σ-method succeeds and reports the correct structural index 5. Pryce introduces the term *structure-revealing* to conjecture that a nonsingular System Jacobian might be an effect of DAE formulation, but not of DAE's inherent nature.

Chowdhry *et al.* propose a method called symbolic numeric index analysis (SNIA) [7]. Their method can accurately detect symbolic cancellation of variables that appear linearly in equations, and therefore can deal with linear constant coefficient systems. For general nonlinear DAEs, SNIA is claimed to provide a correct result in some cases, but not all. Furthermore, it is limited to first-order systems, and cannot handle complex expression substitution and symbolic cancellations, such as $(x \cos y)' - x' \cos y$. For the general case, their method does not derive from the original problem an equivalent one that has the correct index.

Scholz and Steinbrecher are interested in a class of DAEs called coupled systems [52]. A coupled system in [52] is composed by coupling two semi-explicit systems of differentiation index 1, and the resulting system can be of high index. They assert

that the Σ-method succeeds if and only if the coupled system is again of differentiation index 1, and fails if the coupled system is of high index. They show that several simulation environments such as Dymola, OpenModelica, SimulationX, and MapleSim all fail on a simple, solvable linear constant coefficient DAE; we shall discuss this in Example 3.18. They propose a structural-algebraic method to deal with such SA failures occurring in coupled systems. Their method differentiates a linear combination of certain algebraic equations that contribute to singularity, appends the resulting equations, and replaces certain derivatives with newly introduced variables. They use their regularization process to convert the regular coupled system to a DAE of index 1, on which SA succeeds with a nonsingular System Jacobian.

Other SA-failure cases include the transistor amplifier and the ring modulator [27]. We shall discuss them in §7.1 and §7.2, respectively. In this thesis, we shall construct more SA-unfriendly DAEs and show how to convert them to SA-friendly ones.

Another limitation of SA is the index overestimation problem: when SA succeeds and produces a nonsingular System Jacobian, the structural index may overestimate the differentiation index. Reißig *et al.* construct a class of linear constant coefficient DAEs of differentiation index 1, and claim that the structural index of such DAEs increases with the problem size and hence can be arbitrarily high [49]. On these DAEs, Pantelides's algorithm performs a high number of iterations and differentiations, and obtains a high structural index exceeding 1. Pryce shows in [42] the application of the Σ-method on one such DAE of size 5. Producing a nonsingular System Jacobian, this SA succeeds, but still reports the same high structural index 3 as does Pantelides's. This situation is not favoured, since the difficulty of numerically solving each such DAE is exaggerated. We also attempt to tackle this problem in §7.3.

1.4 Contributions

This thesis focuses on handling SA-unfriendly DAEs. When SA fails on a DAE and produces an identically singular System Jacobian, our goal is to convert automatically this DAE into an equivalent problem on which SA succeeds or (at least) it is more likely to succed. This thesis is devoted to developing such methods—*the conversion methods*.

The main contributions of this thesis are as follows.

- We develop two conversion methods that reformulate an SA-unfriendly DAE to an equivalent SA-friendly problem formulation. Using a symbolic tool, we can perform the conversions in a systematic way. We identify the equivalence of the original DAE and the converted one, and ensure that both have the same solution (if any). We also give rationale for choosing the most suitable conversion among possibly several ones. See §4.

- We combine block triangularization of a DAE with our conversion methods. The block conversion methods can improve the efficiency of finding a useful conversion for fixing SA's failures. See §6.

- We give insight into SA's failures, which were not well understood before. We point out that the reason behind such failures is related to an overestimation of the number of degrees of freedom of a DAE.

- We show how to fix the SA-unfriendly DAEs in the existing literature by our conversion methods. See §5, §7.1, and §7.2. We also show how to resolve the index overestimation problem on the family of DAEs by Reißig. See §7.3.

1.5 Thesis organization

The rest of this thesis is organized as follows.

Chapter 2 summarizes the Σ-method and gives definitions and tools that are needed for our theoretical development. We give a definition for SA's success and failure on a DAE, and show how to derive block triangular forms (BTFs) of a DAE.

Chapter 3 describes the problem of SA's failures on some DAEs. We show how to distinguish two types of SA's failure: in one type, SA produces a System Jacobian that is structurally singular; in the other case, the System Jacobian is structurally nonsingular but is still identically singular.

Chapter 4 presents the basic version of the conversion methods, the *linear combination (LC) method* and the *expression substitution (ES) method*. By "basic" we mean that these methods do not exploit a BTF of a DAE. We derive conditions under which we can convert an SA-unfriendly DAE to an equivalent SA-friendly DAE. The equivalence of DAEs is also discussed.

Chapter 5 illustrates the basic conversion methods with two more examples.

Chapter 6 shows how to combine the conversion methods with a block triangularization of a DAE. Using a BTF based on a Jacobian sparsity pattern, we can identify which blocks of a Jacobian are identically singular, and then apply the conversion methods on each such block.

Chapter 7 illustrates these block conversion methods with two DAEs from electrical circuit analysis, and shows our treatment for the index overestimation problem on the family of DAEs by Reißig.

Chapter 8 gives concluding remarks.

Chapter 2

Summary of the Σ-method

In this chapter, we review the Σ-method that performs structural analysis (SA) of a DAE. We present in §2.1 this method and the notation we use, and describe in §2.2 the block triangularization of DAEs. Terms at their defining occurrence are in *slanted* font.

Throughout this thesis, we assume that the functions f_i in (1.1) are sufficiently differentiable.

2.1 A simple structural analysis

We call this SA method [42] the Σ-method, because it constructs for (1.1) an $n \times n$ *signature matrix* $\mathbf{\Sigma} = (\sigma_{ij})$ such that each *signature entry*

$$\sigma_{ij} = \begin{cases} \text{highest order of derivative to which } x_j \text{ occurs in } f_i\text{; or} \\ -\infty \qquad \text{if } x_j \text{ does not occur in } f_i. \end{cases} \tag{2.1}$$

A *transversal* T of $\mathbf{\Sigma}$ is a set of n positions (i, j) with exactly one position in each row and each column. The sum of signature entries over T, or $\sum_{(i,j)\in T} \sigma_{ij}$, is called the *value of* T, written $\mathrm{Val}(T)$. We seek a *highest-value transversal* (HVT) that gives this sum the largest value. We call the maximum sum the *value of the signature matrix*, written $\mathrm{Val}(\mathbf{\Sigma})$.

We give a definition for a DAE's structural posedness.

Definition 2.1 ***(Structural well-posedness of a DAE)*** *A DAE is structurally well posed (SWP) if there is some one-to-one correspondence between equations and variables, or equivalently* $\mathrm{Val}(\mathbf{\Sigma}) > -\infty$*; otherwise, the DAE is structurally ill posed (SIP) and* $\mathrm{Val}(\mathbf{\Sigma}) = -\infty$*.*

In the SWP case, we have some transversal T on which all signature entries σ_{ij} are finite. Such a transversal is said to be *finite*. Since $\mathrm{Val}(\mathbf{\Sigma})$ is the maximum value of $\mathrm{Val}(T)$, $\mathrm{Val}(\mathbf{\Sigma})$ is also finite. In contrast, in the SIP case, there is no finite transversal, so $\mathrm{Val}(\mathbf{\Sigma}) = -\infty$.

We henceforth consider the SWP case. Using an HVT, we find $2n$ integers

$$\mathbf{c} = (c_1, \ldots, c_n) \quad \text{and} \quad \mathbf{d} = (d_1, \ldots, d_n)$$

associated with the equations and variables of (1.1), respectively. These integers satisfy

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

We refer to such $\mathbf{c}$ and $\mathbf{d}$, written as a pair $(\mathbf{c}; \mathbf{d})$, as a *valid offset pair*. A valid offset pair 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* [42].

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

$$k = k_d, k_d + 1, \ldots, 0, 1, \ldots, \quad \text{where } k_d = -\max_j d_j. \tag{2.3}$$

At each stage k, we solve a system

$$0 = f_i^{(c_i+k)} \qquad \text{for all } i \text{ such that } c_i + k \geq 0 \tag{2.4}$$

for derivatives

$$x_j^{(d_j+k)} \qquad \text{for all } j \text{ such that } d_j + k \geq 0, \tag{2.5}$$

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

The systems at stages $k \geq 0$ are always square, since $c_i + k \geq 0$ and $d_j + k \geq 0$ for all $i, j = 1\!:\!n$. For $k = k_d, \ldots, -1$, there are usually more derivatives in (2.5) than equations in (2.4)—that is, the stage k subsystem is *underdetermined*. In this case, we provide trial values $\widetilde{x}_j^{(d_j+k)}$, and solve the equations $f_i^{(c_i+k)}$ as a least-square problem for the derivatives $x_j^{(d_j+k)}$. Also, if the stage k system is well determined (meaning square) and the derivatives $x_j^{(d_j+k)}$ occur in a jointly nonlinear way, then we also need trial values and find a least-square solution for these derivatives. We refer the reader to [38] for a detailed discussion of the solution scheme.

We give a definition for a success of our SA.

Definition 2.2 ***(Success of the Σ-method)*** *If the solution scheme (2.3–2.5) 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 they fail, in the sense that the Jacobian used to solve (2.4) at some stage $k \in k_d : 0$ does not have full row rank.*

The Jacobian used to solve (2.4) for stages $k \ge 0$ is called the *System Jacobian* of (1.1), an $n \times n$ matrix $\mathbf{J}(\mathbf{c};\mathbf{d}) = (J_{ij})$ defined by

$$J_{ij} = \frac{\partial f_i^{(c_i)}}{\partial x_j^{(d_j)}} = \frac{\partial f_i}{\partial x_j^{(d_j - c_i)}} = \begin{cases} \frac{\partial f_i}{\partial x_j^{(\sigma_{ij})}} & \text{if } d_j - c_i = \sigma_{ij}, \text{ and} \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

with $i, j = 1:n$. The second "=" in (2.6) is based on Griewank's Lemma below, and the third "=" follows from (2.2).

Lemma 2.3 ***(Griewank's Lemma)*** *[42] 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 $p \ge 0$. If $\sigma(x_j, w) \le q$, then*

$$\frac{\partial w}{\partial x_j^{(q)}} = \frac{\partial w'}{\partial x_j^{(q+1)}} = \cdots = \frac{\partial w^{(p)}}{\partial x_j^{(q+p)}}. \tag{2.7}$$

Using the derivatives computed in stages $k = k_d : 0$, we have found a *consistent point*: it is either $\left(t, x_1^{(<d_1)}, \ldots, x_n^{(<d_n)}\right)$, if every $x_j^{(d_j)}$ occurs in a *jointly linear* way in every $f_i^{(c_i)}$, or $\left(t, x_1^{(\le d_1)}, \ldots, x_n^{(\le d_n)}\right)$, if some $x_j^{(d_j)}$ occurs in a *jointly nonlinear* way in every $f_i^{(c_i)}$, equations from the stage $k = 0$ subsystem. We refer to [34, 46] for a more rigorous discussion of a consistent point.

As noted earlier, we assume that the equations in (1.1) are sufficiently differentiable and that derivatives of a solution $\mathbf{x}(t)$ to (1.1) exist to some order. Theorem 4.2 in [42] proves existence of a DAE (1.1), and Section 3 in [34] extends this existence result to a needed smoothness result: if $\mathbf{J}$ is nonsingular at a consistent point of $t = t^*$ and each function f_i has $(N + c_i)$ continuous derivatives in a neighbourhood of this consistent point, for some integer $N \geq 1$, then each of $x_j(t)$ has $(N + d_j)$ continuous derivatives in a neighbourhood of t^*, and the solution scheme (2.3–2.5) can compute these derivatives up to stage $k = N$.

Although a different offset pair $(\mathbf{c}; \mathbf{d})$ produces a different solution scheme (2.3–2.5) and generally a different System Jacobian $\mathbf{J}(\mathbf{c}; \mathbf{d})$, all $\mathbf{J}$'s nevertheless share the same determinant [34]. If one $\mathbf{J}$ is nonsingular—whence so are all $\mathbf{J}$'s—at a consistent point, then SA succeeds, and there exists (locally) a unique solution through this point [42]. Then we use the canonical offset pair $(\mathbf{c}; \mathbf{d})$ to determine the *structural index* and the number of *DOF* [42]:

$$\nu_S = \max_i c_i + \begin{cases} 1 & \text{if } \min_j d_j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

$$\text{and} \quad \text{DOF} = \text{Val}(\mathbf{\Sigma}) = \sum_{(i,j)\in T} \sigma_{ij} = \sum_j d_j - \sum_i c_i. \tag{2.9}$$

Here "DOF" refers to the phrase "degrees of freedom", while DOF is the corresponding number.

Remark 2.4 In most cases where SA succeeds, we use the canonical $(\mathbf{c}; \mathbf{d})$ to derive the structural index ν_S, which is an upper bound for the differentiation index [42]. However, in some special cases, a non-canonical offset pair may give a smaller ν_S in

(2.8) than the canonical offset pair does. We provide such examples in §7.3, and shall see that, for those DAEs of differentiation index 1, a non-canonical offset pair gives $\nu_S = 1$, while the canonical offset gives an overestimated $\nu_S = 2$.

To perform a numerical check for SA's success, or a *success check* for short, we attempt to compute numerically a consistent point at which $\mathbf{J}$ is nonsingular within roundoff. We assign initial values to an appropriate set of derivatives of x_j's and carry out the solution scheme (2.4–2.5) for stages $k = k_d : 0$. There is a minimal set of derivatives required for a DAE initial value problem; see discussion in [46].

When SA succeeds, the structural index is an upper bound for the differentiation index, that is $\nu_d \leq \nu_S$, and often they are the same [42]. In the failure case, our experience suggests the following.

(i) Usually (but not always) $\nu_d > \nu_S$ holds. That is, the index of a DAE is underestimated, and hence some equations do not receive enough differentiations.

(ii) The true DOF of a DAE is overestimated by $\mathrm{Val}(\mathbf{\Sigma})$.

We shall see these facts throughout the following chapters.

Example 2.5 We illustrate[1] the above concepts with the simple pendulum, a DAE of differentiation index 3.

$$\begin{aligned} 0 &= f_1 = x'' + x\lambda \\ 0 &= f_2 = y'' + y\lambda - G \\ 0 &= f_3 = x^2 + y^2 - \ell^2 \end{aligned} \tag{2.10}$$

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

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x & y & \lambda & c_i \end{array} \\ \begin{bmatrix} 2^\bullet & & 0^\circ \\ & 2^\circ & 0^\bullet \\ 0^\circ & 0^\bullet & \end{bmatrix} \begin{array}{c} 0 \\ 0 \\ 2 \end{array} \\ \begin{array}{ccc} 2 & 2 & 0 \end{array} \end{array} \quad \mathrm{Val}(\boldsymbol{\Sigma}) = 2 \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3'' \end{array} \begin{array}{c} \begin{array}{ccc} x'' & y'' & \lambda \end{array} \\ \begin{bmatrix} 1 & & x \\ & 1 & y \\ 2x & 2y & \end{bmatrix} \end{array} \quad \det(\mathbf{J}) = -2\ell^2$$

The state variables are x, y, λ; G is gravity and $\ell > 0$ is the length of the pendulum. Two HVTs of $\boldsymbol{\Sigma}$ are marked with $\bullet$ and $\circ$, respectively. A blank in $\boldsymbol{\Sigma}$ denotes $-\infty$, and a blank in $\mathbf{J}$ denotes 0. The row and column labels in $\mathbf{J}$, showing equations and variables differentiated to order c_i and d_j, aim to remind the reader of the formula for $\mathbf{J}$ in (2.6).

Since $\det(\mathbf{J}) = -2\ell^2 \neq 0$, $\mathbf{J}$ is nonsingular, and SA succeeds. The derivatives x'', y'', λ occur in a jointly linear way in (2.10), so a consistent point comprises

$$\big(t, x^{(<d_1)}, y^{(<d_2)}, \lambda^{(<d_3)}\big) = \big(t, x^{(<2)}, y^{(<2)}, \lambda^{(<0)}\big) = (t, x, x', y, y')$$

that satisfy (2.4) in stages $k = -2, -1$, that is, $f_3 = f_3' = 0$. The structural index is $\nu_S = \min_i c_i + 1 = 2 + 1 = 3$ (because $\min_j d_j = d_3 = 0$), which equals the differentiation index. The number of DOF is $\mathrm{DOF} = \mathrm{Val}(\boldsymbol{\Sigma}) = \sum_j d_j - \sum_i c_i = 4 - 2 = 2$. The solution scheme prescribed by the canonical offset is shown in Table 2.1. □

k	solve	for	using	Jacobian
-2	f_3	x, y	$-$	$[2x \quad 2y]$
-1	f_3'	x', y'	x, y	$[2x \quad 2y]$
≥ 0	$f_1^{(k)}, f_2^{(k)}, f_3^{(k+2)}$	$x^{(k+2)}, y^{(k+2)}, \lambda^{(k)}$	$x^{(<k+2)}, y^{(<k+2)}, \lambda^{(<k)}$	$\mathbf{J}$

Table 2.1: Solution scheme for the simple pendulum DAE.

2.2 Block triangularization of a DAE

In §2.2.1, we introduce notation for a block triangular form (BTF) of a sparsity pattern, and shall use such notation throughout this thesis. In §2.2.2, we review how to derive a BTF of a DAE.

We use bold font for matrices that may split into blocks, and also for the resulting sub-matrices. Individual entries of a matrix are in lowercase. For example, matrix $\mathbf{A}$ has sub-matrices $\mathbf{A}_{lm}$ and entries a_{ij}.

2.2.1 Block triangular form of a sparsity pattern

Let $R = 1:n$ be the set of indices of n rows (equations), and let $C = 1:n$ be the set of indices of n columns (variables). A *sparsity pattern* $\mathbf{A}$ is a subset of the Cartesian product $R \times C$ that contains row-column index pairs (i, j). We can view $\mathbf{A}$ as its incidence matrix (a_{ij}), where a_{ij} equals 1 if $(i, j) \in \mathbf{A}$ and 0 otherwise. A *transversal* of $\mathbf{A}$ is n positions in $\mathbf{A}$ with exactly one position in each row and each column. If $\mathbf{A}$ has some transversal, then it is *structurally nonsingular*. The union of all transversals of $\mathbf{A}$ comprise its *essential sparsity pattern* $\mathbf{A}_{\text{ess}}$ [47]. Obviously, $\mathbf{A}$ is structurally nonsingular if and only if $\mathbf{A}_{\text{ess}}$ is nonempty.

Assume henceforth that $\mathbf{A}$ is structurally nonsingular. Let $\mathbf{P}$ and $\mathbf{Q}$ be two suitable permutation matrices for $\mathbf{A}$, such that the permuted incidence matrix $\mathbf{A}' = \mathbf{PAQ}$ can be written in a $p \times p$ block form

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1p} \\ & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2p} \\ & & \ddots & \vdots \\ & & & \mathbf{A}_{pp} \end{bmatrix}, \tag{2.11}$$

where each diagonal block $\mathbf{A}_{ww}$, $w = 1:p$, is structurally nonsingular and square of size $N_w > 0$. Here $\mathbf{A}_{kl}$, $k, l = 1:p$, is a submatrix in the context of incidence matrix, or a sparsity sub-pattern in the context of a sparsity pattern. We say the block form (2.11) is a *BTF* of $\mathbf{A}$, in which a below diagonal submatrix $\mathbf{A}_{kl}$, $k > l$, is empty.

A sparsity pattern is *irreducible* if it cannot be permuted to a BTF (2.11) with $p > 1$ [9]; otherwise it is *reducible*. A BTF is an *irreducible BTF* if its diagonal blocks are all irreducible; otherwise it is a *reducible BTF* [47]. Hence, if (2.11) is the irreducible case, then p is the largest number of diagonal blocks among all possible BTFs of $\mathbf{A}'$. Since every structurally nonsingular $\mathbf{A}$ is in a BTF of $p = 1$, such a BTF is said to be *trivial*, while a BTF of $p > 1$ is *nontrivial*.

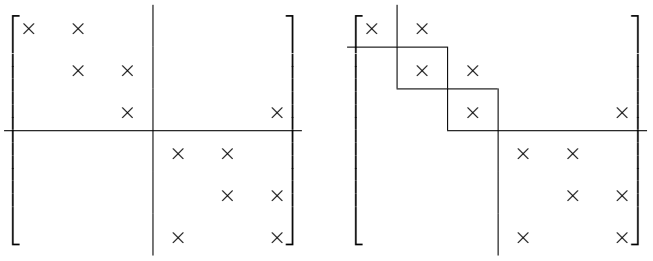
Without loss of generality, we deal with matrices that are already permuted to some BTF. In other words, the details of permutations are not important to our exposition, and we can use the same notation for permuted matrices. For example, we can leave out the apostrophe in (2.11). When we say block w of a matrix in some BTF, we shall refer to the wth diagonal block submatrix.

For each block $w \in 1:p$, we define the index set

$$B_w = \text{the set of indices } i \text{ that belong to block } w.$$

Another useful notation is blockOf(i) that denotes the block number w such that index $i \in B_w$. Since each diagonal block is square, each notation applies to rows and columns equally. To summarize, for $i \in 1:n$ and $w \in 1:p$,

$$\text{blockOf}(i) = w \Longleftrightarrow i \in B_w \Longleftrightarrow \sum_{w=1}^{w-1} N_w + 1 \leq i \leq \sum_{w=1}^{w} N_w.$$



w	N_w	B_w
1	1	$\{1\}$
2	1	$\{2\}$
3	1	$\{3\}$
4	3	$\{4,5,6\}$

(a) (b)

Figure 2.1: (a) Two nontrivial BTFs of the same sparsity pattern. The left one is reducible with number of blocks $p = 2$. The right one is irreducible with $p = 4$. (b) Block information for the irreducible BTF.

Example 2.6 We illustrate in Figure 2.1 the above block notation with a sparsity pattern of two nontrivial BTFs. □

The following lemma connects the transversals of a sparsity pattern $\mathbf{A}$ and the transversals of its diagonal blocks in some BTF.

Lemma 2.7 *[47, Lemma 2.4] Any transversal T of a sparsity pattern $\mathbf{A}$ is contained in the union of the diagonal blocks of any BTF of $\mathbf{A}$, that is, $T \subseteq \mathbf{A}_{11} \cup \cdots \cup \mathbf{A}_{pp}$.*

Equivalently, the intersection of T with block w of $\mathbf{A}$ is a transversal T_w of $\mathbf{A}_{ww}$.

2.2.2 Block triangular forms of a DAE

The natural sparsity pattern of a DAE indicates if a variable x_j occurs in an equation f_i or not. Each such occurrence corresponds to a finite entry σ_{ij} in $\mathbf{\Sigma}$, and hence we have

$$\mathbf{S} = \{\, (i,j) \mid \sigma_{ij} > -\infty \,\} \qquad \text{(the sparsity pattern of } \mathbf{\Sigma}\text{)}.$$

From the concepts introduced in §2.1, it is not difficult to show that the following arguments are equivalent.

Sparsity pattern $\mathbf{S}$ of $\mathbf{\Sigma}$ is structurally nonsingular.

$\Leftrightarrow$ $\mathbf{S}$ has some transversal.

$\Leftrightarrow$ $\mathbf{\Sigma}$ has some finite transversal and hence has a finite Val($\mathbf{\Sigma}$).

$\Leftrightarrow$ DAE is structurally well posed (SWP).

$\Leftrightarrow$ There is some one-to-one correspondence between equations and variables.

Another BTF derives from the sparsity pattern $\mathbf{S}_0 = \mathbf{S}_0(\mathbf{c};\mathbf{d})$ of a System Jacobian $\mathbf{J} = \mathbf{J}(\mathbf{c};\mathbf{d})$ as defined in (2.6):

$$\mathbf{S}_0(\mathbf{c};\mathbf{d}) = \left\{ (i,j) \;\mid\; d_j - c_i = \sigma_{ij} \right\} \qquad \text{(the sparsity pattern of } \mathbf{J}\text{)}. \tag{2.12}$$

We call $\mathbf{S}_0(\mathbf{c};\mathbf{d})$ a *Jacobian pattern* for short, and also write it as $\mathbf{S}_0$ for brevity, omitting the argument $(\mathbf{c};\mathbf{d})$. By (2.2), $d_j - c_i = \sigma_{ij}$ holds on an HVT T of $\mathbf{\Sigma}$, so T is a transversal of $\mathbf{S}_0$.

A less obvious set contains the positions that contribute to det($\mathbf{J}$):

$$\mathbf{S}_{\text{ess}} = \text{the union of all HVTs of } \mathbf{\Sigma} \qquad \text{(the essential sparsity pattern of } \mathbf{\Sigma}\text{)}, \tag{2.13}$$

which is also the essential sparsity pattern of $\mathbf{S}_0$ for any offset pair $(\mathbf{c};\mathbf{d})$ [47, Lemma 3.1].

An equality $d_j - c_i = \sigma_{ij}$ on some HVT also holds on all HVTs [46]. Such an

equality implies $\sigma_{ij} > -\infty$, so we have

$$\mathbf{S}_{\text{ess}} \subseteq \mathbf{S}_0 \subseteq \mathbf{S} \qquad \text{for any offset pair } (\mathbf{c}; \mathbf{d}).$$

A BTF of $\boldsymbol{\Sigma}$ means a BTF based on the sparsity pattern $\mathbf{S}$ of $\boldsymbol{\Sigma}$, so a signature matrix in its BTF has the form

$$\begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \cdots & \boldsymbol{\Sigma}_{1p} \\ -\boldsymbol{\infty} & \boldsymbol{\Sigma}_{22} & \cdots & \boldsymbol{\Sigma}_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ -\boldsymbol{\infty} & \cdots & -\boldsymbol{\infty} & \boldsymbol{\Sigma}_{pp} \end{bmatrix},$$

where the bolded $-\boldsymbol{\infty}$ means a below diagonal block that is filled with $-\infty$'s. A BTF of $\mathbf{J}$ means a BTF based on the Jacobian pattern $\mathbf{S}_0$, so a System Jacobian in its BTF has the form

$$\begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} & \cdots & \mathbf{J}_{1p} \\ \mathbf{0} & \mathbf{J}_{22} & \cdots & \mathbf{J}_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_{pp} \end{bmatrix},$$

where below diagonal blocks are identically zero.

Our experience suggests that the irreducible BTF of $\mathbf{J}$ is often significantly finer than that of $\boldsymbol{\Sigma}$. We refer to the former BTF as the *fine BTF*, and to the latter as the *coarse BTF*. We call the diagonal blocks in the fine BTF *fine blocks*, and call those in the coarse BTF *coarse blocks*.

Assume that a Jacobian pattern $\mathbf{S}_0$ is permuted into a $p \times p$ BTF, which is not

necessarily irreducible. Following this BTF, we apply the same permutations on $\mathbf{J}$ and $\mathbf{\Sigma}$, and write them in $p \times p$ block forms:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} & \cdots & \mathbf{J}_{1p} \\ \mathbf{0} & \mathbf{J}_{22} & \cdots & \mathbf{J}_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_{pp} \end{bmatrix}, \qquad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} & \cdots & \mathbf{\Sigma}_{1p} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} & \cdots & \mathbf{\Sigma}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Sigma}_{p1} & \mathbf{\Sigma}_{p2} & \cdots & \mathbf{\Sigma}_{pp} \end{bmatrix}. \tag{2.14}$$

We call this procedure a *block triangularization* of the DAE. When we say block w of a DAE, we shall refer to the rows and columns of the wth diagonal block, or refer to the functions and variables (and derivatives of them) in this block, depending on context.

Notice that in (2.14) $\mathbf{\Sigma}$ may not be permuted into a BTF. That is, every σ_{ij} in the below diagonal blocks of $\mathbf{\Sigma}$ is not necessarily $-\infty$, but must satisfy $\sigma_{ij} < d_j - c_i$ as $J_{ij} \equiv 0$.

Example 2.8 We illustrate the coarse and fine BTFs with the (artificially) modified double pendula DAE MOD2PEND (2.15) in [38]. The state variables are x, y, λ, u, v, μ; G is gravity, $\ell > 0$ is the length of the first pendulum, and α is a constant.

$$
\begin{aligned}
0 &= f_1 = x'' + x\lambda \\
0 &= f_2 = y'' + y\lambda + (x')^3 - G \\
0 &= f_3 = x^2 + y^2 - \ell^2 \\
0 &= f_4 = u'' + u\mu \\
0 &= f_5 = (v''')^3 + v\mu - G \\
0 &= f_6 = u^2 + v^2 - (\ell + \alpha\lambda)^2 + \lambda''.
\end{aligned}
\tag{2.15}
$$

In the original index-5 double pendula from [34],

$$
\begin{aligned}
0 &= f_2 = y'' + y\lambda - G \\
0 &= f_5 = v'' + v\mu - G \\
0 &= f_6 = u^2 + v^2 - (\ell + \alpha\lambda)^2.
\end{aligned}
$$

Figures 2.2–2.5 illustrate a block triangularization of MOD2PEND.

This DAE has two coarse blocks of size 3. The first one, comprising equations f_5, f_4, f_6 and variables v, μ, u, can further decompose into three fine blocks of size 1, while the second coarse block, comprising equations f_3, f_2, f_1 and variables x, y, λ, is irreducible. Hence there are four blocks in the fine BTF.

The sparsity pattern $\mathbf{S}_0$ of $\mathbf{J}$ is in Figure 2.1(a), so the fine BTF information is in Figure 2.1(b). □

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ \hline d_j \end{array} \begin{array}{c} \begin{array}{cccccc} x & y & \lambda & u & v & \mu \end{array} \\ \left[\begin{array}{cccccc} 2^{\bullet} & & 0 & & & \\ 1 & 2 & 0^{\bullet} & & & \\ 0 & 0^{\bullet} & & & & \\ & & & 2 & & 0^{\bullet} \\ & & & & 3^{\bullet} & 0 \\ & & 2 & 0^{\bullet} & 0 & \end{array}\right] \\ \begin{array}{cccccc} 6 & 6 & 4 & 2 & 3 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ \hline 4 \\ 4 \\ 6 \\ 0 \\ 0 \\ 2 \\ \hline \mathrm{Val}(\boldsymbol{\Sigma}) = 5 \end{array}$$

$$\mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x & y & \lambda & u & v & \mu \end{array} \\ \left[\begin{array}{cccccc} 1 & & x & & & \\ & 1 & y & & & \\ 2x & 2y & & & & \\ & & & 1 & & u \\ & & & & 2v''' & v \\ & & 1 & 2u & & \end{array}\right] \end{array} \qquad \det(\mathbf{J}) = 8\ell^2 u^2 v'''$$

Figure 2.2: $\boldsymbol{\Sigma}$ and $\mathbf{J}$ of MOD2PEND (2.15).

$$\mathbf{S} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x & y & \lambda & u & v & \mu \end{array} \\ \left[\begin{array}{cccccc} \times & & \times & & & \\ \times & \times & \times & & & \\ \times & \times & & & & \\ & & & \times & & \times \\ & & & & \times & \times \\ & & \times & \times & \times & \end{array}\right] \end{array} \qquad \mathbf{S}_0 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x & y & \lambda & u & v & \mu \end{array} \\ \left[\begin{array}{cccccc} \times^{\bullet} & & \times^{\bullet} & & & \\ & \times^{\bullet} & \times^{\bullet} & & & \\ \times^{\bullet} & \times^{\bullet} & & & & \\ & & & \times & & \times^{\bullet} \\ & & & & \times^{\bullet} & \times \\ & & \times & \times^{\bullet} & & \end{array}\right] \end{array}$$

Figure 2.3: Sparsity patterns $\mathbf{S}$ and $\mathbf{S}_0$ of MOD2PEND (2.15). In $\mathbf{S}_0$, the positions marked by $\bullet$ lie on some HVT and compose the essential sparsity pattern $\mathbf{S}_{\text{ess}}$.

$$\mathbf{S} = \begin{array}{c} \\ f_5 \\ f_4 \\ f_6 \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x & y & \lambda & u & v & \mu \end{array} \\ \left[\begin{array}{cccccc} \times & \times & & & & \\ & \times & \times & & & \\ \times & & \times & & & \times \\ \hline & & & \times & \times & \\ & & & \times & \times & \times \\ & & & \times & & \times \end{array}\right] \end{array} \qquad \mathbf{S}_0 = \begin{array}{c} \\ f_5 \\ f_4 \\ f_6 \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x & y & \lambda & u & v & \mu \end{array} \\ \left[\begin{array}{cccccc} \times^{\bullet} & \times & & & & \\ & \times^{\bullet} & \times & & & \\ & & \times^{\bullet} & & & \times \\ \hline & & & \times^{\bullet} & \times & \\ & & & & \times^{\bullet} & \times \\ & & & \times & & \times^{\bullet} \end{array}\right] \end{array}$$

Figure 2.4: Permuted sparsity patterns $\mathbf{S}$ and $\mathbf{S}_0$ of MOD2PEND (2.15).

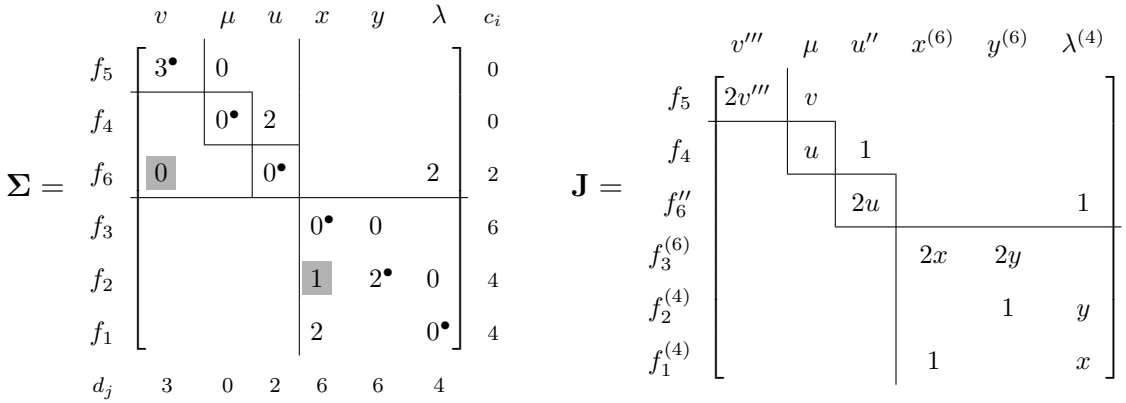


Figure 2.5: $\mathbf{\Sigma}$ and $\mathbf{J}$ permuted to BTF based on a Jacobian sparsity pattern $\mathbf{S}_0$.

If we state Lemma 2.7 in the context of a Jacobian pattern, then we have the following lemma.

Lemma 2.9 *[47, Lemma 3.3] Assume that a Jacobian pattern $\mathbf{S}_0$ is in some BTF. Let $(\mathbf{\Sigma}_{wm})_{w,m=1:p}$ be the corresponding sub-matrices of $\mathbf{\Sigma}$. Then a highest-value transversal T of $\mathbf{\Sigma}$ is the union of HVTs T_w of the diagonal blocks $\mathbf{\Sigma}_{ww}$: $T = T_1 \cup \cdots \cup T_p$.*

This lemma is not difficult to prove, given that a transversal T of $\mathbf{S}_0$ is the union of transversals T_w of the diagonal blocks of $\mathbf{S}_0$.

The following lemma is useful for proving the main Theorems 6.1 and 6.4 of the block conversion methods in §6.

Lemma 2.10 *Assume that $\mathbf{\Sigma}$ has a finite* Val($\mathbf{\Sigma}$) *and is in a $p \times p$ block form as in (2.14) with square diagonal blocks. Let $\mathbf{c}$ and $\mathbf{d}$ be two nonnegative integer n-vectors. Assume also that*

(i) $d_j - c_i > \sigma_{ij}$ *holds for all entries below the diagonal blocks of* $\mathbf{\Sigma}$*,*

(ii) $d_j - c_i \geq \sigma_{ij}$ *holds elsewhere, and*

(iii) $\mathrm{Val}(\mathbf{\Sigma}) = \sum_j d_j - \sum_i c_i$*.*

Then

(a) $(\mathbf{c}; \mathbf{d})$ *is a valid offset pair of* $\mathbf{\Sigma}$*,*

(b) the block form of $\mathbf{\Sigma}$ *is a BTF of the Jacobian pattern* $\mathbf{S}_0$*, and*

(c) a HVT of $\mathbf{\Sigma}$ *is the union of HVTs of the diagonal blocks* $\mathbf{\Sigma}_{ww}$*, for all* $w = 1 : p$*.*

Proof. (a) We let T denote an HVT of $\mathbf{\Sigma}$. Since $\mathrm{Val}(\mathbf{\Sigma})$ is finite by condition *(iii)*, $\sigma_{ij} \geq 0$ for all $(i, j) \in T$. For $(\mathbf{c}; \mathbf{d})$ to be a valid offset of $\mathbf{\Sigma}$, $d_j - c_i \geq \sigma_{ij}$ must hold for all $i, j = 1 : n$, with equalities for all $(i, j) \in T$.

By conditions *(i)* and *(ii)*, $d_j - c_i \geq \sigma_{ij}$ holds everywhere. Summing these inequalities over T gives

$$\sum_{(i,j)\in T} (d_j - c_i) \geq \sum_{(i,j)\in T} \sigma_{ij}.$$

The left-hand side equals $\sum_j d_j - \sum_i c_i$, and the right-hand side equals $\mathrm{Val}(\mathbf{\Sigma})$ by definition. By *(iii)*, these two values are equal, so $d_j - c_i = \sigma_{ij}$ holds for all $(i, j) \in T$, and $(\mathbf{c}; \mathbf{d})$ is valid for $\mathbf{\Sigma}$.

(b) By *(i)*, the below diagonal blocks in $\mathbf{S}_0$, derived from $\mathbf{\Sigma}$ and $(\mathbf{c}; \mathbf{d})$ using (2.12), are empty. By the definition of a BTF of a Jacobian sparsity pattern, $\mathbf{S}_0$ is in a BTF as described by the $p \times p$ block form.

(c) follows immediately from (b) and Lemma 2.9. □

Following a $p \times p$ BTF based on a Jacobian pattern $\mathbf{S}_0$, we can write any valid offset pair $(\mathbf{c}; \mathbf{d})$ of $\mathbf{\Sigma}$ in a block form as

$$(\mathbf{c}_1; \mathbf{d}_1), (\mathbf{c}_2; \mathbf{d}_2), \ldots, (\mathbf{c}_p; \mathbf{d}_p), \tag{2.16}$$

where each of the sub-vectors $\mathbf{c}_w$ and $\mathbf{d}_w$ is of length N_w, $w = 1 : p$.

Lemma 2.11 *Assume that a Jacobian pattern $\mathbf{S}_0$ is in some BTF. Let $(\mathbf{c}; \mathbf{d})$ be valid for $\mathbf{\Sigma}$, and write $(\mathbf{c}; \mathbf{d})$ into block form as in (2.16). Then $(\mathbf{c}_w; \mathbf{d}_w)$ is a valid offset pair of $\mathbf{\Sigma}_{ww}$.*

Proof. Let T be an HVT of $\mathbf{\Sigma}$. By Lemma 2.9, the intersection of T with block w is an HVT T_w of $\mathbf{\Sigma}_{ww}$. Then $d_j - c_i = \sigma_{ij}$ holds for all $(i, j) \in T_w \subseteq T$. Since $(\mathbf{c}; \mathbf{d})$ is valid for $\mathbf{\Sigma}$, $d_j - c_i \geq \sigma_{ij}$ and $c_i \geq 0$ hold on $\mathbf{\Sigma}_{ww}$, that is, for all $i, j \in B_w$. Thus the offset pair $(\mathbf{c}_w; \mathbf{d}_w)$ matched to block w satisfies the conditions (2.2) for being valid for $\mathbf{\Sigma}_{ww}$. □

From the view of Lemma 2.11, we can regard each diagonal block $\mathbf{\Sigma}_{ww}$ as a signature matrix in its own right. Equivalently, each block w, having N_w equations in N_w variables, can be viewed as a sub-DAE in its own right also, with a signature matrix $\mathbf{\Sigma}_{ww}$, a finite value $\mathrm{Val}(\mathbf{\Sigma}_{ww})$, a *local offset pair* $(\mathbf{c}_w; \mathbf{d}_w)$, and a *sub-Jacobian* $\mathbf{J}_{ww}$. Expressions that contribute to entries in an off-diagonal block $\mathbf{\Sigma}_{wm}$, $w \neq m$, can be considered as driving terms, or equivalently, the influence of variables in block m on those in block w. We refer to $(\mathbf{c}; \mathbf{d})$ of $\mathbf{\Sigma}$ as a *global offset pair*. The reader is referred to [47] for more theoretical results about block triangularization and global and local offset pairs.

Chapter 3

When structural analysis fails

In this chapter, we investigate several cases where SA fails. In these cases, SA produces an identically singular system Jacobian, while the DAE may be solvable. In §3.1, we give a definition of a structurally singular DAE. In §3.2, we classify the SA's failure cases into two types.

We use $u \equiv 0$ to mean u is *identically zero*. In contrast, by $u \not\equiv 0$ we mean u is *generically nonzero*. This u may be a scalar, vector, or matrix, depending on context.

3.1 Success check

To perform a success check for SA on a structurally well-posed DAE, we may simply follow the solution scheme in (2.4–2.5) and solve the systems for stages $k = k_d : 0$. We can also obtain a symbolic form of a System Jacobian $\mathbf{J}$ in (2.6) and evaluate its value once we find the derivatives therein. For example, we can evaluate $\mathbf{J}$ of the pendulum DAE (2.10) as soon as we find x and y at stage $k = -2$.

In the definitions that follow, we let $\mathbf{A}$ be an $n \times n$ matrix function.

Definition 3.1 *An (i, j) position is a structural zero of $\mathbf{A}$ if $a_{ij} \equiv 0$; otherwise it is a structural nonzero.*

Definition 3.2 *Matrix $\mathbf{A}$ is structurally nonsingular if it has a transversal of structural nonzeros; otherwise it is structurally singular.*

Definition 3.3 *Matrix $\mathbf{A}$ is identically singular if $\det(\mathbf{A}) \equiv 0$; otherwise it is generically nonsingular.*

For a matrix function, being structurally singular is a special case of being identically singular. Similarly, being generically nonsingular is a special case of being structurally nonsingular.

Example 3.4 Consider the following three matrix functions of variables x and y:

$$\mathbf{A}_1 = \begin{bmatrix} x & x \\ 0 & y-y \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} x & x \\ y & y \end{bmatrix}, \quad \text{and} \quad \mathbf{A}_3 = \begin{bmatrix} x & y \\ y & x \end{bmatrix}.$$

$\mathbf{A}_1$ is identically singular because $\det(\mathbf{A}_1) \equiv 0$. It is structurally singular also, since there exists no transversal of structural nonzeros, as position $(2, 2)$ is a structural zero position.

$\mathbf{A}_2$ is identically singular (because $\det(\mathbf{A}_2) \equiv 0$) but structurally nonsingular: there are two transversals of structural nonzeros.

$\mathbf{A}_3$ is both structurally and generically nonsingular, since $\det(\mathbf{A}_3) = x^2 - y^2 \not\equiv 0$. Only when $x = \pm y$ is this determinant 0. □

In the following, we denote a DAE (1.1) by $\mathcal{F}$ and define two concepts for it:

- a *structural zero in the System Jacobian* $\mathbf{J}$, and
- a *structurally singular DAE* whose $\mathbf{J}$ is identically singular.

Let $\mathcal{J}$ be the (infinite) set of index-order (j, l) pairs

$$\mathcal{J} = \big\{\, (j,l) \mid j = 1:n\,,\ l \in \mathbb{N} \,\big\}.$$

Given an n-vector function $\mathbf{x} = \mathbf{x}(t)$ that is sufficiently smooth (but not necessarily a solution of $\mathcal{F}$), we let

$$\mathbf{x}_{\mathcal{J}} = \big\{\, x_j^{(l)} \mid (j,l) \in \mathcal{J} \,\big\}.$$

For a finite subset J of $\mathcal{J}$, $\mathbf{x}_J$ contains derivatives $x_j^{(l)}$ whose index-order pairs (j, l) range over J. We may also regard $\mathbf{x}_J$ as a $|J|$-vector, in which the ordering of $x_j^{(l)}$ does not matter.

Now we define the *derivative set* of $\mathcal{F}$ as

$$\mathrm{derset}(\mathcal{F}) = \big\{\, (j,l) \mid x_j^{(l)} \text{ occurs in } \mathcal{F} \,\big\}.$$

Then the derivatives occurring in $\mathcal{F}$ can be denoted concisely as $\mathbf{x}_{\mathrm{derset}(\mathcal{F})}$.

By a *value point* we mean a $\xi = (t, \mathbf{x}_{\mathrm{derset}(\mathcal{F})}) \in \mathbb{R} \times \mathbb{R}^{|\mathrm{derset}(\mathcal{F})|}$ that contains values for t and values for the derivative symbols in $\mathbf{x}_{\mathrm{derset}(\mathcal{F})}$.

Example 3.5 In the simple pendulum DAE (2.10), denote x, y, λ as x_1, x_2, x_3, respectively. Let $\ell = 5$ and $G = 9.8$. Then

$$\mathrm{derset}(\mathcal{F}) = \big\{\, (1,0),\ (1,2),\ (2,0),\ (2,2),\ (3,0) \,\big\}.$$

A possible value point can be

$$\xi = (t, x_1, x_1'', x_2, x_2'', x_3) = (2, 3, -3, 4, 1.6, 1),$$

which satisfies f_1 and f_3 but not f_2. □

Similarly, we define the *derivative set of* $\mathbf{J}$:

$$\mathrm{derset}(\mathbf{J}) = \big\{\, (j,l) \mid x_j^{(l)} \text{ occurs in } \mathbf{J} \,\big\}.$$

From (2.6), a derivative occurring in $\mathbf{J}$ must also occur in $\mathcal{F}$, but not vice versa. For example, in the pendulum DAE, x'', y'', λ do not appear in $\mathbf{J}$, and $\mathrm{derset}(\mathbf{J}) = \big\{\, (1,0), (2,0) \,\big\}$; cf. Example 2.5. The derivative set of $\mathbf{J}$ is a subset of that of $\mathcal{F}$: $\mathrm{derset}(\mathbf{J}) \subseteq \mathrm{derset}(\mathcal{F})$.

Definition 3.6 ***(Structural zero in System Jacobian)*** *An (i,j) position is a structural zero in $\mathbf{J}$, if J_{ij} is identically zero at all value points $\xi \in \mathbb{R}^{|\mathrm{derset}(\mathcal{F})|+1}$ that satisfy some equations from*

$$0 = f_i^{(m)}, \quad m \geq 0, \quad i = 1:n. \tag{3.1}$$

Otherwise, (i,j) is a structural nonzero of $\mathbf{J}$.

For the present purpose, we do not require the DAE to have a unique solution, or even any solution. That is, we do not consider existence and uniqueness of the DAE at this stage, while identifying structural zeros of $\mathbf{J}$ and discussing its singularity.

Recall (2.6) that defines $\mathbf{J}$. If $d_j - c_i > \sigma_{ij}$, then $J_{ij} \equiv 0$ and thus position (i,j) is a structural zero in $\mathbf{J}$. The converse is not true; see the following example.

Example 3.7 Consider an artificially modified simple pendulum DAE. We multiply the first equation f_1 by $x^2 + y^2 - \ell^2$ and obtain

$$\begin{aligned} 0 &= f_1 = (x'' + x\lambda)(x^2 + y^2 - \ell^2) \\ 0 &= f_2 = y'' + y\lambda - G \\ 0 &= f_3 = x^2 + y^2 - \ell^2. \end{aligned} \tag{3.2}$$

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x & y & \lambda & c_i \end{array} \\ \begin{bmatrix} 2^{\bullet} & 0 & 0 \\ & 2 & 0^{\bullet} \\ 0 & 0^{\bullet} & \end{bmatrix} \begin{array}{c} 0 \\ 0 \\ 2 \end{array} \\ \begin{array}{cccc} 2 & 2 & 0 & \end{array} \end{array} \quad \mathrm{Val}(\boldsymbol{\Sigma}) = 2$$

$$\mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3'' \end{array} \begin{array}{c} \begin{array}{ccc} x'' & y'' & \lambda \end{array} \\ \begin{bmatrix} \mu & & x\mu \\ & 1 & y \\ 2x & 2y & \end{bmatrix} \end{array} \quad \det(\mathbf{J}) = -2\mu(x^2 + y^2)$$

Here $\mu = x^2 + y^2 - \ell^2$. For this DAE,

$$\begin{aligned} \mathrm{derset}(\mathcal{F}) &= \big\{ (1,0),\, (1,2),\, (2,0),\, (2,2),\, (3,0) \big\}, \\ \xi &= (t, x_1, x_1'', x_2, x_2'', x_3) \in \mathbb{R}^6, \\ \text{and} \qquad \mathrm{derset}(\mathbf{J}) &= \big\{ (1,0),\, (2,0) \big\}. \end{aligned}$$

If we evaluate $\mathbf{J}$ at some random $\xi \in \mathbb{R}^6$, then μ is generically nonzero, and so are positions (f_1, x) and (f_1, λ). In this case, $\mathbf{J}$ is generically nonsingular. However, we should evaluate $\mathbf{J}$ at some ξ that satisfies $\mu = f_3 = x^2 + y^2 - \ell^2 = 0$. According to Definition 3.6, positions (f_1, x) and (f_1, λ) are structural zeros of $\mathbf{J}$. □

We give a definition for *structural regularity* of a DAE.

Definition 3.8 ***(Structurally regular DAE)*** *A DAE is structurally singular if* $\mathbf{J}$ *is identically singular at all value points* $\xi \in \mathbb{R}^{|\text{derset}(\mathcal{F})|+1}$ *that satisfy some equations from (3.1). Otherwise the DAE is structurally nonsingular, or structurally regular.*

Example 3.9 In the previous example, positions (f_1, x) and (f_1, λ) are structural zeros of $\mathbf{J}$ at any point that satisfies $f_3 = 0$. Then $\mathbf{J}$ is structurally singular, and by Definition 3.8, DAE (3.2) is structurally singular.

In fact, it can be shown that a solution of (2.10) is a solution to (3.2), but not vice versa. □

Example 3.10 Consider the DAE in [1, p. 235, Example 9.2], written as

$$\begin{aligned} 0 = f_1 &= -y_1' + y_3 \\ 0 = f_2 &= \quad y_2(1 - y_2) \\ 0 = f_3 &= \quad y_1 y_2 + y_3(1 - y_2) - t. \end{aligned} \tag{3.3}$$

$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 1^{\bullet} & & 0 \\ & 0^{\bullet} & \\ 0 & 0 & 0^{\bullet} \end{bmatrix} \\ \begin{array}{ccc} 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 0 \\ \text{Val}(\mathbf{\Sigma}) = 1 \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} y_1' & y_2 & y_3 \end{array} \\ \begin{bmatrix} -1 & & 1 \\ & 1 - 2y_2 & \\ & y_1 - y_3 & 1 - y_2 \end{bmatrix} \end{array} \quad \det(\mathbf{J}) = -(1 - 2y_2)(1 - y_2)$$

SA gives $\nu_S = 1$, and $\det(\mathbf{J})$ depends solely on y_2. From $f_2 = 0$, either $y_2 = 0$ or $y_2 = 1$. To examine if $\mathbf{J}$ is nonsingular, we consider each of the following two cases.

- If $y_2 = 0$, then $\det(\mathbf{J}) = -1$ and SA succeeds. In this case (3.3) is of differentiation index 1.

- If $y_2 = 1$, then $\det(\mathbf{J}) = 0$ and SA fails on this structurally singular DAE. The failure comes as no surprise because DAE (3.3) is now of differentiation index 2 and SA underestimates its index; see the discussion in §1.2. □

Remark 3.11 We hereby distinguish the difference between a structurally ill-posed (SIP) DAE and a structurally singular DAE. A SIP DAE has no finite transversal in $\mathbf{\Sigma}$ and hence no valid offset pair $(\mathbf{c}; \mathbf{d})$, so we cannot define a System Jacobian. In contrast, a structurally singular DAE has some valid $(\mathbf{c}; \mathbf{d})$ and an identically singular $\mathbf{J}$.

If $\mathbf{J}$ is generically nonsingular but numerically singular when evaluated at a value point ξ, then we say the DAE is *locally unsolvable* at ξ.

Example 3.12 [12] Consider

$$\begin{aligned} 0 &= f_1 = -x' + y \\ 0 &= f_2 = x + \cos(t) y. \end{aligned} \tag{3.4}$$

$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1^{\bullet} & 0 \\ 0 & 0^{\bullet} \end{bmatrix} \\ \begin{array}{cc} 1 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{cc} x' & y \end{array} \\ \begin{bmatrix} -1 & 1 \\ & \cos(t) \end{bmatrix} \end{array}$$

$$\mathrm{Val}(\mathbf{\Sigma}) = 1 \qquad \det(\mathbf{J}) = -\cos(t)$$

Since $\det(\mathbf{J})$ is generically nonzero, DAE (3.4) is structurally nonsingular. We can integrate this problem from $t = 0$ with any consistent initial value $(x(0), y(0)) =$

(x_0, y_0), and the problem is index-1 (both differentiation and structural indices) as long as $\det(\mathbf{J}) \neq 0$. However, $\mathbf{J}$ is singular at $t = t_k = \left(k + \frac{1}{2}\right)\pi$, $k = 0, 1, \ldots$. Hence, we say the DAE has a singularity point at each t_k. □

3.2 Identifying structural analysis's failure

We give below a definition for the *true highest-order derivative* (HOD) of a variable x_j in a function u. This u may be a scalar, vector, or matrix, depending on context.

Definition 3.13 ***(True highest-order derivative)*** *The true HOD of x_j in u is*

$$\sigma(x_j, u) = \begin{cases} \textit{highest order of derivatives of } x_j \textit{ on which } u \text{ truly } \textit{depends; or} \\ -\infty \quad \textit{if } u \textit{ does not depend on } x_j \textit{ at all.} \end{cases} \tag{3.5}$$

By "truly" we mean that, if $r = \sigma(x_j, u) > -\infty$, then u is not a constant with respect to $x_j^{(r)}$. For example, $u = x' + \cos^2 x'' + \sin^2 x'' = x' + 1$ truly depends on x' but not x'', resulting in $\sigma(x, u) = 1$. If an f_i truly depends on $x_j^{(\sigma_{ij})}$, then $\sigma(x_j, f_i) = \sigma_{ij}$, so (3.5) can be considered a generalization of (2.1). However, we should note that the problem of detecting such true dependence (which is equivalent to recognizing zero) in any expressions is unsolvable in general [51].

The DAETS and DAESA codes, which implement [35, Algorithm 4.1], find the *formal HOD* of x_j in u, denoted by $\widetilde{\sigma}(x_j, u)$, instead of the true HOD. By "formal" we mean the dependence of an expression (or function) on a derivative without symbolic simplifications. For example, $u = x' + \cos^2 x'' + \sin^2 x''$ formally depends on x'' and hence $\widetilde{\sigma}(x, u) = 2$, while $u = x' + 1$ and $\sigma(x, u) = 1$. We can write $\widetilde{\sigma}_{ij} = \widetilde{\sigma}(x_j, f_i)$ in a

similar way to $\sigma_{ij} = \sigma(x_j, f_i)$, so DAETS and DAESA find formal signature entries $\widetilde{\sigma}_{ij}$.

Since the formal dependence is also used in [35, §4], we can adopt the rules in [35, Lemma 4.1], which indicate how to propagate the formal HOD in an expression. The most useful rules are:

- if a variable v is a purely algebraic function of a set U of variables u, then

$$\widetilde{\sigma}(x_j, v) = \max_{u \in U} \widetilde{\sigma}(x_j, u), \tag{3.6}$$

and

- if $v = \mathrm{d}^p u/\mathrm{d}t^p$, where $p > 0$, then

$$\widetilde{\sigma}(x_j, v) = \widetilde{\sigma}(x_j, u) + p. \tag{3.7}$$

These rules are proved in [35], to which we refer for details. We illustrate these rules in Example 3.14.

Example 3.14 Let $u = (x_1 x_2)' - x_1' x_2$. Applying (3.6) and (3.7), we derive the formal HOD of x_1 in u:

$$\begin{aligned}
\widetilde{\sigma}(x_1, u) &= \max\{\, \widetilde{\sigma}(x_1, (x_1x_2)')\,,\, \widetilde{\sigma}(x_1, x_1'x_2) \,\} \\
&= \max\{\, \widetilde{\sigma}(x_1, x_1x_2) + 1,\, \max\{\, \widetilde{\sigma}(x_1, x_1')\,, \widetilde{\sigma}(x_1, x_2) \,\} \,\} \\
&= \max\{\, \max\{\, \widetilde{\sigma}(x_1, x_1)\,,\, \widetilde{\sigma}(x_1, x_2) \,\} + 1, \max\{\, 1, -\infty \,\} \,\} \\
&= \max\{\, \max\{\, 0, -\infty \,\} + 1,\, 1 \,\} \\
&= \max\{\, 0 + 1,\, 1 \,\} = 1.
\end{aligned}$$

Similarly $\widetilde{\sigma}(x_2, u) = 1$. A simplification on u gives

$$u = (x_1 x_2)' - x_1' x_2 = x_1 x_2' + \cancel{x_1' x_2} - \cancel{x_1' x_2}. \tag{3.8}$$

Hence, the true HOD of x_1 in u is $\sigma(x_1, u) = 0$, and that of x_2 in u is $\sigma(x_2, u) = 1$. □

The cancellation occurring in (3.8) is a *hidden symbolic cancellation*. When such cancellations happen, a formal HOD $\widetilde{\sigma}(x_j, u)$ can overestimate the true HOD $\sigma(x_j, u)$. If u is an equation f_i, then the formal $\widetilde{\sigma}_{ij} = \widetilde{\sigma}(x_j, f_i)$ may not be the true $\sigma_{ij} = \sigma(x_j, f_i)$. We call the matrix $\widetilde{\mathbf{\Sigma}} = (\widetilde{\sigma}_{ij})$ the "formal" signature matrix. Also, let $(\widetilde{\mathbf{c}}; \widetilde{\mathbf{d}})$ be any valid offset pair for $\widetilde{\mathbf{\Sigma}}$, and let $\widetilde{\mathbf{J}}$ be the resulting Jacobian defined by (2.6) with $\widetilde{\mathbf{\Sigma}}$ and $(\widetilde{\mathbf{c}}; \widetilde{\mathbf{d}})$.

If some $\widetilde{\sigma}_{ij} > \sigma_{ij}$, then hidden symbolic cancellations happen in f_i and f_i does not truly depend on $x_j^{(\widetilde{\sigma}_{ij})}$. Then $\widetilde{\mathbf{J}}_{ij} \equiv 0$, and (i, j) is a structural zero in $\widetilde{\mathbf{J}}$. Due to such cancellations, $\widetilde{\mathbf{J}}$ has more structural zeros than $\mathbf{J}$ does, so $\widetilde{\mathbf{J}}$ is more likely to be structurally singular. It is also possible that the DAE itself is structurally ill posed.

Since $\widetilde{\sigma}_{ij} \geq \sigma_{ij}$ for all $i, j = 1\!:\!n$, we can write $\widetilde{\mathbf{\Sigma}} \geq \mathbf{\Sigma}$ meaning "elementwise greater or equal".

Recall the essential sparsity pattern $\mathbf{S}_{\text{ess}}$ of $\mathbf{\Sigma}$ in (2.13). This set is the union of all (i, j) positions that lie on any HVT. We give two theorems below, which are Theorems 5.1 and 5.2 in [34].

Theorem 3.15 *Suppose that a valid offset pair* $(\mathbf{c}; \mathbf{d})$ *of* $\mathbf{\Sigma}$ *gives a nonsingular* $\mathbf{J}$ *as defined by (2.6) at some consistent point. Then every valid offset pair* $(\mathbf{c}; \mathbf{d})$ *gives a nonsingular* $\overline{\mathbf{J}}$ *(not necessarily the same as* $\mathbf{J}$*) at this point. All resulting* $\overline{\mathbf{J}}$*, including* $\mathbf{J}$*, are equal on* $\mathbf{S}_{ess}$*, and all have the same determinant* $\det(\overline{\mathbf{J}}) = \det(\mathbf{J})$.

By "equal on $\mathbf{S}_{\text{ess}}$" we mean $\widetilde{J}_{ij} = J_{ij}$ for all $(i,j) \in \mathbf{S}_{\text{ess}}$.

Theorem 3.16 *Assume that* $\mathbf{J}$*, derived from* $\boldsymbol{\Sigma}$ *and a valid offset pair* $(\mathbf{c};\mathbf{d})$*, is generically nonsingular. Let* $(\widetilde{\mathbf{c}};\widetilde{\mathbf{d}})$ *be a valid offset pair of the formal signature matrix* $\widetilde{\boldsymbol{\Sigma}}$*, and let* $\widetilde{\mathbf{J}}$ *be the Jacobian derived from* $\widetilde{\boldsymbol{\Sigma}}$ *and* $(\widetilde{\mathbf{c}};\widetilde{\mathbf{d}})$*. In exact arithmetic, one of the following two alternatives must occur:*

(i) $\text{Val}(\widetilde{\boldsymbol{\Sigma}}) = \text{Val}(\boldsymbol{\Sigma})$*. Then every HVT of* $\boldsymbol{\Sigma}$ *is an HVT of* $\widetilde{\boldsymbol{\Sigma}}$*, and* $(\widetilde{\mathbf{c}};\widetilde{\mathbf{d}})$ *is valid for* $\boldsymbol{\Sigma}$*. Consequently,* $\widetilde{\mathbf{J}}$ *is also generically nonsingular.*

(ii) $\text{Val}(\widetilde{\boldsymbol{\Sigma}}) > \text{Val}(\boldsymbol{\Sigma})$*. Then* $\widetilde{\mathbf{J}}$ *is structurally singular.*

We give explanations for each of these two cases.

(i) If $\widetilde{\boldsymbol{\Sigma}} \geq \boldsymbol{\Sigma}$ and $\text{Val}(\widetilde{\boldsymbol{\Sigma}}) = \text{Val}(\boldsymbol{\Sigma})$, then overestimating some σ_{ij} does not pose a danger to SA's success. In this case, SA uses a valid, but not necessarily canonical, offset pair $(\mathbf{c};\mathbf{d})$ of the true $\boldsymbol{\Sigma}$. As a consequence, we would treat some identically zero entries of $\mathbf{J}$ as nonzeros; this may make the solution scheme slightly less efficient; see [34, Examples 5.1 and 5.2].

(ii) If $\widetilde{\boldsymbol{\Sigma}} \geq \boldsymbol{\Sigma}$ and $\text{Val}(\widetilde{\boldsymbol{\Sigma}}) > \text{Val}(\boldsymbol{\Sigma})$, then $\widetilde{\mathbf{J}}$ is guaranteed to be structurally singular. Since $\mathbf{J}$, derived from $\boldsymbol{\Sigma}$ and $(\mathbf{c};\mathbf{d})$ is generically singular, the singularity of $\widetilde{\mathbf{J}}$ should attribute to the overestimations of some σ_{ij}.

Fortunately, modern modeling environments usually perform simplifications on a problem formulation [6, 24, 53]. These simplifications may hopefully convert Case *(ii)* to Case *(i)*, and hence reduce the occurrence of a structurally singular $\mathbf{J}$ when SA is applied.

Example 3.17 Consider

$$\begin{aligned} 0 &= f_1 = (xy)' - x'y - xy' + 2x + y - 3 \\ 0 &= f_2 = x + y - 2. \end{aligned} \tag{3.9}$$

$$\widetilde{\mathbf{\Sigma}} = \begin{array}{c} \\ f_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1^{\bullet} & 1 \\ 0 & 0^{\bullet} \end{bmatrix} \\ \begin{array}{cc} 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ \text{Val}(\widetilde{\mathbf{\Sigma}}) = 1 \end{array} \qquad \widetilde{\mathbf{J}} = \begin{array}{c} \\ f_1 \\ f_2' \end{array} \begin{array}{c} \begin{array}{cc} x' & y' \end{array} \\ \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \det(\widetilde{\mathbf{J}}) = 0$$

Here, the signature matrix and Jacobian are the formal ones. Since $\det(\widetilde{\mathbf{J}}) = 0$, SA fails. Simplifying f_1 to $f_1 = 2x + y - 3$ reveals that (3.9) is a simple linear algebraic system:

$$\begin{aligned} 0 &= f_1 = 2x + y - 3 \\ 0 &= f_2 = x + y - 2. \end{aligned}$$

$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 0^{\bullet} & 0 \\ 0 & 0^{\bullet} \end{bmatrix} \\ \begin{array}{cc} 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ \text{Val}(\mathbf{\Sigma}) = 0 \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \quad \det(\mathbf{J}) = 1$$

Hereafter we shall focus on another kind of failure of SA. In this case, no σ_{ij} is overestimated, and $\mathbf{J}$ is identically singular but structurally nonsingular. Examples 3.18 and 3.19 illustrate this case.

Example 3.18 Consider the linear constant coefficient DAE[1] from [52] :

$$\begin{aligned} 0 = f_1 &= -x_1' + x_3 + b_1(t) \\ 0 = f_2 &= -x_2' + x_4 + b_2(t) \\ 0 = f_3 &= \quad x_2 + x_3 + x_4 + c_1(t) \\ 0 = f_4 &= -x_1 + x_3 + x_4 + c_2(t). \end{aligned} \tag{3.10}$$

$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1^{\bullet} & & 0 & \\ & 1^{\bullet} & & 0 \\ & 0 & 0^{\bullet} & 0 \\ 0 & & 0 & 0^{\bullet} \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathrm{Val}(\mathbf{\Sigma}) = 2 \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ \\ \det(\mathbf{J}) \equiv 0 \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ & -1 & & 1 \\ & & 1 & 1 \\ & & 1 & 1 \end{bmatrix} \end{array}$$

This DAE is of differentiation index 3 [52], while SA finds structural index 1 and singular $\mathbf{J}$. Hence SA fails. □

Example 3.19 In the following DAE, SA reports structural index 2, which equals the differentiation index. However, $\mathbf{J}$ is identically singular.

[1] We consider this DAE with parameters $\beta = \epsilon = 1$, $\alpha_1 = \alpha_2 = \delta = 1$, and $\gamma = -1$. In [52] superscripts are used as indices, while we use subscripts instead. We also change the (original) equation names g_1, g_2 to f_3, f_4, and the (original) variable names y_1, y_2 to x_3, x_4.

$$
\begin{aligned}
0 = f_1 &= -x_1' - x_3' + x_1 + x_2 + g_1(t) \\
0 = f_2 &= -x_2' - x_3' + x_1 + x_2 + x_3 + x_4 + g_2(t) \\
0 = f_3 &= \quad x_2 + x_3 + g_3(t) \\
0 = f_4 &= \quad x_1 - x_4 + g_4(t)
\end{aligned}
\tag{3.11}
$$

$$
\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ d_j \end{array}
\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
1^{\bullet} & 0 & 1 & \\
0 & 1^{\bullet} & 1 & 0 \\
 & 0 & 0^{\bullet} & \\
0 & & & 0^{\bullet}
\end{bmatrix} \\
\begin{array}{cccc} 1 & 1 & 1 & 0 \end{array}
\end{array}
\begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 0 \\ \mathrm{Val}(\mathbf{\Sigma}) = 2 \end{array}
\qquad
\mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3' \\ f_4 \end{array}
\begin{array}{c}
\begin{array}{cccc} x_1' & x_2' & x_3' & x_4 \end{array} \\
\begin{bmatrix}
-1 & & -1 & \\
 & -1 & -1 & 1 \\
 & -1 & -1 & \\
 & & & 1
\end{bmatrix} \\
\det(\mathbf{J}) \equiv 0
\end{array}
$$

Using the solution scheme derived from the SA result, we would try to solve at stage $k = 0$ the linear system $0 = f_1, f_2, f_3', f_4$ for x_1', x_2', x_3', x_4, where the matrix is $\mathbf{J}$. Since it is singular, the solution scheme fails in solving (3.11) at this stage; see Table 3.1.

stage k	solve	for	using	comment
-1	f_3	x_2, x_3	$-$	initialize x_1
0	f_1, f_2, f_3', f_4	x_1', x_2', x_3', x_4	x_1, x_2, x_3	singular $\mathbf{J}$; solution scheme fails

Table 3.1: Solution scheme for (3.11).

Now we replace f_2 by $\overline{f}_2 = f_2 + f_3'$ to obtain

$$\begin{aligned} 0 = f_1 &= -x_1' - x_3' + x_1 + x_2 + g_1(t) \\ 0 = \overline{f}_2 &= x_1 + x_2 + x_3 + x_4 + g_2(t) + g_3'(t) \\ 0 = f_3 &= x_2 + x_3 + g_3(t) \\ 0 = f_4 &= x_1 - x_4 + g_4(t). \end{aligned} \tag{3.12}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ f_1 \\ \overline{f}_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1^{\bullet} & \\ 0^{\bullet} & 0 & 0 & 0 \\ & 0^{\bullet} & 0 & \\ 0 & & & 0^{\bullet} \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ 1 \\ 1 \\ \end{array} \quad \text{Val}(\overline{\mathbf{\Sigma}}) = 1$$

$$\overline{\mathbf{J}} = \begin{array}{c} \\ f_1 \\ \overline{f}_2' \\ f_3' \\ f_4' \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3' & x_4' \end{array} \\ \begin{bmatrix} -1 & & -1 & \\ 1 & 1 & 1 & 1 \\ & 1 & 1 & \\ 1 & & & -1 \end{bmatrix} \end{array} \quad \det(\overline{\mathbf{J}}) = 2$$

The solution scheme succeeds; see Table 3.2. The resulting DAE (3.12) is of structural index $\nu_S = 1$, which equals the differentiation index.

stage k	solve	for	using	comment
-1	$\overline{f}_2, f_3, f_4$	x_1, x_2, x_3, x_4	–	–
0	$f_1, \overline{f}_2', f_3', f_4'$	x_1', x_2', x_3', x_4'	x_1, x_2, x_3, x_4	nonsingular $\mathbf{J}$; solution scheme succeeds

Table 3.2: Solution scheme for (3.12).

At stage $k = 0$, we solve $0 = f_1, \overline{f}_2', f_3', f_4'$ for x_1', x_2', x_3', x_4' using x_1, x_2, x_3, x_4. Since $\overline{f}_2' = f_2' + f_3''$, we need f_3'' to find these first-order derivatives. Therefore, the original DAE (3.11) is of differentiation index 2.

Note that by setting $f_2 = \overline{f}_2 - f_3'$ we can recover the original system. It can be easily verified that a vector function

$$\mathbf{x}(t) = \big[x_1(t),\, x_2(t),\, x_3(t),\, x_4(t)\big]^T$$

that satisfies (3.12) also satisfies (3.11), and vice versa. We explain in §4.1 how this conversion makes SA succeed. □

In Examples 3.18 and 3.19, $\mathbf{J}$ is identically singular but structurally nonsingular. No symbolic cancellation occurs in the equations therein. Therefore, this kind of failure is more difficult to detect and remedy.

From our experience, we conjecture that a decrease in the value of a signature matrix can lead to a better DAE formulation from SA's perspective. Our techniques, the *conversion methods*, are aimed at achieving such a decrease. We describe them in the upcoming chapters. Provided some conditions are satisfied, these methods convert a structurally singular DAE into an equivalent DAE on which SA is more likely to produce a nonsingular System Jacobian and hence succeed. By "equivalent" we mean that the original DAE and the converted one have (at least locally) the same solution (if any). We shall also elaborate on this equivalence issue.

Chapter 4

Basic conversion methods

In this chapter, we present two conversion methods. They attempt to fix SA's failures systematically by reducing the value of a signature matrix. The first method is based on replacing an equation by a linear combination of some existing 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 replacing derivatives by a linear combination of other derivatives and newly introduced variables. After these replacements, also referred to as *expression substitutions*, we append new equations that define the new variables, so the resulting DAE is an enlarged one. We call this method the expression substitution (ES) method and describe it in §4.2.

Given a DAE (1.1), we assume that it has a finite Val($\mathbf{\Sigma}$) and an identically (but not structurally) singular System Jacobian $\mathbf{J}$. We still assume that the equations in (1.1) are sufficiently differentiable, so that our conversion methods fit into the Σ-method theory.

After a conversion, we denote the corresponding signature matrix as $\overline{\mathbf{\Sigma}}$ and System Jacobian as $\overline{\mathbf{J}}$. If Val($\overline{\mathbf{\Sigma}}$) 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 ends up with a solvable DAE with $\mathrm{Val}(\overline{\mathbf{\Sigma}}) \geq 0$ and a generically nonsingular $\overline{\mathbf{J}}$. Given the fact that each conversion reduces the value of a 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, $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = -\infty$, 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 use u_i to mean the ith component of $\mathbf{u}$.

Remark 4.1 We give several remarks about $\mathbf{u}$.

- For our exposition, we regard the x_j's and derivatives of them as symbols instead of functions of t. Therefore, we view $\mathbf{J}$ and $\mathbf{u}$ as functions of t, the x_j's, and derivatives of them.

- We assume that entries in $\mathbf{u}$ do not share a common multiplier. For instance, if $\mathbf{u} = [0, 0, 1, -1]^T \in \mathrm{coker}(\mathbf{J})$, then we shall not use, for example, $\mathbf{u} = [0, 0, x_1', -x_1']^T \in \mathrm{coker}(\mathbf{J})$.

- We avoid "unnecessary" fractions in the entries of u. For example, if $\mathbf{u} = [0, 0, x_1', x_1^{-1}]^T \in \mathrm{coker}(\mathbf{J})$, we shall use $\mathbf{u} = [0, 0, x_1x_1', 1]^T$.

Denote[1]

$$I = \{\, i \mid u_i \not\equiv 0 \,\}, \quad \underline{c} = \min_{i \in I} c_i, \quad \text{and} \quad L = \big\{\; l \in I \mid\; c_l = \underline{c} \,\big\}. \tag{4.1}$$

Here, I is the set of indices for which the ith component of $\mathbf{u}$ is not identically zero, and obviously $|I| \geq 2$; L a subset of I such that $f_l^{(c_l)}$ for $l \in L$ is a least differentiated equation among the $f_i^{(c_i)}$ for $i \in I$. From (4.1), there exists at least one $l \in I$ such that $c_l = \underline{c}$, so $L \neq \emptyset$.

We prove two preliminary lemmas before Theorem 4.4, on which the LC method is based.

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

$$\sigma\left(x_j, \mathbf{u}\right) < d_j - \underline{c} \quad \textit{for all } j = 1 \colon n, \tag{4.2}$$

then

$$\sigma\left(x_j, \overline{f}\right) < d_j - \underline{c}, \quad \textit{for all } j = 1 \colon n, \tag{4.3}$$

where

$$\overline{f} = \sum_{i \in I} u_i f_i^{(c_i - \underline{c})}. \tag{4.4}$$

Proof. By (4.1), $c_i - \underline{c} \geq 0$ for all $i \in I$. By (2.2), $\sigma\left(x_j, f_i\right) = \sigma_{ij} \leq d_j - c_i$. Applying

[1] Although I, $\underline{c}$, and L depend on $\mathbf{u}$, we omit the argument $\mathbf{u}$ to simplify notation.

Griewank's Lemma (2.7) to (2.6) with $w = f_i$ and $q = c_i - \underline{c}$ yields

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j^{(d_j - c_i)}} = \frac{\partial f_i^{(c_i - \underline{c})}}{\partial x_j^{(d_j - c_i + c_i - \underline{c})}} = \frac{\partial f_i^{(c_i - \underline{c})}}{\partial x_j^{(d_j - \underline{c})}} \qquad \text{for } i \in I \text{ and all } j = 1\!:\!n. \quad (4.5)$$

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

$$\begin{aligned} \frac{\partial \overline{f}}{\partial x_j^{(d_j - \underline{c})}} &= \frac{\partial \left(\sum_{i \in I} u_i f_i^{(c_i - \underline{c})} \right)}{\partial x_j^{(d_j - \underline{c})}} && \text{by the definition of } \overline{f} \text{ in (4.4)} \\ &= \sum_{i \in I} u_i \frac{\partial f_i^{(c_i - \underline{c})}}{\partial x_j^{(d_j - \underline{c})}} = \sum_{i \in I} u_i \mathbf{J}_{ij} && \text{by (4.2) and then (4.5)} \\ &= (\mathbf{J}^T \mathbf{u})_j = 0 && \text{since } \mathbf{u} \in \operatorname{coker}(\mathbf{J}) \ . \end{aligned}$$

Hence $\overline{f}$ depends on $x_j^{(< d_j - \underline{c})}$ only, for all j—this results in the inequality in (4.3). □

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

Proof. Since $\overline{\sigma}_{lj} < d_j - c_l$ for all j, the intersection of a HVT $\overline{T}$ of $\overline{\mathbf{\Sigma}}$ with row l is a position (l, r) with $\overline{\sigma}_{lr} < d_r - c_l$. Then

$$\operatorname{Val}(\overline{\mathbf{\Sigma}}) = \sum_{(i,j) \in T} \overline{\sigma}_{ij} = \overline{\sigma}_{lr} + \sum_{(i,j) \in T \setminus \{ (l,r) \}} \sigma_{ij} < \sum_j d_j - \sum_i c_i = \operatorname{Val}(\mathbf{\Sigma}). \qquad \square$$

The LC method is based on the following theorem.

Theorem 4.4 *Let I, $\underline{c}$, and L be as defined in (4.1). If we replace an equation f_l, $l \in L$, by $\overline{f}$ in (4.4), then* $\operatorname{Val}(\overline{\mathbf{\Sigma}}) < \operatorname{Val}(\mathbf{\Sigma})$*, where $\overline{\mathbf{\Sigma}}$ is the signature matrix of the resulting DAE.*

Proof. By Lemma 4.2, such a replacement results in $\overline{\sigma}_{lj} = \sigma\left(x_j, \overline{f}_l\right) < d_j - c_l$ for all $j = 1:n$. Immediate from Lemma 4.3 is $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$. □

Usually we write $\overline{f}$ as $\overline{f}_l$ in the resulting DAE, and call the conversion procedure in Theorem 4.4 an *LC conversion*.

The inequality in (4.2) is referred to as the *LC condition*, which is merely *sufficient*: if we allow the "<" in (4.2) to be "=" for some i, then we have only $\mathrm{Val}(\overline{\mathbf{\Sigma}}) \le \mathrm{Val}(\mathbf{\Sigma})$, while the strict "<" is not guaranteed. See Example 4.16 for the $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma})$ case and the example in §6.1 for the $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ case.

Example 4.5 We illustrate this method with the following (artificial) example:

$$\begin{aligned} 0 = f_1 &= -x_1' + x_3 \\ 0 = f_2 &= -x_2' + x_4 \\ 0 = f_3 &= \quad F(x_1, x_2) \\ 0 = f_4 &= \quad x_3 F_{x_1}(x_1, x_2) + x_4 F_{x_2}(x_1, x_2) + G(x_1, x_2). \end{aligned} \tag{4.6}$$

Here, variables x_1, x_2 occur in $F(x_1, x_2)$ and $G(x_1, x_2)$. The notation $F_{x_1}(x_1, x_2)$ means the partial derivative $\partial F(x_1, x_2)/\partial x_1$, and we write similarly $F_{x_2}(x_1, x_2)$, $G_{x_1}(x_1, x_2)$, and $G_{x_2}(x_1, x_2)$.

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1^{\bullet} & & 0 & \\ & 1 & & 0^{\bullet} \\ 0 & 0^{\bullet} & & \\ 0 & 0 & 0^{\bullet} & 0 \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 0 \\ \mathrm{Val}(\boldsymbol{\Sigma}) = 1 \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3' \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ & -1 & & 1 \\ F_{x_1} & F_{x_2} & & \\ & & F_{x_1} & F_{x_2} \end{bmatrix} \\ \det(\mathbf{J}) \equiv 0 \end{array}$$

Because of the identically singular $\mathbf{J}$, the SA fails. It reports structural index 2, but the DAE (4.6) is of differentiation index 3. We choose $\mathbf{u} = [F_{x_1},\, F_{x_2},\, 1,\, -1]^T \in \mathrm{coker}(\mathbf{J})$ and illustrate (4.1):

$$\begin{aligned} I &= \{\, i \mid u_i \not\equiv 0 \,\} = \{\, 1, 2, 3, 4 \,\}, \\ \underline{c} &= \min_{i \in I} c_i = 0, \\ L &= \{\, l \in I \mid c_l = \underline{c} = 0 \,\} = \{\, 1, 2, 4 \,\}. \end{aligned}$$

Then we check the LC condition (4.2):

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

Hence $\sigma(x_j, \mathbf{u}) < d_j - \underline{c}$ for all j and the LC condition holds.

Using (4.4) gives

$$\begin{aligned}\overline{f} &= \sum_{i\in I} u_i f_i^{(c_i-\underline{c})} = \sum_{i\in I} u_i f_i^{(c_i)} = F_{x_1} f_1 + F_{x_2} f_2 + f_3' - f_4 \\ &= F_{x_1}\cdot(-x_1' + x_3) + F_{x_2}\cdot(-x_2' + x_4) + F' - (x_3 F_{x_1} + x_4 F_{x_2} + G) \\ &= -x_1' F_{x_1} + x_3 F_{x_1} - x_2' F_{x_2} + x_4 F_{x_2} + x_1' F_{x_1} + x_2' F_{x_2} - x_3 F_{x_1} - x_4 F_{x_2} - G \\ &= -G.\end{aligned}$$

For each $l \in L = \{1, 2, 4\}$, assuming $u_l \neq 0$, we can replace f_l by $\overline{f}_l = \overline{f}$. We show in the following the three possible converted DAEs, each with $\mathrm{Val}(\overline{\boldsymbol{\Sigma}}) = 0$ and a generically nonsingular $\overline{\mathbf{J}}$.

- $l = 1$:

$$\begin{aligned} 0 &= \overline{f}_1 = -G(x_1, x_2) \\ 0 &= f_2 = -x_2' + x_4 \\ 0 &= f_3 = F(x_1, x_2) \\ 0 &= f_4 = x_3 F_{x_1}(x_1, x_2) + x_4 F_{x_2}(x_1, x_2) + G(x_1, x_2) \end{aligned} \tag{4.7}$$

$$\overline{\boldsymbol{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0^{\bullet} & 0 & & \\ & 1 & & 0^{\bullet} \\ 0 & 0^{\bullet} & & \\ 0 & 0 & 0^{\bullet} & 0 \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 1 \\ 0 \\ 1 \\ 0 \\ \mathrm{Val}(\overline{\boldsymbol{\Sigma}}) = 0 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}_1' \\ f_2 \\ f_3' \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} -G_{x_1} & -G_{x_2} & & \\ & -1 & & 1 \\ F_{x_1} & F_{x_2} & & \\ & & F_{x_1} & F_{x_2} \end{bmatrix} \end{array}$$

$$\det(\overline{\mathbf{J}}) = F_{x_1}(F_{x_1} G_{x_2} - F_{x_2} G_{x_1})$$

When $u_1 = F_{x_1} \neq 0$ and $F_{x_1}G_{x_2} \neq F_{x_2}G_{x_1}$, the determinant is nonzero and the SA succeeds.

- $l = 2$:

$$\begin{aligned} 0 &= f_1 = -x_1' + x_3 \\ 0 &= \overline{f}_2 = -G(x_1, x_2) \\ 0 &= f_3 = F(x_1, x_2) \\ 0 &= f_4 = x_3 F_{x_1}(x_1, x_2) + x_4 F_{x_2}(x_1, x_2) + G(x_1, x_2) \end{aligned} \tag{4.8}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ f_1 \\ \overline{f}_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1 & & 0^{\bullet} & \\ 0^{\bullet} & 0 & & \\ 0 & 0^{\bullet} & & \\ 0 & 0 & 0 & 0^{\bullet} \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ 1 \\ 0 \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 0 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ f_1 \\ \overline{f}_2' \\ f_3' \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ -G_{x_1} & -G_{x_2} & & \\ F_{x_1} & F_{x_2} & & \\ & & F_{x_1} & F_{x_2} \end{bmatrix} \end{array} \quad \det(\overline{\mathbf{J}}) = F_{x_2}(F_{x_1}G_{x_2} - F_{x_2}G_{x_1})$$

Similarly, the SA succeeds when $u_2 = F_{x_2} \neq 0$ and $F_{x_1}G_{x_2} \neq F_{x_2}G_{x_1}$.

- $l = 4$:

$$\begin{aligned} 0 &= f_1 = -x_1' + x_3 \\ 0 &= f_2 = -x_2' + x_4 \\ 0 &= f_3 = F(x_1, x_2) \\ 0 &= \overline{f}_4 = -G(x_1, x_2) \end{aligned} \tag{4.9}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ \overline{f}_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1 & & 0^{\bullet} & \\ & 1 & & 0^{\bullet} \\ 0 & 0^{\bullet} & & \\ 0^{\bullet} & 0 & & \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 1 \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 0 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3' \\ \overline{f}_4' \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ & -1 & & 1 \\ F_{x_1} & F_{x_2} & & \\ -G_{x_1} & -G_{x_2} & & \end{bmatrix} \end{array} \quad \det(\overline{\mathbf{J}}) = -F_{x_1}G_{x_2} + F_{x_2}G_{x_1}$$

In this case, SA's success requires only $F_{x_1}G_{x_2} \neq F_{x_2}G_{x_1}$. □

Using the LC method, we obtain three converted DAEs (4.7)-(4.9). However, only (4.9) and (4.6) have exactly the same solution sets, that is, they are *always* equivalent. In the rest of this section, we address the equivalence between a converted DAE and the original DAE.

First, we give a definition for equivalent DAEs.

Definition 4.6 ***(Equivalent DAEs)*** *Let $\mathcal{F}$ and $\overline{\mathcal{F}}$ denote two DAEs. They are equivalent for all t on some interval $\mathbb{I} \subset \mathbb{R}$, if a solution of $\mathcal{F}$ is a solution to $\overline{\mathcal{F}}$ and vice versa.*

In the following context, we denote by $\mathcal{F}$ the original DAE with equations f_i and an identically singular System Jacobian $\mathbf{J}$. After an LC conversion, we obtain a (converted) DAE, denoted by $\overline{\mathcal{F}}$, with equations $\overline{f}_i$ and a System Jacobian $\overline{\mathbf{J}}$, whose non-singularity does not matter here.

Theorem 4.7 *We assume the following holds.*

(i) A DAE $\mathcal{F}$ has a finite $\mathrm{Val}(\mathbf{\Sigma})$ *and an identically singular System Jacobian* $\mathbf{J}$.

(ii) A vector $\mathbf{u} \in \operatorname{coker}(\mathbf{J})$ *is well defined for all* t *on some time interval* $\mathbb{I}$.

(iii) The LC condition (4.2) is satisfied, and we perform an LC conversion to obtain a DAE $\overline{\mathcal{F}}$.

Then DAEs $\mathcal{F}$ *and* $\overline{\mathcal{F}}$ *are equivalent, if* $u_l \neq 0$ *for all* $t \in \mathbb{I}$.

Proof. Let a solution of $\mathcal{F}$ over some interval $\mathbb{I} \subset \mathbb{R}$ be a vector-valued function

$$\mathbf{x}(t) = \big[x_1(t), \ldots, x_n(t)\big]^T.$$

Then functions in (1.1) and derivatives of them vanish for all $t \in \mathbb{I}$, or we say they "vanish on $\mathbb{I}$". Since $\mathbf{u}$ is well defined for all $t \in \mathbb{I}$ and can be evaluated by $\mathbf{x}(t)$ and derivatives of them, functions

$$\overline{f}_l = \sum_{i \in I} u_i f_i^{(c_i - \underline{c})} \quad \text{and} \quad \overline{f}_i = f_i \ \text{ for } \ i \neq l$$

and derivatives of $\overline{f}_i$ also vanish on $\mathbb{I}$. Therefore $\mathbf{x}(t)$ is a solution to $\overline{\mathcal{F}}$.

Conversely, assume that $\overline{\mathbf{x}}(t)$ is a solution of $\overline{\mathcal{F}}$ on $\mathbb{I}$. Since $\mathbf{u}$ is well defined on $\mathbb{I}$ and $u_l(t) \neq 0$ for all $t \in \mathbb{I}$,

$$f_l = \frac{1}{u_l}\left(\overline{f}_l - \sum_{i \in I \setminus \{l\}} u_i \overline{f}_i^{(c_i - \underline{c})}\right) \quad \text{and} \quad f_i = \overline{f}_i \ \text{ for } \ i \neq l$$

and derivatives of them vanish on $\mathbb{I}$. Hence $\overline{\mathbf{x}}(t)$ is a solution to $\mathcal{F}$.

By Definition 4.6, $\mathcal{F}$ and $\overline{\mathcal{F}}$ are equivalent. □

We learn from Theorem 4.7 that, it is desirable to choose a row index $l \in L$ such that u_l is an expression that *never* becomes zero. For example, u_l is a nonzero

constant, or expressions like $x_1^2 + 1$ and $2 + \cos x_2'$. Such a choice of l guarantees that the resulting DAE is *always* equivalent to the original DAE. However, in general, it is undecidable whether an expression is identically zero or not [51]. Hence, we consider a (nonzero) constant u_l as the most preferable choice among all $l \in L$, and use $\overline{L}$ to denote a set of indices l for such u_l:

$$\overline{L} = \left\{ l \in L \mid u_l \text{ is constant} \right\}. \tag{4.10}$$

Example 4.8 In Example 4.5, $\overline{L} = \left\{ 4 \right\}$. Case $l = 1$ [resp. $l = 2$] requires $F_{x_1} \neq 0$ [resp. $F_{x_2} \neq 0$] to recover the original DAE (4.6) from (4.7) [resp. from (4.8)]. However, for case $l = 4 \in \overline{L}$, $u_4 = 1$ is a nonzero constant for any t. Therefore this choice is the most desirable among the three.

Remark 4.9 We name our method the "LC method" because of the following consideration. The LC condition (4.2) says that, the vector $\mathbf{u} \in \operatorname{coker}(\mathbf{J})$ comprises *only* derivatives $x_j^{(<d_j - \underline{c})}$ for all j, while $x_j^{(d_j - \underline{c})}$ are the leading derivatives in $f_i^{(c_i - \underline{c})}$ for all $i \in I$. Therefore, by regarding each u_i as a "constant" in $\sum_{i \in I} u_i f_i^{(c_i - \underline{c})} = \overline{f}_l$, we say $\overline{f}_l$ is a "linear combination" of the equations $f_i^{(c_i - \underline{c})}$.

If $\mathbf{u}$ is a constant vector, then $\sigma(x_j, \mathbf{u}) = -\infty$ for every x_j. In this case, the condition (4.2) certainly holds, so we do not need to check it. We illustrate this in the next example.

Example 4.10 Consider

$$0 = f_1 = x_1 + tx_2 + t^2x_3 + g_1(t)$$
$$0 = f_2 = x_1' + tx_2' + t^2x_3' + g_2(t)$$
$$0 = f_3 = x_1'' + tx_2'' + 2t^2x_3'' + g_3(t).$$

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & c_i \end{array} \\ \begin{bmatrix} 0^\bullet & 0 & 0 \\ 1 & 1^\bullet & 1 \\ 2 & 2 & 2^\bullet \end{bmatrix} \begin{array}{c} 2 \\ 1 \\ 0 \end{array} \\ \begin{array}{ccc} 2 & 2 & 2 \end{array} \end{array} \quad \text{Val}(\boldsymbol{\Sigma}) = 3 \qquad \mathbf{J} = \begin{array}{c} \\ f_1'' \\ f_2' \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} x_1'' & x_2'' & x_3'' \end{array} \\ \begin{bmatrix} 1 & t & t^2 \\ 1 & t & t^2 \\ 1 & t & 2t^2 \end{bmatrix} \end{array} \quad \det(\mathbf{J}) \equiv 0$$

For $\mathbf{u} = [-1, 1, 0]^T \in \text{coker}(\mathbf{J})$, we use (4.1) and (4.10) to derive

$$I = \{\, 1, 2 \,\}, \quad \underline{c} = c_2 = 1, \quad \text{and} \quad L = \overline{L} = \{\, 2 \,\}.$$

Since $\mathbf{u}$ is a constant vector, the LC condition (4.2) is satisfied. We replace f_2 by

$$\begin{aligned} \overline{f}_2 &= u_1 f_1^{(2-1)} + u_2 f_2^{(1-1)} = -f_1' + f_2 \\ &= -\big(x_1 + tx_2 + t^2x_3 - g_1\big)' + (x_1' + tx_2' + t^2x_3' + g_2) \\ &= -x_2 - 2tx_3 - g_1' + g_2. \end{aligned}$$

The converted DAE is

$$0 = f_1 = x_1 + tx_2 + t^2x_3 + g_1$$
$$0 = \overline{f}_2 = -x_2 - 2tx_3 - g_1' + g_2$$
$$0 = f_3 = x_1'' + tx_2'' + 2t^2x_3'' + g_3.$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ f_1 \\ \overline{f}_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{bmatrix} 0^\bullet & 0 & 0 \\ & 0^\bullet & 0 \\ 2 & 2 & 2^\bullet \end{bmatrix} \\ \begin{array}{ccc} 2 & 2 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 2 \\ 2 \\ 0 \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 2 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ f_1'' \\ \overline{f}_2'' \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} x_1'' & x_2'' & x_3'' \end{array} \\ \begin{bmatrix} 1 & t & t^2 \\ & -1 & -2t \\ 1 & t & 2t^2 \end{bmatrix} \\ \det(\mathbf{J}) = -t^2 \end{array}$$

If $t \neq 0$ and its magnitude is not too small compared to the scale of the derivatives, then $\mathbf{J}$ is computably nonsingular.

Below we define an ill-posed DAE using the structural posedness defined in the DAESA papers [37, 46]; see also Definition 2.1.

Definition 4.11 ***(Well-posedness of a DAE)*** *A DAE is ill posed if it has an equivalent DAE that is structurally ill posed (SIP); otherwise it is well posed.*

Example 4.12 Consider problem (3.2). Using $0 = f_3 = x^2 + y^2 - \ell^2$, we reduce f_1 to $\overline{f}_1$, a trivial equation $0 = 0$. This is simply performing a simple substitution, and is not applying the LC method. The signature matrix

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \begin{bmatrix} & & \\ & 2 & 0 \\ 0 & 0 & \end{bmatrix} \end{array} \tag{4.11}$$

does not have a finite HVT, so the resulting DAE is SIP. Hence, by Definition 4.11 , the original SWP DAE (3.2) is ill posed.

Corollary 4.13 *If we can reformulate a structurally well-posed DAE, by a conversion method, into an equivalent DAE that is structurally ill-posed, then the original DAE is ill posed.*

Proof. This follows from Theorem 4.7 and Definition 4.11. □

Example 4.14 Consider the following SWP DAE

$$\begin{aligned} 0 &= f_1 = y''' + y'\lambda + y\lambda' \\ 0 &= f_2 = y'' + y\lambda - G \\ 0 &= f_3 = x^2 + y^2 - \ell^2. \end{aligned} \tag{4.12}$$

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \begin{bmatrix} & 3 & 1^{\bullet} \\ & 2^{\bullet} & 0 \\ 0^{\bullet} & 0 & \end{bmatrix} \\ \begin{array}{ccc} 0 & 3 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ 0 \\ \end{array} \qquad \mathrm{Val}(\boldsymbol{\Sigma}) = 3$$

$$\mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2' \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} x & y''' & \lambda' \end{array} \\ \begin{bmatrix} & 1 & y \\ & 1 & y \\ 2x & & \end{bmatrix} \end{array} \qquad \det(\mathbf{J}) \equiv 0$$

For $\mathbf{u} = [1, -1, 0]^T$, $\mathbf{J}^T\mathbf{u} = 0$. Using (4.1) and (4.10) gives

$$I = \{\, 1, 2 \,\}, \quad \underline{c} = c_1 = 0, \quad \text{and} \quad L = \overline{L} = \{\, 1 \,\}.$$

Since $\mathbf{u}$ is a constant vector, the LC condition (4.2) is satisfied. We replace f_1 by

$$\overline{f}_1 = f_1 - f_2' = (y''' + y'\lambda + y\lambda') - (y'' + y\lambda - G)' = 0.$$

The signature matrix of the resulting problem is exactly (4.11). Hence, by Corollary 4.13, DAE (4.12) is ill posed. □

Example 4.15 We construct the following (artificial) DAE from the pendulum DAE (2.10):

$$\begin{aligned} 0 = A &= f_1' + f_3 = x^2 + y^2 - \ell^2 + (x'' + x\lambda)' \\ 0 = B &= f_1 + A'' = f_1 + (f_1' + f_3)'' \\ &= x'' + x\lambda + \left(x^2 + y^2 - \ell^2 + (x'' + x\lambda)'\right)'' \\ 0 = C &= f_2 + A''' = f_2 + (f_1' + f_3)''' \\ &= y'' + y\lambda - G + \left(x^2 + y^2 - \ell^2 + (x'' + x\lambda)'\right)'''. \end{aligned} \tag{4.13}$$

$$\boldsymbol{\Sigma}^0 = \begin{array}{c} \\ A \\ B \\ C \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \begin{bmatrix} 3^\bullet & 0 & 1 \\ 5 & 2^\bullet & 3 \\ 6 & 3 & 4^\bullet \end{bmatrix} \\ \begin{array}{ccc} 6 & 3 & 4 \end{array} \end{array} \begin{array}{c} c_i \\ 3 \\ 1 \\ 0 \\ \mathrm{Val}(\boldsymbol{\Sigma}^0) = 9 \end{array} \qquad \mathbf{J}^0 = \begin{array}{c} \\ A''' \\ B' \\ C \end{array} \begin{array}{c} \begin{array}{ccc} x^{(6)} & y''' & \lambda^{(4)} \end{array} \\ \begin{bmatrix} 1 & 2y & x \\ 1 & 2y & x \\ 1 & 2y & x \end{bmatrix} \\ \det(\mathbf{J}^0) \equiv 0 \end{array}$$

(A superscript denotes an iteration number, not a power.) We show how to recover the simple pendulum problem.

We find $\mathbf{u} = [-1, 1, 0]^T \in \mathrm{coker}(\mathbf{J}^0)$. Then by (4.1), $I = \{1, 2\}$, $\underline{c} = 1$, and $L = \overline{L} = \{2\}$. We replace the second equation B by

$$-A^{(3-1)} + B = -A'' + (A'' + f_1) = f_1 = x'' + x\lambda.$$

The converted DAE is

$$\begin{aligned} 0 = A &= x^2 + y^2 - \ell^2 + (x'' + x\lambda)' \\ 0 = f_1 &= x'' + x\lambda \\ 0 = C &= y'' + y\lambda - G + \left(x^2 + y^2 - \ell^2 + (x'' + x\lambda)'\right)'''. \end{aligned}$$

$$\boldsymbol{\Sigma}^1 = \begin{array}{c} \\ A \\ f_1 \\ C \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \begin{bmatrix} 3^\bullet & 0 & 1 \\ 2 & & 0^\bullet \\ 6 & 3^\bullet & 4 \end{bmatrix} \\ \begin{array}{ccc} 6 & 3 & 4 \end{array} \end{array} \begin{array}{c} c_i \\ 3 \\ 4 \\ 0 \\ \mathrm{Val}(\boldsymbol{\Sigma}^1) = 6 \end{array} \qquad \mathbf{J}^1 = \begin{array}{c} \\ A''' \\ f_1^{(4)} \\ C \end{array} \begin{array}{c} \begin{array}{ccc} x^{(6)} & y''' & \lambda^{(4)} \end{array} \\ \begin{bmatrix} 1 & 2y & x \\ 1 & & x \\ 1 & 2y & x \end{bmatrix} \\ \det(\mathbf{J}^1) \equiv 0 \end{array}$$

Although $\mathrm{Val}(\mathbf{\Sigma}^1) = 6 < 9 = \mathrm{Val}(\mathbf{\Sigma}^0)$, the System Jacobian $\mathbf{J}^1$ is still singular, so we attempt the LC method again.

Choosing $\mathbf{u} = [-1, 0, 1]^T \in \mathrm{coker}(\mathbf{J}^1)$ gives

$$I = \{\, 1, 3 \,\}, \quad \underline{c} = 0, \quad \text{and} \quad L = \overline{L} = \{\, 3 \,\}.$$

We replace the third equation C by

$$-A^{(3-0)} + C = -A''' + (f_2 + A''') = f_2 = y'' + y\lambda - G.$$

The converted DAE is

$$\begin{aligned} 0 = A &= x^2 + y^2 - \ell^2 + (x'' + x\lambda)' \\ 0 = f_1 &= x'' + x\lambda \\ 0 = f_2 &= y'' + y\lambda - G. \end{aligned}$$

$$\mathbf{\Sigma}^2 = \begin{array}{c} \\ A \\ f_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x & y & \lambda & c_i \end{array} \\ \begin{bmatrix} 3^\bullet & \boxed{0} & 1 \\ 2 & & 0^\bullet \\ & 2^\bullet & \boxed{0} \end{bmatrix} \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \\ \begin{array}{ccc} 3 & 2 & 1 \end{array} \end{array} \quad \mathrm{Val}(\mathbf{\Sigma}^2) = 5 \qquad \mathbf{J}^2 = \begin{array}{c} \\ A \\ f_1' \\ f_2 \end{array} \begin{array}{c} \begin{array}{ccc} x''' & y'' & \lambda' \end{array} \\ \begin{bmatrix} 1 & & x \\ 1 & & x \\ & 1 & \end{bmatrix} \end{array}.$$

$$\det(\mathbf{J}^2) \equiv 0$$

We have $\mathrm{Val}(\mathbf{\Sigma}^2) = 5 < 6 = \mathrm{Val}(\mathbf{\Sigma}^1)$, but the System Jacobian $\mathbf{J}^2$ is still singular.

Attempting again the LC methods, we find $\mathbf{u} = [1, -1, 0]^T \in \operatorname{coker}(\mathbf{J}^2)$. Then

$$I = \{1, 2\}, \quad \underline{c} = 0, \quad \text{and} \quad L = \overline{L} = \{\, 1 \,\}.$$

Replacing the first equation A by

$$A - f_1' = (f_3 + f_1') - f_1' = f_3 = x^2 + y^2 - \ell^2,$$

we recover f_1, f_2, f_3 from (4.13). The resulting solvable DAE is exactly the pendulum DAE (2.10), with $\operatorname{Val}(\mathbf{\Sigma}) = 2$ and $\det(\mathbf{J}) = -2\ell^2$; cf. Example 2.5.

Since each $\mathbf{u}$ is a constant vector in each of the iterations, each u_l we pick is a nonzero constant. The DAE (4.13) and the pendulum DAE (2.10) are equivalent. Hence, we can solve (4.13) by simply solving (2.10). □

We summarize the steps of an LC conversion.

1) Obtain a symbolic form of $\mathbf{J}$.

2) Compute a vector $\mathbf{u} \in \operatorname{coker}(\mathbf{J})$.

3) Derive I, $\underline{c}$, and L as defined in (4.1).

4) Check the LC condition (4.2). If it is not satisfied, then the LC method is not applicable and we set $L \leftarrow \emptyset$; otherwise proceed to the next step.

5) Derive $\overline{L} \leftarrow \{\, l \in L \mid u_l \text{ is constant} \,\}$. If $\overline{L} \neq \emptyset$, then we choose an $l \in \overline{L}$; otherwise we choose an $l \in L$.

6) Perform an LC conversion: replace f_l by $\overline{f}_l = \overline{f}$ as defined in (4.4).

The sets L and $\overline{L}$ are used to decide the desirable conversion method; see §4.3 and Table 4.1 therein.

In the following example, we show that the LC method cannot fix the (artificially constructed) DAE (4.14). The LC condition (4.2) is not satisfied, so $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ is not guaranteed. This incapability of the LC method leads to a motivation to develop its "dual method", which is the ES method introduced in the next section.

Example 4.16 Consider

$$\begin{aligned} 0 &= f_1 = x_1 + e^{-x_1' - x_2 x_2''} + h_1(t) \\ 0 &= f_2 = x_1 + x_2 x_2' + x_2^2 + h_2(t). \end{aligned} \tag{4.14}$$

$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 1^{\bullet} & 2 \\ 0 & 1^{\bullet} \end{bmatrix} \\ \begin{array}{cc} 1 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ \mathrm{Val}(\mathbf{\Sigma}) = 2 \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2' \end{array} \begin{array}{c} \begin{array}{cc} x_1' & x_2'' \end{array} \\ \begin{bmatrix} -\alpha & -\alpha x_2 \\ 1 & x_2 \end{bmatrix} \\ \det(\mathbf{J}) \equiv 0 \end{array}$$

Here h_1 and h_2 are given driving functions, and $\alpha = e^{-x_1' - x_2 x_2''}$. The SA fails.

Choose $\mathbf{u} = [\alpha^{-1}, 1]^T = [e^{x_1' + x_2 x_2''}, 1]^T \in \mathrm{coker}(\mathbf{J})$. Then (4.1) and (4.10) read

$$I = \{\, 1, 2 \,\}, \quad \underline{c} = 0, \quad L = \{\, 1 \,\}, \quad \text{and} \quad \overline{L} = \emptyset.$$

Since x_1' and x_2'' occur in $\mathbf{u}$, $\sigma(x_1, \mathbf{u}) = d_1 - \underline{c}$ and $\sigma(x_2, \mathbf{u}) = d_2 - \underline{c}$ violate the LC condition (4.2). If we choose $l = 1 \in L$ and replace f_1 by

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

then the resulting DAE is $0 = \big(\overline{f}_1, f_2\big)$. Here $\beta = \alpha^{-1}(x_1 + h_1(t)) + 1$.

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \left[\begin{array}{cc} 1^{\bullet} & 2 \\ 0 & 1^{\bullet} \end{array}\right] \\ \begin{array}{cc} 1 & 2 \end{array} \end{array} \begin{array}{l} c_i \\ 0 \\ 1 \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 2 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2' \end{array} \begin{array}{c} \begin{array}{cc} x_1' & x_2'' \end{array} \\ \left[\begin{array}{cc} \beta & \beta x_2 \\ 1 & x_2 \end{array}\right] \end{array} \quad \det(\overline{\mathbf{J}}) \equiv 0$$

The SA fails still.

Since the LC condition does not hold, we return $L = \emptyset$ instead of $L = \{\, 1 \,\}$ by (4.1) to indicate the inapplicability of the LC method.

We shall show in Example 4.18 that the ES method can fix (4.14). □

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 $\mathbf{J}\mathbf{v} = \mathbf{0}$. We also consider $\mathbf{v}$ in its simplest form; see Remark 4.1 for a vector $\mathbf{u}$ in the LC method.

Denote

$$\begin{aligned} J &= \big\{\; j \;\mid\; v_j \not\equiv 0 \;\big\}, \quad s = |J|, \\ M &= \big\{\; i \;\mid\; d_j - c_i = \sigma_{ij} \text{ for some } j \in J \;\big\}, \quad \text{and} \quad \bar{c} = \max_{i \in M} c_i. \end{aligned} \tag{4.15}$$

Here, J is the set of column indices j for which the jth component of $\mathbf{v}$ is generically nonzero, and s is the number of these indices. Since $\mathbf{J}$ is identically singular, $s \geq 2$.

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

$$y_j = x_j^{(d_j-\overline{c})} - \frac{v_j}{v_l} \cdot x_l^{(d_l-\overline{c})} \qquad \text{for all } j \in J \setminus \{\, l \,\}. \tag{4.16}$$

In each f_i, we

$$\begin{array}{lll} \text{replace every} & x_j^{(\sigma_{ij})} = x_j^{(d_j-c_i)} & \text{with} \qquad j \in J \setminus \{\, l \,\} \\ \text{by} & \left(y_j + \dfrac{v_j}{v_l} \cdot x_l^{(d_l-\overline{c})}\right)^{(\overline{c}-c_i)}. & \end{array} \tag{4.17}$$

From the formula (4.15) 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 $j \in J$. The replacements use the fact that, for such an $x_j^{(\sigma_{ij})}$ with $i \in M$ and $j \in J \setminus \{\, l \,\}$,

$$x_j^{(\sigma_{ij})} = x_j^{(d_j-c_i)} = \left(x_j^{(d_j-\overline{c})}\right)^{(\overline{c}-c_i)} = \left(y_j + \frac{v_j}{v_l} \cdot x_l^{(d_l-\overline{c})}\right)^{(\overline{c}-c_i)}.$$

After these substitutions, denote each equation by $\overline{f}_i$ (for $i \notin M$, $\overline{f}_i$ and f_i are the same). Using (4.16), we introduce $s-1$ equations

$$0 = g_j = -y_j + x_j^{(d_j-\overline{c})} - \frac{v_j}{v_l} \cdot x_l^{(d_l-\overline{c})} \quad \text{for all } j \in J \setminus \{\, l \,\} \tag{4.18}$$

that define the variables y_j and prescribe the substitutions in (4.17). Appending (4.18) 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 4.17 *Let J, s, M, and $\overline{c}$ be as defined in (4.15). Assume*

$$\begin{aligned} \sigma\left(x_j, \mathbf{v}\right) &\begin{cases} < d_j - \overline{c} & \text{if } j \in J \\ \leq d_j - \overline{c} & \text{otherwise,} \end{cases} \\ d_j - \overline{c} &\geq 0 \qquad\qquad \text{for all } j \in J. \end{aligned} \tag{4.19}$$

For any $l \in J$, if we

1) introduce $s-1$ new variables y_j, $j \in J \setminus \{\, l \,\}$, as defined in (4.16),

2) perform substitutions in f_i, for all $i = 1:n$, by (4.17), and

3) append $s-1$ equations g_j, $j \in J \setminus \{\, l \,\}$, as defined in (4.18),

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

We refer to the procedure described by 1–3 in Theorem 4.17 as an *ES conversion*, and we refer to (4.19) as the *ES conditions*. These conditions are again *sufficient* for obtaining $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$.

Before proving this theorem, we give an example that illustrates the ES method.

Example 4.18 We illustrate the application of the ES method on the DAE (4.14). Suppose we choose $\mathbf{v} = [x_2, -1]^T \in \ker(\mathbf{J})$. Then (4.15) 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 ES conditions (4.19) hold:

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

We choose $l = 2 \in J$. Now $J \setminus \{ l \} = \{ 1 \}$. Using (4.16) and (4.18), we introduce for x_1 a new variable

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

and append the equation $0 = g_1 = -y_1 + x_1 + x_2 x_2'$ to the original equations. Then we replace x_1' by $(y_1 - x_2 x_2')'$ in f_1 to obtain $\overline{f}_1$, and replace x_1 by $y_1 - x_2 x_2'$ in f_2 to obtain $\overline{f}_2$. The resulting DAE and its SA results are shown below.

$$\begin{aligned}0 = \overline{f}_1 &= x_1 + e^{-y_1' + x_2'^2} + h_1(t)\\ 0 = \overline{f}_2 &= y_1 + x_2^2 + h_2(t)\\ 0 = g_1 &= -y_1 + x_1 + x_2 x_2'\end{aligned}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2 \\ g_1 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & y_1 \end{array} \\ \begin{bmatrix} 0 & 1 & 1^{\bullet} \\ & 0^{\bullet} & 0 \\ 0^{\bullet} & 1 & \boxed{0} \end{bmatrix} \\ \begin{array}{ccc} 0 & 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ 0 \\ \end{array} \quad \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1$$

$$\overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2' \\ g_1 \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2' & y_1' \end{array} \\ \begin{bmatrix} 1 & 2x_2'\gamma & -\gamma \\ & 2x_2 & 1 \\ 1 & x_2 & \end{bmatrix} \end{array} \quad \det(\overline{\mathbf{J}}) = 2\gamma(x_2 + x_2') - x_2$$

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

We prove a lemma related to Theorem 4.17, using the following assumptions for the sake of the proof.

(a) Without loss of generality, we assume 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 $J = \{1, \ldots, s\}$ by (4.15).

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

$$\widetilde{d}_j = \begin{cases} d_j & \text{if } j = 1:n \\ \overline{c} & \text{if } j = n+1:n+s, \end{cases} \qquad \widetilde{c}_i = \begin{cases} c_i & \text{if } i = 1:n \\ \overline{c} & \text{if } i = n+1:n+s, \end{cases} \tag{4.20}$$

where $\overline{c}$ *is as defined (4.15). Then the signature matrix* $\overline{\mathbf{\Sigma}}$ *of the resulting DAE from the ES method has the form in Figure 4.1.*

The proof of this lemma is rather technical, so we present it in Appendix A.1. Using Lemma 4.19, we prove Theorem 4.17.

Proof. Let $\overline{T}$ be an HVT of $\overline{\mathbf{\Sigma}}$. By Lemma 4.19,

$$\begin{aligned} \mathrm{Val}(\overline{\mathbf{\Sigma}}) &= \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} \le \sum_{(i,j)\in\overline{T}} (\widetilde{d}_j - \widetilde{c}_i) && \text{since } \widetilde{d}_j - \widetilde{c}_i \ge \overline{\sigma}_{ij} \\ &= \sum_{j=1}^{n+s} \widetilde{d}_j - \sum_{i=1}^{n+s} \widetilde{c}_i = \sum_{j=1}^{n} d_j + s\overline{c} - \sum_{i=1}^{n} c_i - s\overline{c} && \text{by (4.20)} \\ &= \sum_{j=1}^{n} d_j - \sum_{i=1}^{n} c_i = \mathrm{Val}(\mathbf{\Sigma}). \end{aligned}$$

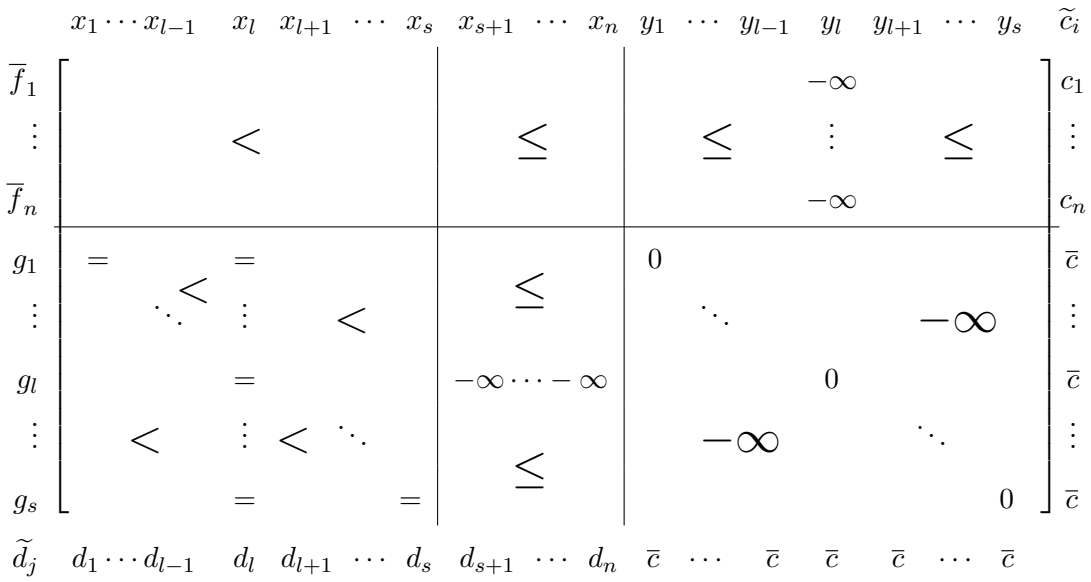


Figure 4.1: The form of $\overline{\mathbf{\Sigma}}$ for the resulting DAE from the ES method. The $<$, $\leq$, and $=$ mean the relations between $\overline{\sigma}_{ij}$ and $\widetilde{d}_j - \widetilde{c}_i$, respectively. For instance, every $\overline{\sigma}_{ij}$ whose (i, j) position is in the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

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

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

$$\widetilde{d}_j - \widetilde{c}_i = \overline{\sigma}_{ij} > -\infty \qquad \text{for all } (i, j) \in \overline{T}. \tag{4.21}$$

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 row numbers $i_1, \ldots, i_s$ can only take values among

$$1, 2, \ldots, n, n + 1, \ldots, n + l - 1, n + l + 1, \ldots, n + s.$$

Here only $s-1$ numbers are greater than n, so at least one of them is among $1:n$. In other words, there exists a position $(r,j) \in \overline{T}$ with $1 \le r \le n$ and $1 \le j \le s$ in the "$<$" region in Figure 4.1. Hence $\widetilde{d}_j - \widetilde{c}_r > \overline{\sigma}_{rj}$, which yields a contradiction of (4.21). Therefore $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$.

Finally, we remove the y_l column and its matched g_l row. The resulting signature matrix is still of value $\mathrm{Val}(\overline{\mathbf{\Sigma}})$, since $(n+l, n+l) \in \overline{T}$ and $\overline{\sigma}_{n+l,n+l} = 0$. □

Remark 4.20 We give several remarks about the ES method.

- After an ES conversion, we do symbolic simplifications on $\overline{f}_i$ for $i \in M$ and ensure that the $x_j^{(d_j-c_i)}$ for $j \in J = \{1, \ldots, s\}$ no longer occur in these equations. That is, $\sigma\left(x_j, \overline{f}_i\right) < d_j - c_i$ for $j = 1:s$ and $i \in M$.

- If some derivative $x_j^{(d_j-c_i)}$, for $i = 1:n$ and $j \in J \setminus \{\, l \,\}$, appears implicitly in an expression in f_i, then we need to write this expression into a form in which $x_j^{(d_j-c_i)}$ appears explicitly. See Example 4.21 below.

Example 4.21 Assume that f_1 contains $(\sin 2x_1')''$, $\sigma_{11} = 3$, and the ES method finds

$$\mathbf{v} = [1,1]^T, \quad J = \{\, 1,2 \,\}, \quad l = 2, \quad d_1 = d_2 = 3, \quad \text{and} \quad \overline{c} = c_1 = 0.$$

To replace $x_1^{(d_1-c_1)} = x_1'''$ by

$$\left(y_1 + \frac{v_1}{v_2} x_2^{(d_2-\overline{c})}\right)^{(\overline{c}-c_1)} = y_1 + x_2''',$$

we first need to write

$$(\sin 2x_1')'' = 2x_1''' \cos x_1' - 4(x_1'')^2 \sin x_1'$$

so that x_1''' appears explicitly in f_1. Then we can substitute $y_1 + x_2'''$ for this x_1'''. □

In the following, we analyze the equivalence between the original DAE and the converted DAE resulting from the ES method. Our analysis below is similar to the analysis of the equivalence for the LC method.

We denote by $\mathcal{F}$ the original DAE with equations f_i and an identically singular System Jacobian $\mathbf{J}$. After an ES conversion, we obtain a converted DAE $\overline{\mathcal{F}}$ with equations $\overline{f}_i$, $i = 1:n$, and g_j, $j \in J \setminus \{ l \}$, and a System Jacobian $\overline{\mathbf{J}}$, whose nonsingularity does not matter here.

Assume that $\mathbf{x}(t) = \big[x_1(t), \ldots, x_n(t)\big]^T$ is a solution of $\mathcal{F}$ on some time interval $\mathbb{I} \subset \mathbb{R}$. Then functions f_i and derivatives of them vanish on $\mathbb{I}$. Assume also that $\mathbf{v} \in \ker(\mathbf{J})$ is well defined on $\mathbb{I}$ and $v_l(t) \neq 0$ for all $t \in \mathbb{I}$. Then we can use $\mathbf{x}(t)$ and (4.16) to construct $\mathbf{y}(t)$ comprising $y_j(t)$, $j \in J \setminus \{ l \}$, such that each function g_j in (4.18) and derivatives of them vanish on $\mathbb{I}$. Using (4.16) again, we perform substitutions in f_i, $i \in M$, to obtain $\overline{f}_i$, and let $\overline{f}_i = f_i$ for $i \notin M$ to obtain the rest of the functions $\overline{f}_i$. Obviously these substitutions do not change the function values, $\overline{f}_i$ and derivatives of them also vanish on $\mathbb{I}$. Therefore $(\mathbf{x}(t), \mathbf{y}(t))$ is a solution to $\overline{\mathcal{F}}$.

Conversely, assume that $(\overline{\mathbf{x}}(t), \overline{\mathbf{y}}(t))$ is a solution of $\overline{\mathcal{F}}$ on $\mathbb{I} \subset \mathbb{R}$. Recall that the vector $\mathbf{v}$, depending solely on t and $\overline{\mathbf{x}}(t)$, is well defined for all $t \in \mathbb{I}$. Since v_l is a denominator in each g_j in (4.18), the existence of the solution already implies $v_l(t) \neq 0$ on $\mathbb{I}$. Given that functions g_j and derivatives of them vanish on $\mathbb{I}$, from

(4.16) we have

$$y_j^{(m)} = \begin{cases} \left(x_j^{(d_j - \overline{c})} - \dfrac{v_j}{v_l} x_l^{(d_l - \overline{c})} \right)^{(m)} & j \in J \setminus \{ l \} \\ \left(x_l^{(d_l - \overline{c})} \right)^{(m)} & j = l, \end{cases}$$

where $m \geq 0$. If we substitute the expressions on the right-hand side for the derivatives of y_j in each $\overline{f}_i$, then we recover the original functions f_i and meanwhile do not change their function values. Therefore, functions f_i and derivatives of them also vanish on $\mathbb{I}$. Now variables $\overline{\mathbf{y}}(t)$ do not appear in $\mathcal{F}$, and $\overline{\mathbf{x}}(t)$ is a solution to $\mathcal{F}$.

The above discussion gives the following theorem.

Theorem 4.22 *If*

(i) a DAE $\mathcal{F}$ has a finite $\mathrm{Val}(\mathbf{\Sigma})$ *and an identically singular System Jacobian* $\mathbf{J}$*,*

(ii) a vector $\mathbf{v} \in \ker(\mathbf{J})$ *is well defined for all* t *on some time interval* $\mathbb{I}$*,*

(iii) the ES conditions (4.19) are satisfied, and we perform an ES conversion to obtain a DAE $\overline{\mathcal{F}}$*, and*

(iv) $v_l \neq 0$ *for all* $t \in \mathbb{I}$*,*

then DAEs $\mathcal{F}$ and $\overline{\mathcal{F}}$ are equivalent.

Example 4.23 In Example 4.18, assume we pick $l = 1$. Then by (4.16) we introduce for x_2 a new variable

$$y_2 = x_2^{(d_2 - \overline{c})} - \frac{v_2}{v_1} x_1^{(d_1 - \overline{c})} = x_2' - \frac{1}{x_2} x_1.$$

Here we use

$$d_1 = 1, \quad d_2 = 2, \quad \overline{c} = 1, \quad \text{and} \quad \mathbf{v} = [x_2, -1]^T.$$

Then we

substitute	for	in
$(y_2 - x_1/x_2)'$	x_2''	f_1
$y_2 - x_1/x_2$	x_2'	f_2

The resulting DAE is

$$\begin{aligned}
0 = \overline{f}_1 &= x_1 + e^{-x_1' - x_2 \cdot (y_2 - x_1/x_2)'} + h_1(t) \\
&= x_1 + e^{-x_1' - x_2 y_2' - x_2' x_1/x_2 + x_1'} + h_1(t) \\
&= x_1 + e^{-x_2 y_2' - x_2' x_1/x_2} + h_1(t) \\
0 = \overline{f}_2 &= x_1 + x_2(y_2 - x_1/x_2) + x_2^2 + h_2(t) \\
&= x_2 y_2 + x_2^2 + h_2(t) \\
0 = g &= -y_2 + x_2' + x_1/x_2.
\end{aligned}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2 \\ g \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & y_2 \end{array} \\ \left[\begin{array}{ccc} 0 & 1 & 1^\bullet \\ & 0^\bullet & 0 \\ 0^\bullet & 1 & 0 \end{array}\right] \\ \begin{array}{ccc} 0 & 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 1 \\ 0 \\ \text{Val}(\overline{\mathbf{\Sigma}}) = 1 \end{array}$$

$$\overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2' \\ g \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2' & y_2' \end{array} \\ \begin{bmatrix} 1 - x_2'\beta/x_2 & -x_1\beta/x_2 & -x_2\beta \\ & 2x_2 + y_2 & x_2 \\ 1/x_2 & 1 & \end{bmatrix} \end{array}$$

$$\det(\overline{\mathbf{J}}) = -x_2 + \beta(2x_2 + y_2 + x_2' - x_1/x_2)$$

In $\overline{\mathbf{J}}$, $\beta = \exp(-x_2y_2' - x_2'x_1/x_2)$. If $\det(\overline{\mathbf{J}}) \neq 0$, then SA succeeds and gives structural index $\nu_S = 2$. Here $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1 < 2 = \mathrm{Val}(\mathbf{\Sigma})$.

However, the original DAE and the resulting one are equivalent only if $v_1 = x_2 \neq 0$ on some time interval $\mathbb{I}$. In practice, it is more desirable to choose $l = 2$ since $v_l = -1$ is a nonzero constant; see also Example 4.18. □

Comparing Examples 4.18 with 4.23, we can see that it is again desirable to choose a column index $l \in J$, such that the v_l is an expression that never becomes 0, or even better, a (nonzero) constant. With this choice, the original DAE and the converted one are *always* equivalent. We hence derive a set $\overline{J}$, a subset of J that contains these column indices l for which $l \in J$ and v_l is constant:

$$\overline{J} = \big\{\, l \in J \mid v_l \text{ is constant} \,\big\}. \tag{4.22}$$

We summarize the steps of an ES conversion.

1) Obtain a symbolic form of $\mathbf{J}$.

2) Compute a vector $\mathbf{v} \in \ker(\mathbf{J})$.

3) Derive J, s, M, and $\overline{c}$ as defined in (4.15).

4) Check the ES conditions (4.19). If either of the conditions is not satisfied, then the ES method is not applicable and we set $J \leftarrow \emptyset$; otherwise proceed to the next step.

5) Set $\overline{J} \leftarrow \{\, l \in J \mid v_l \text{ is constant} \,\}$. If $\overline{J} \neq \emptyset$, then we choose an $l \in \overline{J}$; otherwise we choose an $l \in J$.

6) For each $j \in J \setminus \{\, l \,\}$, introduce y_j, as defined in (4.16), and append the corresponding equation g_j, as defined in (4.18).

7) Replace each $x_j^{(d_j - c_i)}$ in f_i by $\left(y_j + (v_j/v_l)\cdot x_l^{(d_l - \overline{c})}\right)^{(\overline{c} - c_i)}$, for all i and all $j \in J \setminus \{\, l \,\}$.

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

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

4.3 Choosing a desirable conversion

We present our rationale for choosing a conversion method in Table 4.1 and base our choice on the following observations. For some failure cases, either LC condition (4.2) or the ES conditions (4.19) are satisfied, but not both, so we can apply one conversion method only. For other cases where both methods are applicable, we consider as priority the equivalence between the original DAE and the resulting one. As discussed in the above two sections, we wish to choose a nonzero constant u_l [resp. v_l] in the LC [resp. ES] method, that is, $l \in \overline{L}$ [resp. $l \in \overline{J}$]. Our experience suggests that such a constant frequently exists for one of the methods. If both methods

Desirable conversion method		ES method		
		$\overline{J} \neq \emptyset$	$\overline{J} = \emptyset$ and $J \neq \emptyset$	$J = \emptyset$
LC method	$\overline{L} \neq \emptyset$	LC	LC	LC
	$\overline{L} = \emptyset$ and $L \neq \emptyset$	ES	LC	LC
	$L = \emptyset$	ES	ES	N/A

Table 4.1: The rationale for choosing the desirable conversion method.

guarantee equivalence or neither of them does, then we choose the LC method, as it replaces only one existing equation and maintains the problem size.

We summarize in Table 4.1 the above logic of finding a desirable conversion in the sense of equivalence. The three rows correspond to the three cases where

- some LC conversion is available with a constant u_l,
- some LC conversion is available but none of u_l is constant, and
- the LC method is not applicable.

The three columns correspond to the three cases where

- some ES conversion is available with a constant v_l,
- some ES conversion is available but none of v_l is constant, and
- the ES method is not applicable.

Take for instance the two cases for the positions $(2,1)$ and $(2,2)$ in Table 4.1. For position $(2,1)$, we can perform an LC conversion, but since $\overline{L} = \emptyset$ and $L \neq \emptyset$, none of $u_l \not\equiv 0$ for $l \in L$ is constant. Hence, no LC conversion is desirable. Meanwhile, because $\overline{J} \neq \emptyset$ in the ES method, some ES conversion with a nonzero constant $v_l \in \overline{J}$

is desirable. That is, this ES conversion guarantees the equivalence between the original DAE and the converted one, and hence is desirable. Therefore, we perform the ES conversion with this v_l. For position $(2,2)$, both methods are applicable but neither of them gives a desirable conversion, so we choose the LC method, because it is simpler to perform than the ES method.

In Chapters 6 and 7, when we apply the block conversion methods on an SA-unfriendly DAE, we shall use the same rationale for choosing a desirable conversion.

We end this chapter with another simple DAE on which neither the LC method nor the ES method is applicable, though the conversion is easy to find.

Example 4.24 Consider

$$\begin{aligned} 0 &= f_1 = x_1' x_2' + h_1(t) \\ 0 &= f_2 = (x_1' x_2')^2 + x_1 + x_2 + h_2(t). \end{aligned} \tag{4.23}$$

$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ d_j \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 1^{\bullet} & 1 \\ 1 & 1^{\bullet} \end{bmatrix} \\ \begin{array}{cc} 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ \mathrm{Val}(\mathbf{\Sigma}) = 2 \end{array} \qquad \mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ \det(\mathbf{J}) \equiv 0 \end{array} \begin{array}{c} \begin{array}{cc} x_1' & x_2' \end{array} \\ \begin{bmatrix} x_2' & x_1' \\ 2x_1'(x_2')^2 & 2x_2'(x_1')^2 \end{bmatrix} \end{array}$$

It is straightforward to come up with a fix: we introduce a new variable x_3 to represent the common sub-expression $x_1' x_2'$, and then replace this expression by x_3 in the two equations. This procedure seems an ES conversion. The resulting DAE is

$$\begin{aligned} 0 &= \overline{f}_1 = x_3 + h_1(t) \\ 0 &= \overline{f}_2 = x_3^2 + x_1 + x_2 + h_2(t) \\ 0 &= \overline{f}_3 = -x_3 + x_1' x_2'. \end{aligned}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2 \\ \overline{f}_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \left[\begin{array}{ccc} & & 0^{\bullet} \\ 0 & 0^{\bullet} & 0 \\ 1^{\bullet} & 1 & 0 \end{array}\right] \\ \begin{array}{ccc} 1 & 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 1 \\ 1 \\ 0 \\ \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}'_1 \\ \overline{f}'_2 \\ \overline{f}_3 \\ \\ \det(\overline{\mathbf{J}}) = x'_1 - x'_2 \end{array} \begin{array}{c} \begin{array}{ccc} x'_1 & x'_2 & x'_3 \end{array} \\ \left[\begin{array}{ccc} & & 1 \\ 1 & 1 & 2x_3 \\ x'_2 & x'_1 & -1 \end{array}\right] \end{array}$$

SA succeeds with $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1 < 2 = \mathrm{Val}(\mathbf{\Sigma})$ at any point where $x'_1 \neq x'_2$.

In the LC method, we choose $\mathbf{u} = [2x'_1x'_2, 1]^T \in \mathrm{coker}(\mathbf{J})$, and use (4.1) and (4.10) to find

$$I = \{\, 1, 2 \,\}, \quad \underline{c} = 0, \quad L = \{\, 1, 2 \,\}, \quad \text{and} \quad \overline{L} = \{\, 2 \,\}.$$

However, x'_1 and x'_2 occur in $\mathbf{u}$, so the LC condition (4.2) is violated:

$$\sigma\,(x_1, \mathbf{u}) = d_1 - \underline{c} = 1 \quad \text{and} \quad \sigma\,(x_2, \mathbf{u}) = d_2 - \underline{c} = 1.$$

Due to these violations, the procedure for the LC method produces $L = \emptyset$ to mean that this method is not applicable.

In the ES method, we choose $\mathbf{v} = [x'_1, x'_2]^T \in \ker(\mathbf{J})$, and use (4.15) and (4.22) to find

$$J = \{\, 1, 2 \,\}, \quad s = |J| = 2, \quad M = \{\, 1, 2 \,\}, \quad \overline{c} = \max_{i \in M} c_i = 0, \quad \text{and} \quad \overline{J} = \emptyset.$$

Similarly, x'_1 and x'_2 occur in v, so the first ES condition in (4.19) is violated:

$$\sigma\,(x_1, \mathbf{v}) = d_1 - \overline{c} = 1 \quad \text{and} \quad \sigma\,(x_2, \mathbf{v}) = d_2 - \overline{c} = 1.$$

Due to these violations, the procedure for the ES method produces $J = \emptyset$ to mean that this method is not applicable.

The incapability of the two conversion methods is due to a nonlinear operation on the common sub-expression that is again nonlinear in the derivatives of highest order. We believe that such a situation is rare in practice and should not affect the usefulness and applicability of our conversion methods. □

Chapter 5

Examples of basic conversion methods

In this chapter, we illustrate the conversion methods with two SA-unfriendly DAEs. After a conversion, if we obtain a DAE with a smaller value of the signature matrix, then we say this conversion *succeeds*.

In §5.1, we apply both conversion methods to the linear constant coefficient DAE (3.10). The LC method converts it to an SA-friendly one in two iterations, and reduces the value of the the signature matrix by 2. In contrast, the ES method reduces the value of the signature matrix by 1 in the first iteration, but becomes inapplicable in the second iteration, as the second ES condition in (4.19) is not satisfied.

In §5.2, we illustrate both conversion methods with an artificially modified DAE derived from the simple pendulum (2.10). The ES method produces an SA-friendly DAE of a relatively simple formulation. The LC method gives a complicated DAE, because the set $\overline{L}$ in the method is empty.

5.1 A simple linear constant coefficient DAE

Recall (3.10):

$$\mathcal{F}^0 : \begin{cases} 0 = f_1 = -x_1' + x_3 + b_1(t) \\ 0 = f_2 = -x_2' + x_4 + b_2(t) \\ 0 = f_3 = \quad x_2 + x_3 + x_4 + c_1(t) \\ 0 = f_4 = -x_1 + x_3 + x_4 + c_2(t). \end{cases} \tag{5.1}$$

$$\boldsymbol{\Sigma}^0 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1^\bullet & & 0 & \\ & 1^\bullet & & 0 \\ & 0 & 0^\bullet & 0 \\ 0 & & 0 & 0^\bullet \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 0 \\ 0 \\ \\ \mathrm{Val}(\boldsymbol{\Sigma}^0) = 2 \end{array} \qquad \mathbf{J}^0 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ \\ \det(\mathbf{J}^0) \equiv 0 \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ & -1 & & 1 \\ & & 1 & 1 \\ & & 1 & 1 \end{bmatrix} \end{array}$$

Here a superscript indicates an iteration number, not a power, so $\mathcal{F}^0$ denotes the original problem formulation. We let $\boldsymbol{\Sigma}^0$ and $\mathbf{J}^0$ denote the signature matrix and Jacobian of the original problem, respectively.

LC method. We perform two LC conversions and obtain an equivalent structurally regular DAE on which SA succeeds.

We compute $\mathbf{u} = [0, 0, -1, 1]^T \in \mathrm{coker}(\mathbf{J}^0)$ and use (4.1) and (4.10) to derive

$$I = \{\, 3, 4 \,\}, \quad \underline{c} = 0, \quad \text{and} \quad L = \overline{L} = \{\, 3, 4 \,\}.$$

We choose $l = 3 \in \overline{L}$ and replace f_3 by

$$\overline{f}_3 = u_3 f_3 + u_4 f_4 = -f_3 + f_4 = -x_1 - x_2 - c_1(t) + c_2(t).$$

The converted DAE is

$$\mathcal{F}^1 : \begin{cases} 0 = f_1 = -x_1' + x_3 + b_1(t) \\ 0 = f_2 = -x_2' + x_4 + b_2(t) \\ 0 = \overline{f}_3 = -x_1 - x_2 - c_1(t) + c_2(t) \\ 0 = f_4 = -x_1 + x_3 + x_4 + c_2(t). \end{cases}$$

$$\boldsymbol{\Sigma}^1 = \begin{array}{c} \\ f_1 \\ f_2 \\ \overline{f}_3 \\ f_4 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1^\bullet & & 0 & \\ & 1 & & 0^\bullet \\ 0 & 0^\bullet & & \\ 0 & & 0^\bullet & 0 \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 0 \\ \mathrm{Val}(\boldsymbol{\Sigma}^1) = 1 \end{array} \qquad \mathbf{J}^1 = \begin{array}{c} \\ f_1 \\ f_2 \\ \overline{f}_3' \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & 1 & \\ & -1 & & 1 \\ -1 & -1 & & \\ & & 1 & 1 \end{bmatrix} \\ \det(\mathbf{J}^1) \equiv 0 \end{array}$$

Since $\mathbf{J}^1$ is still identically singular, we try another LC conversion. We compute $\mathbf{u} = [-1, -1, 1, 1]^T$ in $\mathrm{coker}(\mathbf{J}^1)$. Then

$$I = \{\, 1, 2, 3, 4 \,\}, \quad \underline{c} = 0, \quad \text{and} \quad L = \overline{L} = \{\, 1, 2, 4 \,\}.$$

We choose $l = 1 \in \overline{L}$ and replace f_1 by

$$\begin{aligned}
\overline{f}_1 &= u_1 f_1 + u_2 f_2 + u_3 \overline{f}_3' + u_4 f_4 \\
&= -f_1 - f_2 + \overline{f}_3' + f_4 \\
&= -\left[-x_1' + x_3 + b_1(t)\right] - \left[-x_2' + x_4 + b_2(t)\right] + \left[-x_1 - x_2 - c_1(t) + c_2(t)\right]' \\
&\qquad + \left[-x_1 + x_3 + x_4 + c_2(t)\right] \\
&= -x_1 - b_1(t) - b_2(t) - c_1'(t) + c_2'(t) + c_2(t).
\end{aligned}$$

The converted DAE is

$$\mathcal{F}^2 : \begin{cases} 0 = \overline{f}_1 = -x_1 - b_1(t) - b_2(t) - c_1'(t) + c_2'(t) + c_2(t) \\ 0 = f_2 = -x_2' + x_4 + b_2(t) \\ 0 = \overline{f}_3 = -x_1 - x_2 - c_1(t) + c_2(t) \\ 0 = f_4 = -x_1 + x_3 + x_4 + c_2(t). \end{cases}$$

$$\mathbf{\Sigma}^2 = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ \overline{f}_3 \\ f_4 \\ \widetilde{d}_j \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0^{\bullet} & & & \\ & 1 & & 0^{\bullet} \\ 0 & 0^{\bullet} & & \\ 0 & & 0^{\bullet} & 0 \end{bmatrix} \\ \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \end{array} \begin{array}{c} \widetilde{c}_i \\ 1 \\ 0 \\ 1 \\ 0 \end{array} \qquad \mathrm{Val}(\mathbf{\Sigma}^2) = 0$$

$$\mathbf{J}^2 = \begin{array}{c} \\ \overline{f}_1' \\ f_2 \\ \overline{f}_3' \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1' & x_2' & x_3 & x_4 \end{array} \\ \begin{bmatrix} -1 & & & \\ & -1 & & 1 \\ -1 & -1 & & \\ & & 1 & 1 \end{bmatrix} \end{array} \qquad \det(\mathbf{J}^2) = 1$$

The SA succeeds on this converted DAE and gives structural index $\nu_S = 2$. Since we choose $l \in \overline{L}$ for both LC conversions, the equivalence of DAEs $\mathcal{F}^2$, $\mathcal{F}^1$, and $\mathcal{F}^0$ is

guaranteed.

ES method. We show below that the ES method cannot convert $\mathcal{F}^0$ in (5.1) to a structurally regular DAE. We illustrate one choice of $l \in J$ in each iteration of the ES method, and do not explore all possible combinations of choices. To handle the limitation of the ES method, further development is required and left as future work.

We find $\mathbf{v} = [1, -1, 1, -1]^T \in \ker(\mathbf{J}^0)$, and the ES method uses (4.15) to obtain

$$J = \overline{J} = \{ 1, 2, 3, 4 \}, \quad s = |J| = 4, \quad M = \{ 1, 2, 3, 4 \}, \quad \text{and} \quad \overline{c} = \max_{i \in M} c_i = 0.$$

Assume we pick $l = 3$. Using (4.16), we introduce y_j for each $j \in J \setminus \{ l \} = \{ 1, 2, 4 \}$:

$$\begin{aligned} y_1 &= x_1^{(d_1 - \overline{c})} - (v_1/v_3) x_3^{(d_3 - \overline{c})} = x_1' - x_3 \\ y_2 &= x_2^{(d_2 - \overline{c})} - (v_2/v_3) x_3^{(d_3 - \overline{c})} = x_2' + x_3 \\ y_4 &= x_4^{(d_4 - \overline{c})} - (v_4/v_3) x_3^{(d_3 - \overline{c})} = x_4 + x_3. \end{aligned} \tag{5.2}$$

From (5.2), we construct equations g_j in (4.18). By (5.2), we write

$$x_1' = y_1 + x_3, \quad x_2' = y_2 - x_3, \quad \text{and} \quad x_4 = y_4 - x_3.$$

In (3.10), we

substitute	for	in
$y_1 + x_3$	x_1'	f_1
$y_2 - x_3$	x_2'	f_2
$y_4 - x_3$	x_4	$f_2,\ f_3,\ f_4$

The converted DAE is

$$
\begin{aligned}
0 = f_1 &= -y_1 + b_1(t) \\
0 = f_2 &= \quad y_4 - y_2 + b_2(t) \\
0 = f_3 &= \quad x_2 + y_4 + c_1(t) \\
0 = f_4 &= -x_1 + y_4 + c_2(t) \\
0 = g_1 &= -y_1 + x_1' - x_3 \\
0 = g_2 &= -y_2 + x_2' + x_3 \\
0 = g_4 &= -y_4 + x_4 + x_3.
\end{aligned}
$$

$$
\overline{\boldsymbol{\Sigma}} = \begin{array}{c|ccccccc|c}
 & x_1 & x_2 & x_3 & x_4 & y_1 & y_2 & y_4 & c_i \\
\hline
f_1 & & & & & 0^{\bullet} & & & 0 \\
f_2 & & & & & & 0^{\bullet} & 0 & 0 \\
f_3 & & 0^{\bullet} & & & & & 0 & 1 \\
f_4 & 0 & & & & & & 0^{\bullet} & 1 \\
g_1 & 1^{\bullet} & & 0 & & 0 & & & 0 \\
g_2 & & 1 & 0^{\bullet} & & & 0 & & 0 \\
g_4 & & & 0 & 0^{\bullet} & & & 0 & 0 \\
\hline
d_j & 1 & 1 & 0 & 0 & 0 & 0 & 1 & \mathrm{Val}(\overline{\boldsymbol{\Sigma}}) = 1
\end{array}
$$

$$\overline{\mathbf{J}} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3' \\ f_4' \\ g_1 \\ g_2 \\ g_4 \end{array} \begin{array}{c} \begin{array}{ccccccc} x_1' & x_2' & x_3 & x_4 & y_1 & y_2 & y_4' \end{array} \\ \begin{bmatrix} & & & & -1 & & \\ & & & & & -1 & \\ & 1 & & & & & 1 \\ -1 & & & & & & 1 \\ 1 & & -1 & & -1 & & \\ & 1 & 1 & & & -1 & \\ & & 1 & 1 & & & \end{bmatrix} \end{array}$$

$$\det(\overline{\mathbf{J}}) \equiv 0$$

Since $\overline{\mathbf{J}}$ is still identically singular, we attempt another ES conversion. We compute $\mathbf{v} = [1, -1, 1, -1, 0, 0, 1]^T \in \ker(\overline{\mathbf{J}})$ and use (4.15) to find

$$J = \overline{J} = \{\, 1, 2, 3, 4, 7 \,\}, \quad s = |J| = 5, \quad M = \{\, 3, 4, 5, 6, 7 \,\}, \quad \text{and} \quad \overline{c} = \max_{i \in M} c_i = 1.$$

Since

$$d_3 - \overline{c} = d_4 - \overline{c} = 0 - 1 = -1 < 0 \quad \text{for } j = 3, 4 \in J,$$

the latter ES condition in (4.19) is not satisfied. If we perform an ES conversion, then a strict decrease in $\mathrm{Val}(\overline{\mathbf{\Sigma}})$ does not occur. We omit the details of this conversion due to the large size of the resulting DAE.

5.2 Modified pendulum by change of variables

If we perform the following linear transformation on the state variables in the pendulum DAE (2.10)

$$\begin{bmatrix} x \\ y \\ \lambda \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}, \tag{5.3}$$

then we obtain a new DAE from the original one:

$$\begin{aligned} 0 &= f_1 = (z_1 + z_2)'' + (z_1 + z_2)(z_3 + z_1) \\ 0 &= f_2 = (z_2 + z_3)'' + (z_2 + z_3)(z_3 + z_1) - G \\ 0 &= f_3 = (z_1 + z_2)^2 + (z_2 + z_3)^2 - \ell^2. \end{aligned} \tag{5.4}$$

$$\boldsymbol{\Sigma}^0 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 2 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{array}{ccc} 2 & 2 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 2 \\ \end{array} \quad \mathrm{Val}(\boldsymbol{\Sigma}^0) = 2$$

$$\mathbf{J}^0 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3'' \end{array} \begin{array}{c} \begin{array}{ccc} z_1'' & z_2'' & z_3'' \end{array} \\ \begin{bmatrix} 1 & 1 & \\ & 1 & 1 \\ 2\alpha & 2(\alpha+\beta) & 2\beta \end{bmatrix} \end{array} \quad \det(\mathbf{J}^0) \equiv 0$$

Here $\alpha = z_1 + z_2$ and $\beta = z_2 + z_3$.

LC method. We perform two LC conversions on (5.4) to obtain a structurally regular DAE. Its formulation is complicated, partly because we cannot find a nonzero constant u_l in each iteration of the method. The equivalence of the resulting DAE and the original one requires the u_l in each iteration to be nonzero.

We first compute $\mathbf{u} = \big(\alpha, \beta, -1/2\big) \in \operatorname{coker}(\mathbf{J}^0)$. Using (4.1) and (4.10), we obtain

$$I = \big\{\, 1, 2, 3 \,\big\}, \quad \underline{c} = 0, \quad L = \big\{\, 1, 2 \,\big\}, \quad \text{and} \quad \overline{L} = \emptyset.$$

Since $\overline{L} = \emptyset$ and we cannot guarantee $u_1 = \alpha = z_1 + z_2$ and $u_2 = \beta = z_2 + z_3$ to be always nonzero, the converted DAE is equivalent to (5.4) only if $u_l \neq 0$ for the l we pick.

We illustrate the case $u_1 = \alpha = z_1 + z_2 \neq 0$. The other case $\beta \neq 0$ can be analyzed in an analogous way. We pick $l = 1 \in L$ and replace f_1 by

$$\begin{aligned}
\overline{f}_1 &= u_1 f_1 + u_2 f_2 + u_3 f_3'' \\
&= (z_1 + z_2) f_1 + (z_2 + z_3) f_2 - f_3''/2 \\
&= \cancel{(z_1 + z_2)(z_1 + z_2)''} + (z_1 + z_2)^2 (z_3 + z_1) \cancel{+ (z_2 + z_3)(z_2 + z_3)''} \\
&\qquad + (z_2 + z_3)^2 (z_3 + z_1) - G(z_2 + z_3) \cancel{- (z_1 + z_2)(z_1 + z_2)''} \\
&\qquad - (z_1' + z_2')^2 \cancel{- (z_2 + z_3)(z_2 + z_3)''} - (z_2' + z_3')^2 \\
&= \big[(z_1 + z_2)^2 + (z_2 + z_3)^2\big] (z_3 + z_1) - G(z_2 + z_3) - (z_1' + z_2')^2 - (z_2' + z_3')^2 \\
&= \ell^2 (z_3 + z_1) - G(z_2 + z_3) - (z_1' + z_2')^2 - (z_2' + z_3')^2.
\end{aligned}$$

The resulting DAE is

$$\mathcal{F}^1 : \begin{cases} 0 = \overline{f}_1 = \ell^2 (z_3 + z_1) - G(z_2 + z_3) - (z_1' + z_2')^2 - (z_2' + z_3')^2 \\ 0 = f_2 = (z_2 + z_3)'' + (z_2 + z_3)(z_3 + z_1) - G \\ 0 = f_3 = (z_1 + z_2)^2 + (z_2 + z_3)^2 - \ell^2. \end{cases}$$

$$\mathbf{\Sigma}^1 = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & c_i \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 0 \end{bmatrix} \begin{array}{c} 1 \\ 0 \\ 2 \end{array} \\ \begin{array}{ccc} 2 & 2 & 2 \end{array} \end{array} \quad \text{Val}(\mathbf{\Sigma}^1) = 3$$

$$\mathbf{J}^1 = \begin{array}{c} \\ \overline{f}'_1 \\ f_2 \\ f''_3 \end{array} \begin{array}{c} \begin{array}{ccc} z''_1 & z''_2 & z''_3 \end{array} \\ \begin{bmatrix} -2\alpha' & -2(\alpha+\beta)' & -2\beta' \\ & 1 & 1 \\ 2\alpha & 2(\alpha+\beta) & 2\beta \end{bmatrix} \end{array} \quad \det(\mathbf{J}^1) \equiv 0$$

We use α and β to denote $z_1 + z_2$ and $z_2 + z_3$, respectively. Also let γ denote $z_3 + z_1$ to simplify notation. By (5.3), we notice that variables α, β, γ are in fact the state variables (x, y, λ) in (2.10). However, in our illustration of the LC method, we prefer not to apply the ES method by replacing $z_1 + z_2$, $z_2 + z_3$, and $z_3 + z_1$ by α, β, and γ, respectively.

Since the System Jacobian $\mathbf{J}^1$ is still identically singular, we attempt another LC conversion. We compute $\mathbf{u} = [\alpha,\, 2\alpha\beta' - 2\beta\alpha',\, \alpha']^T \in \text{coker}(\mathbf{J}^1)$. Using (4.1) and (4.10) again, we find $I = \{\, 1, 2, 3 \,\}$, $\underline{c} = 0$, $L = \{\, 2 \,\}$, and $\overline{L} = \emptyset$. Suppose

$$u_2/2 = \alpha\beta' - \beta\alpha' = (z_1 + z_2)(z'_2 + z'_3) - (z_2 + z_3)(z'_1 + z'_2) \neq 0.$$

We choose $l = 2 \in L$, and replace f_2 by

$$\begin{aligned}
\overline{f}_2 &= u_1 f'_1 + u_2 f_2 + u_3 f''_3 = \alpha f'_1 + 2(\alpha\beta' - \alpha'\beta) f_2 + \alpha' f''_3 \\
&= \alpha(\ell^2\gamma' - G\beta' - 2\alpha'\underline{\alpha''} - 2\beta'\underline{\beta''}) + 2(\alpha\beta' - \alpha'\beta)(\underline{\beta''} + \beta\gamma - G) \\
&\qquad + 2\alpha'(\alpha'^2 + \alpha\underline{\alpha''} + \beta'^2 + \beta\underline{\beta''}) \\
&= \alpha(\ell^2\gamma' - G\beta') + 2(\alpha\beta' - \alpha'\beta)(\beta\gamma - G) + 2\alpha'(\alpha'^2 + \beta'^2) \\
&= (z_1 + z_2)\Big[\ell^2(z'_3 + z'_1) - G(z'_1 + z'_2)\Big] \\
&\qquad + 2\Big[(z_1 + z_2)(z'_2 + z'_3) - (z'_1 + z'_2)(z_2 + z_3)\Big]\Big[(z_2 + z_3)(z_3 + z_1) - G\Big] \\
&\qquad + 2(z'_1 + z'_2)\Big[(z'_1 + z'_2)^2 + (z'_2 + z'_3)^2\Big].
\end{aligned}$$

The resulting DAE is

$$\mathcal{F}^2 : \begin{cases} 0 = \overline{f}_1 = \ell^2(z_3 + z_1) - G(z_2 + z_3) - (z_1' + z_2')^2 - (z_2' + z_3')^2 \\ 0 = \overline{f}_2 = (z_1 + z_2)\Big[\ell^2(z_3' + z_1') - G(z_1' + z_2')\Big] \\ \qquad\quad + 2\Big[(z_1 + z_2)(z_2' + z_3') - (z_1' + z_2')(z_2 + z_3)\Big]\Big[(z_2 + z_3)(z_3 + z_1) - G\Big] \\ \qquad\quad + 2(z_1' + z_2')\Big[(z_1' + z_2')^2 + (z_2' + z_3')^2\Big] \\ 0 = f_3 = (z_1 + z_2)^2 + (z_2 + z_3)^2 - \ell^2. \end{cases}$$

$$\mathbf{\Sigma}^2 = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & c_i \end{array} \\ \begin{bmatrix} 1 & 1 & 1^\bullet \\ 1^\bullet & 1 & 1 \\ 0 & 0^\bullet & 0 \end{bmatrix} \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \\ \begin{array}{ccc} 1 & 1 & 1 \end{array} \quad \mathrm{Val}(\mathbf{\Sigma}^2) = 2 \end{array}$$

The System Jacobian $\mathbf{J}^2$ is complicated, so we do not show it here. Its determinant is

$$\begin{aligned} \det(\mathbf{J}^2) &= -4\ell^2\,(z_1 + z_2)\,[(z_1 + z_2)(z_2' + z_3') - (z_2 + z_3)(z_1' + z_2')] \\ &= -4\alpha\ell^2(\alpha\beta' - \beta\alpha') \neq 0. \end{aligned}$$

This nonzero attributes to $\alpha = z_1 + z_2 \neq 0$ and $\alpha\beta' - \beta\alpha' \neq 0$. The converted DAE $\mathcal{F}^2$ is equivalent to $\mathcal{F}^1$ only if $\alpha\beta' - \beta\alpha' \neq 0$ on some time interval $\mathbb{I}$. Hence SA succeeds and gives structural index $\nu_S = 1$.

Now we consider the case $\alpha\beta' - \beta\alpha' = 0$. Since $0 = h' = 2\alpha\alpha' + 2\beta\beta'$ and $\alpha \neq 0$, we have

$$0 = \alpha\beta' - \beta\alpha' = \alpha\beta' + \beta \cdot (\beta\beta')/\alpha = \beta'(\alpha^2 + \beta^2)/\alpha = \beta'\ell^2/\alpha.$$

So $\beta' = \alpha' = 0$. Since $\mathbf{u} = \big(\alpha, 2\alpha\beta' - 2\beta\alpha', \alpha'\big)^T = [\alpha, 0, 0]^T$, the first row in $\mathbf{J}^1$ is identically zero and the System Jacobian is structurally singular. Hence the conversion methods are not applicable here.

To sum up, the DAEs $\mathcal{F}^2$ and $\mathcal{F}^0$ are equivalent, if

$$\alpha = z_1 + z_2 \neq 0 \quad \text{and} \quad \beta' = z_2' + z_3' \neq 0.$$

ES method. Suppose we choose $\mathbf{v} = [1, -1, 1]^T \in \ker(\mathbf{J}^0)$. We use (4.15) to derive

$$J = \overline{J} = \big\{\, 1, 2, 3 \,\big\}, \quad s = |J| = 3, \quad M = \big\{\, 1, 2, 3 \,\big\}, \quad \text{and} \quad \overline{c} = \max_{i \in M} c_i = c_3 = 2.$$

We illustrate below the ES method with the choice $l = 1 \in \overline{J}$.

Since $J \setminus \big\{\, l \,\big\} = \big\{\, 2, 3 \,\big\}$, we introduce for z_2 and z_3 two new variables

$$w_2 = z_2^{(d_2 - \overline{c})} - \frac{v_2}{v_1} z_1^{(d_1 - \overline{c})} = z_2 + z_1$$
$$\text{and} \quad w_3 = z_3^{(d_3 - \overline{c})} - \frac{v_3}{v_1} z_1^{(d_1 - \overline{c})} = z_3 - z_1,$$

respectively. To perform the expression substitutions, we first write explicitly the

derivatives z_1'', z_2'' and z_3'' in f_1 and f_2:

$$0 = f_1 = z_1'' + z_2'' + (z_1 + z_2)(z_3 + z_1)$$
$$0 = f_2 = z_2'' + z_3'' + (z_2 + z_3)(z_3 + z_1) - G.$$

Then we

substitute	for	in
$w_2'' - z_1''$	z_2''	$f_1,\ f_2$
$w_3'' + z_1''$	z_3''	f_2
$w_2 - z_1$	z_2	f_3
$w_3 + z_1$	z_3	f_3

The resulting DAE is

$$\begin{aligned} 0 &= \overline{f}_1 = w_2'' + (x_1 + x_2)(x_3 + x_1) \\ 0 &= \overline{f}_2 = w_2'' + w_3'' + (x_2 + x_3)(x_3 + x_1) - G \\ 0 &= \overline{f}_3 = w_2^2 + (w_2 + w_3)^2 - \ell^2 \\ 0 &= \overline{f}_4 = -w_2 + x_2 + x_1 \\ 0 &= \overline{f}_5 = -w_3 + x_3 - x_1 \end{aligned} \tag{5.5}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2 \\ \overline{f}_3 \\ \overline{f}_4 \\ \overline{f}_5 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & w_2 & w_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 2^{\bullet} & \\ 0^{\bullet} & 0 & 0 & 2 & 2 \\ & & & 0 & 0^{\bullet} \\ 0 & 0^{\bullet} & & 0 & \\ 0 & & 0^{\bullet} & & 0 \end{bmatrix} \\ \begin{array}{ccccc} 0 & 0 & 0 & 2 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \end{array} \quad \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 2$$

$$\overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}_1 \\ \overline{f}_2 \\ \overline{f}_3'' \\ \overline{f}_4 \\ \overline{f}_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & w_2'' & w_3'' \end{array} \\ \begin{bmatrix} 2x_1 + x_2 + x_3 & x_3 + x_1 & x_1 + x_2 & 1 & \\ x_2 + x_3 & x_3 + x_1 & x_1 + x_2 + 2x_3 & 1 & 1 \\ & & & 2(2w_2 + w_3) & 2(w_2 + w_3) \\ 1 & 1 & & & \\ -1 & & 1 & & \end{bmatrix} \end{array}$$

$$\det(\overline{\mathbf{J}}) = -4\ell^2$$

We use equation $\overline{f}_3 = 0$ to derive

$$\det(\overline{\mathbf{J}}) = -4(2w_2^2 + 2w_2w_3 + w_3^2) = -4\ell^2 \neq 0.$$

Hence the SA succeeds on (5.5). Since we choose $l \in \overline{J}$ in the ES conversion, the converted DAE (5.5) and the original (5.4) are always equivalent.

Chapter 6

Block conversion methods

In this chapter, we combine our conversion methods with block triangularization of a DAE, and derive the *block conversion methods*—that is, the *block LC method* and the *block ES method*. If a System Jacobian $\mathbf{J}$ is identically singular, and the DAE has a nontrivial BTF of $p \geq 1$ diagonal blocks, then by (2.14), $\det(\mathbf{J}) = \prod_{q=1}^{p} \det(\mathbf{J}_{qq}) \equiv 0$, so at least one $\mathbf{J}_{qq}$ for some $q = 1:p$ is identically singular. As discussed in §2.2.2, we can regard block q as a sub-DAE, with signature matrix $\boldsymbol{\Sigma}_{qq}$ and System Jacobian $\mathbf{J}_{qq}$. Then we may wish to apply the basic conversion methods on this sub-DAE to achieve a strict decrease in $\mathrm{Val}(\boldsymbol{\Sigma}_{qq})$, provided the conditions for applying these methods are satisfied for those variables and equations within block q.

However, what we should ensure is a strict decrease in the value of the *whole* signature matrix, namely $\mathrm{Val}(\overline{\boldsymbol{\Sigma}}) < \mathrm{Val}(\boldsymbol{\Sigma})$, where $\overline{\boldsymbol{\Sigma}}$ is the signature matrix of the resulting DAE. Proving this inequality from a decrease in $\mathrm{Val}(\boldsymbol{\Sigma}_{qq})$ is nontrivial, because a conversion on block q may affect blocks $\boldsymbol{\Sigma}_{qw}$ for $w = 1, \ldots, q-1, q+1, \ldots, p$. Especially in the ES method, $\boldsymbol{\Sigma}_{qq}$ and these blocks are enlarged. Hence, the conditions and the conversion processes from §4 need to be carefully modified.

Now that the block conversion methods allow us to deal with only those equations and variables within a singular block, which is usually of a smaller size compared to the whole DAE, these methods require fewer symbolic computations and hence are generally more efficient to find a useful conversion for fixing SA's failures.

Recall that when a Jacobian pattern $\mathbf{S}_0$ of a DAE is in some BTF, $\mathbf{\Sigma}$ and $\mathbf{J}$ are in $p \times p$ block form:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} & \cdots & \mathbf{J}_{1p} \\ \mathbf{0} & \mathbf{J}_{22} & \cdots & \mathbf{J}_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_{pp} \end{bmatrix} \qquad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} & \cdots & \mathbf{\Sigma}_{1p} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} & \cdots & \mathbf{\Sigma}_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{\Sigma}_{p1} & \mathbf{\Sigma}_{p2} & \cdots & \mathbf{\Sigma}_{pp} \end{bmatrix}.$$

From this block triangularization, we have

$$\begin{aligned} d_j - c_i & \begin{cases} > \sigma_{ij} & \text{if blockOf}(i) > \text{blockOf}(j) \\ \geq \sigma_{ij} & \text{if blockOf}(i) \leq \text{blockOf}(j), \end{cases} \\ \text{and} \quad J_{ij} & \equiv 0 \qquad\quad \text{if blockOf}(i) > \text{blockOf}(j). \end{aligned} \tag{6.1}$$

This chapter is organized as follows. We give an introductory example in §6.1. Then we present the block LC method in §6.2 and the block ES method in §6.3. We illustrate these two block conversion methods with the Campbell-Griepentrog robot arm DAE [5]. For more examples on which these methods are applied, see Chapter 7.

Throughout the rest of this thesis, we use the fine BTF in the examples for demonstration, since each fine block contains an irreducible sub-Jacobian sparsity pattern. Our experience suggests that a useful conversion can usually be derived from the fine

BTF of a DAE. However, we emphasize that the block conversion methods can be applied not only to the irreducible BTF of a Jacobian pattern $\mathbf{S}_0$ with some valid offset pair $(\mathbf{c};\mathbf{d})$, but also to any BTF of $\mathbf{S}_0$. For example, the basic conversion methods consider a DAE in a (trivial) BTF of one $n \times n$ block.

6.1 An introductory example

Consider

$$\begin{aligned} 0 &= f_1 = x_1 + x_2 + h_1(t) \\ 0 &= f_2 = x_1 + (x_1' + x_2')x_3' + h_2(t) \\ 0 &= f_3 = x_3' + h_3(t). \end{aligned} \tag{6.2}$$

$$\mathbf{\Sigma} = \begin{array}{c|ccc|c} & x_1 & x_2 & x_3 & c_i \\ f_1 & 0^{\bullet} & 0 & & 1 \\ f_2 & 1 & 1^{\bullet} & 1 & 0 \\ \hline f_3 & & & 1^{\bullet} & 0 \\ \hline d_j & 1 & 1 & 1 & \end{array} \qquad \mathrm{Val}(\mathbf{\Sigma}) = 2$$

$$\mathbf{J} = \begin{array}{c|ccc} & x_1' & x_2' & x_3' \\ f_1' & 1 & 1 & \\ f_2 & x_3' & x_3' & x_1' + x_2' \\ \hline f_3 & & & 1 \end{array} \qquad \det(\mathbf{J}) \equiv 0$$

The coarse BTF and the fine BTF are identical.

In the basic LC method, we can choose $\mathbf{u} = [-x_3', 1, -x_1' - x_2']^T \in \mathrm{coker}(\mathbf{J})$. Using (4.1), we have

$$I = \big\{\, i \mid u_i \not\equiv 0 \,\big\} = \big\{\, 1, 2, 3 \,\big\}, \quad \underline{c} = \min_{i \in I} c_i = 0, \quad L = \big\{\, l \in I \mid c_l = \underline{c} \,\big\} = \big\{\, 2, 3 \,\big\}.$$

Since

$$\sigma(x_j, \mathbf{u}) = 1 \not< 1 = d_j - \underline{c} \quad \text{for all} \quad j = 1:3,$$

the LC condition (4.2) is violated. An LC conversion outputs $\overline{L} = \emptyset$, which means the basic LC method is not applicable. Not surprisingly, replacing either f_2 or f_3 by

$$\overline{f} = \sum_{i \in I} u_i f_i^{(c_i - \underline{c})} = -x_3' h_1'(t) + (x_1 + h_2(t)) - (x_1' + x_2')(x_3' + h_3(t))$$

does not result in a decrease in $\mathrm{Val}(\mathbf{\Sigma})$—verifying this is not difficult.

Notice that only the sub-Jacobian of block 1, $\mathbf{J}_{11} = \partial(f_1', f_2)/\partial(x_1', x_2')$, is singular. Suppose we consider block 1, with $B_1 = \{1, 2\}$, as a sub-DAE, and choose $\mathbf{u} = [-x_3', 1]^T \in \mathrm{coker}(\mathbf{J}_{11})$. Within block 1, the LC method derives

$$I = \{\, i \in B_1 \mid u_i \not\equiv 0 \,\} = \{1, 2\}, \quad \underline{c} = \min_{i \in I} c_i = 0, \quad L = \{\, l \in I \mid c_l = \underline{c} \,\} = \{2\}.$$

Now the LC condition (4.2) is satisfied for the column indices in block 1: $\sigma(x_j, \mathbf{u}) = -\infty < d_j - \underline{c}$ for $j = 1, 2 \in B_1$. Replacing f_2 by $\overline{f}_2 = u_1 f_1' + u_2 f_2 = x_1 + h_2(t) - x_3' h_1'(t)$ results in the DAE with the following SA result.

$$\overline{\mathbf{\Sigma}} = \begin{array}{c|ccc|c} & x_2 & x_1 & x_3 & c_i \\ f_1 & 0^{\bullet} & 0 & & 0 \\ \overline{f}_2 & & 0^{\bullet} & 1 & 0 \\ f_3 & & & 1^{\bullet} & 0 \\ \hline d_j & 0 & 0 & 1 & \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c|ccc} & x_2 & x_1 & x_3' \\ f_1 & 1 & 1 & \\ \overline{f}_2 & & 1 & g_1'(t) \\ f_3 & & & 1 \\ \hline & & & \det(\overline{\mathbf{J}}) = 1 \end{array}$$

The SA succeeds as $\overline{\mathbf{J}}$ is nonsingular. The conversion results in a decrease in the value of the signature matrix: $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1 < 2 = \mathrm{Val}(\mathbf{\Sigma})$.

The basic ES method can work on (6.2) by choosing $\mathbf{v} = [1, -1, 0]^T \in \ker(\mathbf{J})$. It is simpler—though trivial for this example—to work on block 1 only. We find $\mathbf{v} = [1, -1]^T \in \ker(\mathbf{J}_{11})$, and use (4.15) to derive

$$J = \{\, l \in B_1 \mid v_l \not\equiv 0 \,\} = \{\, 1, 2 \,\}, \quad s = |J| = 2, \quad M = \{\, 1, 2 \,\}, \quad \overline{c} = \max_{i \in M} c_i = 1.$$

Since $\mathbf{v}$ is constant, the first ES condition in (4.19) certainly hold. The second condition holds also as $d_1 - c_1 = d_2 - c_1 = 0$.

We choose $l = 2 \in J$ and introduce for x_1 a new variable

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

The ES method hence says: replace x_1 by $y_1 - x_2$ in f_1, and replace x_1' by $y_1' - x_2'$ in f_2. Finally we append the equation g_1 that defines y_1 and prescribes such replacements, and obtain

$$0 = \overline{f}_1 = y_1 + h_1(t)$$

$$0 = \overline{f}_2 = x_1 + y_1' x_3' + h_2(t)$$

$$0 = \overline{f}_3 = x_3' + h_3(t)$$

$$0 = g_1 = -y_1 + x_1 + x_2.$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ g_1 \\ \overline{f}_2 \\ \overline{f}_1 \\ f_3 \\ d_j \end{array} \begin{array}{c} \begin{array}{cccc} x_2 & x_1 & y_1 & x_3 \end{array} \\ \left[\begin{array}{cccc} 0^{\bullet} & 0 & 0 & \\ & 0^{\bullet} & 1 & 1 \\ & & 0^{\bullet} & \\ & & & 1^{\bullet} \end{array}\right] \\ \begin{array}{cccc} 0 & 0 & 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 0 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ g_1 \\ \overline{f}_2 \\ \overline{f}_1' \\ f_3 \end{array} \begin{array}{c} \begin{array}{cccc} x_2 & x_1 & y_1' & x_3' \end{array} \\ \left[\begin{array}{cccc} 1 & 1 & -1 & \\ & 1 & x_3' & y_1' \\ & & 1 & \\ & & & 1 \end{array}\right] \end{array}$$

Again $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 1 < 2 = \mathrm{Val}(\mathbf{\Sigma})$, and the SA succeeds as $\det(\overline{\mathbf{J}}) = 1$.

6.2 Block linear combination method

We first introduce some convenient notation for the block LC method. Assume that a $\mathbf{J}_{qq}$ is identically singular. We use $\mathbf{0}_m$ to denote the zero vector of size m. Let $\widehat{\mathbf{u}} \in \mathrm{coker}(\mathbf{J}_{qq})$ and $\widehat{\mathbf{u}} \not\equiv \mathbf{0}_{N_q}$. Let also

$$\mathbf{u} = \begin{bmatrix} \mathbf{0}_{N_1+\cdots+N_{q-1}} \\ \widehat{\mathbf{u}} \\ \mathbf{0}_{N_{q+1}+\cdots+N_p} \end{bmatrix}.$$

We denote

$$\begin{aligned} &I = \big\{\, i \mid u_i \not\equiv 0 \,\big\} \subseteq B_q, \quad \underline{c} = \min_{i\in I} c_i, \quad L = \big\{\, l \in I \mid c_l = \underline{c} \,\big\}, \quad \text{and} \\ &\overline{L} = \big\{\, l \in L \mid u_l \text{ is (nonzero) constant} \,\big\}. \end{aligned} \tag{6.3}$$

The set $\overline{L}$ is used to seek a conversion that guarantees equivalence between the original DAE and the converted one. The block LC method is based on the following theorem.

Theorem 6.1 *If*

$$\sigma\left(x_j, \mathbf{u}\right) < d_j - \underline{c} \qquad \textit{for all } j \in B_q \tag{6.4}$$

and we replace an equation f_l, $l \in L$, by

$$\overline{f} = \sum_{i\in I} u_i f_i^{(c_i - \underline{c})}, \tag{6.5}$$

then $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$, *where* $\overline{\mathbf{\Sigma}} = (\overline{\sigma}_{ij})$ *is the signature matrix of the resulting DAE.*

We give two proofs of this theorem. The first proof connects the block LC method with the basic LC method, and follows from the proof of Lemma 6.2 below. The second proof of Theorem 6.1 is in Appendix B.

Lemma 6.2 *Consider a BTF of a Jacobian pattern* $\mathbf{S}_0$ *derived from* $\boldsymbol{\Sigma}$ *and* $(\mathbf{c};\mathbf{d})$. *If we perform the LC conversion as described in Lemma 6.2, then in the resulting* $\overline{\boldsymbol{\Sigma}}$,

$$d_j - c_i \begin{cases} > \overline{\sigma}_{ij} & \text{if } blockOf(j) < blockOf(i) \\ \geq \overline{\sigma}_{ij} & \text{if } blockOf(j) \geq blockOf(i)\ . \end{cases} \tag{6.6}$$

Proof. We only replace f_l by $\overline{f}_l = \overline{f}$ in a conversion, so $\overline{\sigma}_{ij} = \sigma_{ij}$ for all $i \neq l$ and all j. By (6.1), (6.6) holds for all $i \neq l$.

When $i = l$, we consider two cases: (a) blockOf$(j) < q$, and (b) blockOf$(j) \geq q$.

(a) blockOf$(j) < q =$ blockOf(l). By (6.1), $\sigma_{lj} < d_j - c_l$. Then

$$\begin{aligned} \overline{\sigma}_{lj} &= \sigma\left(x_j, \overline{f}_l\right) \\ &= \sigma\left(x_j, \sum_{i\in I} u_i f_i^{(c_i-\underline{c})}\right) \leq \max\left\{\sigma\left(x_j, \mathbf{u}\right), \max_{i\in I}\sigma\left(x_j, f_i^{(c_i-\underline{c})}\right)\right\}\ . \end{aligned} \tag{6.7}$$

We have

$$\begin{aligned} \sigma\left(x_j, \mathbf{u}\right) \leq \sigma\left(x_j, \mathbf{J}_{qq}\right) &\leq \max_{i\in I}\sigma\left(x_j, f_i\right) = \max_{i\in I}\sigma_{ij} \\ &< \max_{i\in I}(d_j - c_i) = d_j - \min_{i\in I} c_i = d_j - c_l \quad \text{and} \end{aligned} \tag{6.8a}$$

$$\max_{i\in I}\sigma\left(x_j, f_i^{(c_i-\underline{c})}\right) = \max_{i\in I}\left(\sigma_{ij} + c_i - \underline{c}\right) < d_j - \underline{c} = d_j - c_l. \tag{6.8b}$$

Using (6.8a) and (6.8b) in (6.7), we obtain $\overline{\sigma}_{lj} < d_j - c_l$.

(b) blockOf$(j) \geq q =$ blockOf(l). By (6.1), $\sigma_{lj} \leq d_j - c_l$. We can replace the two "$<$" in (6.8) by "$\leq$". Using these inequalities in (6.7), we obtain $\overline{\sigma}_{lj} \leq d_j - c_l$. □

Now we prove Theorem 6.1.

Proof. Let $\overline{T}$ be a transversal of $\overline{\mathbf{\Sigma}}$. Using Lemma 6.2 and (6.6), we derive

$$\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} \le \sum_{(i,j)\in\overline{T}} (d_j - c_i) = \sum_{j=1}^{n} d_j - \sum_{i=1}^{n} c_i = \mathrm{Val}(\mathbf{\Sigma}). \tag{6.9}$$

By Lemma 2.11, we can regard block q as a sub-DAE, with its signature matrix $\mathbf{\Sigma}_{qq}$ and offset pair $(\mathbf{c}_q; \mathbf{d}_q)$. The conversion described in Theorem 6.1 can be considered as an application of the basic LC method to this sub-DAE. Since the block LC condition (6.4) holds, that is, $\sigma(x_j, \widehat{\mathbf{u}}) < d_j - \underline{c}$ for all $j \in B_q$ that belong to this sub-DAE, the basic LC condition (4.2) also holds for the block q sub-DAE. Hence $\mathrm{Val}(\overline{\mathbf{\Sigma}}_{qq}) < \mathrm{Val}(\mathbf{\Sigma}_{qq})$.

We prove $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ by contradiction. Assume $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma}) = \sum_{j=1}^{n} d_j - \sum_{i=1}^{n} c_i \ge 0$. Also, (6.6) holds by Lemma 6.2. So the three conditions in Lemma 2.10 are satisfied. By this lemma, the Jacobian patterns $\overline{\mathbf{S}}_0$, derived from $\overline{\mathbf{\Sigma}}$ and $(\mathbf{c}; \mathbf{d})$, and $\mathbf{S}_0$, derived from $\mathbf{\Sigma}$ and $(\mathbf{c}; \mathbf{d})$, are in the same $p \times p$ BTF.

Let T be a HVT of $\mathbf{\Sigma}$. A HVT $\overline{T}$ of $\overline{\mathbf{\Sigma}}$ is the union of HVTs $\overline{T}_w$ of all diagonal blocks $\overline{\mathbf{\Sigma}}_{ww}$, $w = 1:p$. By the construction of $\overline{\mathbf{\Sigma}}$, we have $\mathrm{Val}(\overline{\mathbf{\Sigma}}_{ww}) = \mathrm{Val}(\mathbf{\Sigma}_{ww})$ for all $w \ne q$. Then a contradiction follows from

$$\begin{aligned}
\mathrm{Val}(\overline{\mathbf{\Sigma}}) &= \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} = \sum_{w=1}^{p} \sum_{(i,j)\in\overline{T}_w} \overline{\sigma}_{ij} = \sum_{w=1}^{p} \mathrm{Val}(\overline{\mathbf{\Sigma}}_{ww}) \\
&= \sum_{w\ne q} \mathrm{Val}(\overline{\mathbf{\Sigma}}_{ww}) + \mathrm{Val}(\overline{\mathbf{\Sigma}}_{qq}) < \sum_{w\ne q} \mathrm{Val}(\mathbf{\Sigma}_{ww}) + \mathrm{Val}(\mathbf{\Sigma}_{qq}) \\
&= \sum_{w=1}^{p} \mathrm{Val}(\mathbf{\Sigma}_{ww}) = \sum_{w=1}^{p} \sum_{(i,j)\in T_w} \sigma_{ij} = \sum_{(i,j)\in T} \sigma_{ij} = \mathrm{Val}(\mathbf{\Sigma}).
\end{aligned} \tag{6.10}$$

Hence, the assumption $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma})$ is erroneous. By (6.9), only $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ can hold. □

Example 6.3 We illustrate the block LC method with the Campbell-Griepentrog two-link robot arm DAE in [5]. We slightly simplify the original first-order problem formulation to (6.11), in which x_1, x_2, x_3 occur of second order and x_1', x_2', and x_3' occur implicitly. The two state variables u_1 and u_2 in the original formulation are renamed x_4 and x_5, respectively, and hence are not to be confused with entries in a vector $\mathbf{u}$ in our notation.

$$
\begin{aligned}
0 = A = x_1'' - \Big[& 2c(x_3)(x_1' + x_3')^2 + x_1'^2 d(x_3) + (2x_3 - x_2)\big(a(x_3) + 2b(x_3)\big) \\
& + a(x_3)(x_4 - x_5)\Big] \\
0 = B = x_2'' - \Big[& -2c(x_3)(x_1' + x_3')^2 - x_1'^2 d(x_3) + (2x_3 - x_2)\big(1 - 3a(x_3) - 2b(x_3)\big) \\
& - a(x_3)x_4 + \big(a(x_3) + 1\big)x_5\Big] \\
0 = C = x_3'' - \Big[& -2c(x_3)(x_1' + x_3')^2 - x_1'^2 d(x_3) + (2x_3 - x_2)\big(a(x_3) - 9b(x_3)\big) \\
& - 2x_1'^2 c(x_3) - d(x_3)\big(x_1' + x_3'\big)^2 - \big(a(x_3) + b(x_3)\big)(x_4 - x_5)\Big] \\
0 = D = & \cos x_1 + \cos(x_1 + x_3) - p_1(t) \\
0 = E = & \sin x_1 + \sin(x_1 + x_3) - p_2(t),
\end{aligned}
\tag{6.11}
$$

where

$$
\begin{aligned}
a(\theta) &= 2/(2 - \cos^2\theta) & b(\theta) &= \cos\theta/(2 - \cos^2\theta) \\
c(\theta) &= \sin\theta/(2 - \cos^2\theta) & d(\theta) &= \sin\theta\cos\theta/(2 - \cos^2\theta) \\
p_1(t) &= \cos(1 - e^t) + \cos(1 - t) & p_2(t) &= \sin(1 - e^t) + \sin(1 - t).
\end{aligned}
$$

$$\boldsymbol{\Sigma} = \begin{array}{c|ccccc|c} & x_2 & x_4 & x_5 & x_1 & x_3 & c_i \\ f_1\ B & 2^{\bullet} & 0 & 0 & 1 & 1 & 0 \\ f_2\ C & 0 & 0^{\bullet} & 0 & 1 & 2 & 0 \\ f_3\ A & 0 & 0 & 0^{\bullet} & 2 & 1 & 0 \\ f_4\ D & & & & 0^{\bullet} & 0 & 2 \\ f_5\ E & & & & 0 & 0^{\bullet} & 2 \\ \hline d_j & 2 & 0 & 0 & 2 & 2 & \end{array} \qquad \mathbf{J} = \begin{array}{c|ccccc} & x_2'' & x_4 & x_5 & x_1'' & x_3'' \\ B & 1 & a_3 & -a_3-1 & & \\ C & & a_3+b_3 & -a_3-b_3 & & 1 \\ A & & -a_3 & a_3 & 1 & \\ D'' & & & & \frac{\partial D}{\partial x_1} & \frac{\partial D}{\partial x_3} \\ E'' & & & & \frac{\partial E}{\partial x_1} & \frac{\partial E}{\partial x_3} \end{array}$$

Here, in $\mathbf{J}$,

$$a_3 = a(x_3) = 2/(2-\cos^2 x_3) \qquad b_3 = b(x_3) = \cos x_3/(2-\cos^2 x_3)$$

$$\partial D/\partial x_1 = -\sin x_1 - \sin(x_1+x_3) \qquad \partial D/\partial x_3 = -\sin(x_1+x_3)$$

$$\partial E/\partial x_1 = \cos x_1 + \cos(x_1+x_3) \qquad \partial E/\partial x_3 = \cos(x_1+x_3).$$

The DAE (6.11) is of differentiation index 5, while the SA reports structural index $\nu_S = 3$. Hence this must be a failure case, because ν_S is an upper bound for the differentiation index when the SA succeeds [42]. We can see that the sub-Jacobian $\mathbf{J}_{22}$ of block 2 is identically singular.

The block LC method first computes $\widehat{\mathbf{u}} = [2, 2+\cos x_3]^T \in \operatorname{coker}(\mathbf{J}_{22})$. Then $\mathbf{u} = [0, 2, 2+\cos x_3, 0, 0]^T$. Note that $\mathbf{u} \notin \operatorname{coker}(\mathbf{J})$. Using (6.3), we have

$$I = \{\, i \mid u_i \not\equiv 0 \,\} = \{\, 2, 3 \,\}, \quad \underline{c} = \min_{i \in I} c_i = 0, \quad L = \{\, 2, 3 \,\}, \quad \text{and} \quad \overline{L} = \{\, 2 \,\}.$$

The variables x_4 and x_5 in block 2 do not occur in $\mathbf{u}$, so the condition (6.4) is satisfied.

Considering equivalence, we pick $l = 2 \in \overline{L}$ over $l = 3 \in L \setminus \overline{L}$, and replace $f_l = C$ by $\overline{C} = u_1 C + u_2 A = 2C + (2 + \cos x_3)A$. According to the proof of Theorem 6.1, neither of x_4 and x_5 occurs in $\overline{C}$. The SA results of the resulting DAE are as follows.

$$\overline{\mathbf{\Sigma}} = \begin{array}{c|ccccc|c} & x_4 & x_5 & x_2 & x_1 & x_3 & c_i \\ A & 0^{\bullet} & 0 & 0 & 2 & 1 & 0 \\ B & 0 & 0^{\bullet} & 2 & 1 & 1 & 0 \\ \overline{C} & & & 0^{\bullet} & 2 & 2 & 2 \\ D & & & & 0^{\bullet} & 0 & 4 \\ E & & & & 0 & 0^{\bullet} & 4 \\ \hline d_j & 0 & 0 & 2 & 4 & 4 & \end{array} \quad \overline{\mathbf{J}} = \begin{array}{c|ccccc} & x_4 & x_5 & x_2'' & x_1^{(4)} & x_3^{(4)} \\ A & -a_3 & a_3 & & & \\ B & a_3 & -a_3 - 1 & 1 & & \\ \overline{C}'' & & & \frac{\partial \overline{C}}{\partial x_2} & 2 + \cos x_3 & 2 \\ D^{(4)} & & & & \frac{\partial D}{\partial x_1} & \frac{\partial D}{\partial x_3} \\ E^{(4)} & & & & \frac{\partial E}{\partial x_1} & \frac{\partial E}{\partial x_3} \end{array}$$

Here $\partial \overline{C}/\partial x_2 = 2(a_3^2 - 3a_3 b_3 + b_3^2)(2 - \cos^2 x_3)$. The SA reports $\nu_S = 5$, and succeeds at any point where $\det(\overline{\mathbf{J}}) = 4(a_3^2 - 3a_3 b_3 + b_3^2)\sin x_3 \neq 0$. Now $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 0 < 2 = \mathrm{Val}(\mathbf{\Sigma})$. □

6.3 Block expression substitution method

Assume that a $\mathbf{J}_{qq}$ is identically singular. Let $\widehat{\mathbf{v}} \in \ker(\mathbf{J}_{qq})$ and $\widehat{\mathbf{v}} \not\equiv \mathbf{0}_{N_q}$. Similarly, we construct

$$\mathbf{v} = \begin{bmatrix} \mathbf{0}_{N_1 + \cdots + N_{q-1}} \\ \widehat{\mathbf{v}} \\ \mathbf{0}_{N_{q+1} + \cdots + N_p} \end{bmatrix}.$$

We use notation similar to that used in the basic ES method:

$$\begin{aligned} &J = \big\{\, j \mid v_j \not\equiv 0 \,\big\} \subseteq B_q, \quad M = \big\{\, i \in B_q \mid d_j - c_i = \sigma_{ij} \text{ for some } j \in J \,\big\}, \\ &s = |J|, \quad \overline{c} = \max_{i \in M} c_i \quad \text{and} \\ &\overline{J} = \big\{\, l \mid v_l \text{ is (nonzero) constant} \,\big\}. \end{aligned} \tag{6.12}$$

The set $\overline{J}$ is used to seek a conversion that guarantees equivalence in the original DAE and the converted one.

The block ES conditions are

$$\begin{aligned} \sigma\left(x_j, \mathbf{v}\right) &\begin{cases} < d_j - \overline{c} & \text{if } j \in J \text{ or } \mathrm{blockOf}(j) < q \\ \leq d_j - \overline{c} & \text{if } j \in B_q \setminus J \text{ or } \mathrm{blockOf}(j) > q \ , \end{cases} \\ d_j - \overline{c} \geq 0 & \qquad \text{for all } j \in J. \end{aligned} \tag{6.13}$$

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

$$y_j = x_j^{(d_j - \overline{c})} - \frac{v_j}{v_l} \cdot x_l^{(d_l - \overline{c})} \quad \text{for all } j \in J \setminus \big\{\, l \,\big\}. \tag{6.14}$$

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

$$\begin{aligned} \text{replace each} \quad & x_j^{(\sigma_{ij})} \quad \text{with} \quad d_j - c_i = \sigma_{ij} \quad \text{and} \quad j \in J \setminus \big\{\, l \,\big\} \\ \text{by} \quad & \left(y_j + \frac{v_j}{v_l} \cdot x_l^{(d_l - \overline{c})} \right)^{(\overline{c} - c_i)}. \end{aligned} \tag{6.15}$$

Note that, because of M in (6.12), we actually perform expression substitutions in only f_i's with $i \in M \subseteq B_q$. Denote each new f_i by $\overline{f}_i$, and let also $\overline{f}_i = f_i$ for the unchanged equations with $i \notin M$.

By (6.14), we append $s-1$ equations that prescribe the substitutions in (6.15):

$$0 = g_j = -y_j + x_j^{(d_j-\overline{c})} - \frac{v_j}{v_l} \cdot x_l^{(d_l-\overline{c})} \quad \text{for all } j \in J \setminus \{\, l \,\}. \tag{6.16}$$

The block ES method is based on the following theorem.

Theorem 6.4 *Let J, s, M, and $\overline{c}$ be as defined in (6.12). Assume that the block ES conditions (6.13) hold. For an $l \in J$, if we*

1) introduce $s-1$ new variables y_j, $j \in J \setminus \{\, l \,\}$, as defined in (6.14),

2) perform replacements in f_i, for all $i \in B_q$, as described in (6.15), and

3) append $s-1$ equations g_j, $j \in J \setminus \{\, l \,\}$, as defined in (6.16),

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

Before proving this theorem, we illustrate the block ES method with (6.11), and prove two related lemmas.

Example 6.5 This method finds first $\widehat{\mathbf{v}} = [1,1]^T \in \ker(\mathbf{J}_{22})$. Then $\mathbf{v} = [0,0,1,1,0]^T$. Note that $\mathbf{v} \notin \ker(\mathbf{J})$. Using (6.12), we have

$$J = \overline{J} = \{\, j \mid v_j \not\equiv 0 \,\} = \{\, 2,3 \,\}, \quad s = |J| = 2, \quad M = \{\, 2,3 \,\}, \quad \overline{c} = \max_{i \in M} c_i = 0.$$

Since $\mathbf{v}$ is constant, $J = \overline{J}$ and the first condition in (6.13) holds. The second condition holds also, as $d_4 - \overline{c} = d_5 - \overline{c} = 0$. We choose x_4, whose column index in the permuted $\boldsymbol{\Sigma}$ is $l = 2 \in \overline{J}$. Then we introduce for x_5, the other variable in block

2 with column index $j = 3$, a new variable

$$y_5 = x_5^{(d_5-\bar{c})} - \frac{v_3}{v_2} \cdot x_4^{(d_4-\bar{c})} = x_5 - x_4.$$

Correspondingly, we append $0 = g_5 = -y_5 + x_5 - x_4$ and replace x_5 by $y_5 + x_4$ in C and A, the equations in block 2.

The resulting DAE has the following new equations

$$\begin{aligned}
0 = \overline{A} &= x_1'' - \Big[2c(x_3)(x_1' + x_3')^2 + x_1'^2 d(x_3) + (2x_3 - x_2)\big(a(x_3) + 2b(x_3)\big) + a(x_3)y_5\Big] \\
0 = \overline{C} &= x_3'' - \Big[-2c(x_3)(x_1' + x_3')^2 - x_1'^2 d(x_3) + (2x_3 - x_2)\big(a(x_3) - 9b(x_3)\big) \\
&\qquad - 2x_1'^2 c(x_3) - d(x_3)\big(x_1' + x_3'\big)^2 - \big(a(x_3) + b(x_3)\big)y_5\Big] \\
0 &= g_5 = -y_5 + x_4 - x_5.
\end{aligned}$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c|cccccc|c}
 & x_4 & x_5 & x_2 & y_5 & x_1 & x_3 & c_i \\
g_5 & 0^\bullet & 0 & & 0 & & & 0 \\
B & 0 & 0^\bullet & 2 & & 1 & 1 & 0 \\
\overline{C} & & & 0^\bullet & 0 & 1 & 2 & 2 \\
\overline{A} & & & 0 & 0^\bullet & 2 & 1 & 2 \\
D & & & & & 0^\bullet & 0 & 4 \\
E & & & & & 0 & 0^\bullet & 4 \\
\hline
d_j & 0 & 0 & 2 & 2 & 4 & 4 &
\end{array}$$

$$\overline{\mathbf{J}} = \begin{array}{c|cc|ccc|cc}
 & x_4 & x_5 & x_2'' & y_5'' & & x_1^{(4)} & x_3^{(4)} \\
g_5 & -1 & 1 & & & & & \\
B & a_3 & -a_3 - 1 & 1 & & & & \\
\overline{C}'' & & & a_3 - 9b_3 & -a_3 - b_3 & & & 1 \\
\overline{A}'' & & & a_3 + 2b_3 & a_3 & & 1 & \\
D^{(4)} & & & & & & \frac{\partial D}{\partial x_1} & \frac{\partial D}{\partial x_3} \\
E^{(4)} & & & & & & \frac{\partial E}{\partial x_1} & \frac{\partial E}{\partial x_3}
\end{array}$$

Now the System Jacobian $\overline{\mathbf{J}}$ is generically nonsingular. The SA reports the correct index 5, and succeeds at any point where $\det(\overline{\mathbf{J}}) = 2(a_3^2 - 3a_3b_3 + b_3^2)\sin x_3 \neq 0$. Again $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 0 < 2 = \mathrm{Val}(\mathbf{\Sigma})$. □

In [41, Example 6.1], Pryce fixed the SA's failure on (6.11), and pointed out that *only* the introduction of $x_4 - x_5$ as a separate variable is essential to his fix. Example 6.5 verifies Pryce's argument and shows that the block ES method finds his reformulation in a systematic way.

To prove Theorem 6.4, we shall use the following two assumptions.

(a) Without loss of generality, we assume that the entries $\widehat{v}_j \not\equiv 0$ are in the first s positions of $\widehat{\mathbf{v}}$, that is, $\widehat{\mathbf{v}} = [\widehat{v}_1, \ldots, \widehat{v}_s, 0, \ldots, 0]^T$. By (6.12),

$$J = \sum_{w=1}^{q-1} N_w + 1 : \sum_{w=1}^{q-1} N_w + s.$$

(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 $0 = g_l = -y_l + x_l^{(d_l - \overline{c})}$.

We show first that the signature matrix $\overline{\mathbf{\Sigma}}$ of the resulting DAE can be put in the block structure as shown in Figure 6.1. Then we construct two $(n+s)$-vectors $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$ in (6.17), and prove in Lemma 6.6 that $\widetilde{d}_j - \widetilde{c}_i > \overline{\sigma}_{ij}$ holds in the below diagonal blocks, while $\widetilde{d}_j - \widetilde{c}_i \geq \overline{\sigma}_{ij}$ holds elsewhere. The proof of this lemma is rather technical, so we present it in Appendix A.2. Lastly, we prove Theorem 6.4.

From the description of the ES conversion in Theorem 6.4, the substitutions (6.15) only occur in equations f_i with $i \in B_q$. Hence, in the resulting DAE, variables y_j for $j \in J$ only appear in equations $\overline{f}_i$ for $i \in M \subseteq B_q$ and equations g_r for $r \in J$.

Considering the block structure of $\overline{\mathbf{\Sigma}}$ in Figure 6.1, we elaborate on four cases for a block submatrix $\overline{\mathbf{\Sigma}}_{w_1 w_2}$: (a) $w_1 \neq q$ and $w_2 \neq q$, (b) $w_1 \neq q$ and $w_2 = q$, (c) $w_1 = q$ and $w_2 \neq q$, and (d) $w_1 = w_2 = q$.

(a) $w_1 \neq q$ and $w_2 \neq q$. In $\overline{\mathbf{\Sigma}}_{w_1 w_2}$, equations f_i are of indices $i \in B_{<q} \cup B_{>q}$. As noted

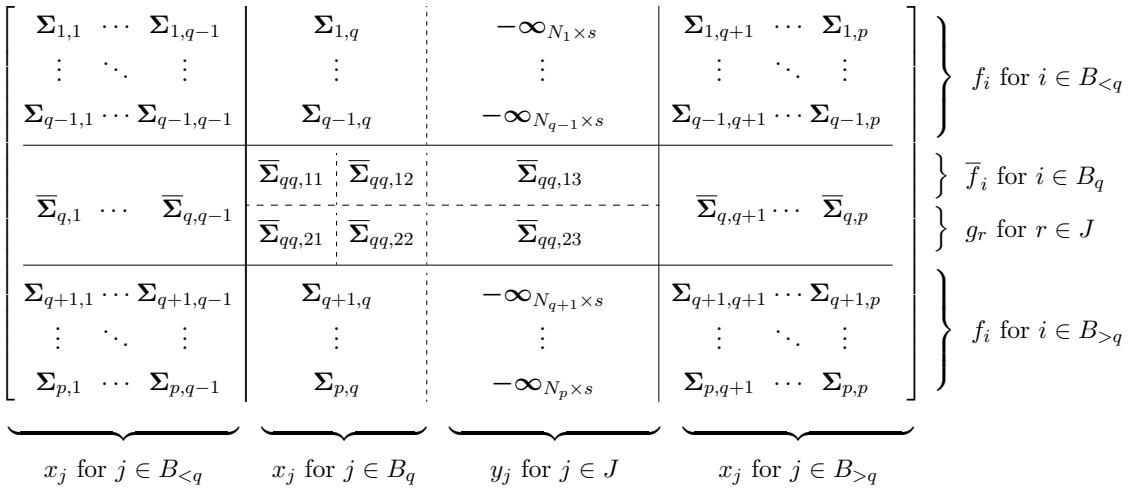


Figure 6.1: Block structure of $\overline{\boldsymbol{\Sigma}}$ of the resulting DAE by the block ES method. The notation $B_{<q}$ is short for $\cup_{w=1}^{q-1} B_w$, and $B_{>q}$ is short for $\cup_{w=q+1}^{p} B_w$.

in (6.12), the expression substitutions described in (6.15) only take place in $f_{i'}$ with $i' \in M \subseteq B_q$, so do not happen in such blocks $\overline{\boldsymbol{\Sigma}}_{w_1 w_2}$. Hence, each $\boldsymbol{\Sigma}_{w_1 w_2}$ remains unchanged in $\overline{\boldsymbol{\Sigma}}$, and

$$\overline{\boldsymbol{\Sigma}}_{w_1 w_2} = \boldsymbol{\Sigma}_{w_1 w_2} \quad \text{for } w_1 \neq q \text{ and } w_2 \neq q.$$

(b) $w_1 \neq q$ and $w_2 = q$. In $\overline{\boldsymbol{\Sigma}}_{w_1 q}$, we include variables y_j for $j \in J$ as defined in (6.14). By the same arguments as in (a), the expression substitutions do not happen in these blocks. That is, y_j for $j \in J$ do not appear in equations f_i for $i \in B_{<q} \cup B_{>q}$. Hence, we can obtain $\overline{\boldsymbol{\Sigma}}_{w_1 q}$ by concatenating horizontally $\boldsymbol{\Sigma}_{w_1 q}$ with an $N_{w_1} \times s$ matrix of $-\infty$'s:

$$\overline{\boldsymbol{\Sigma}}_{w_1 q} = \begin{bmatrix} \boldsymbol{\Sigma}_{w_1 q} & -\boldsymbol{\infty}_{N_{w_1} \times s} \end{bmatrix}, \qquad w_1 = 1, \ldots, q-1, q+1, \ldots, p.$$

(c) $w_1 = q$ and $w_2 \neq q$. In $\overline{\mathbf{\Sigma}}_{qw_2}$, we include equations g_r for $r \in J$ as defined in (6.16). Also, due to the expression substitutions (6.15) occurring in f_i with $i \in M \subseteq B_q$, $\sigma(x_j, f_i)$ and $\sigma\left(x_j, \overline{f}_i\right)$ may not be the same for $i \in B_q$ and all $j = 1:n$. Hence, in contrast to cases (a) and (b), there are no obvious connections between $\mathbf{\Sigma}_{w_1 w_2}$ and $\overline{\mathbf{\Sigma}}_{w_1 w_2}$ for $w_1 = q$ and $w_2 \neq q$.

(d) $w_1 = w_2 = q$. $\overline{\mathbf{\Sigma}}_{qq}$ contains signature entries for equations $\overline{f}_i$ and g_r, where $i \in B_q$ and $r \in J$, in variables x_j and y_r, where $j \in B_q$ and $r \in J$. Similar to $\overline{\mathbf{\Sigma}}$ in the basic ES method (cf. Figure 4.1 and (A.4) in Appendix A.1), $\overline{\mathbf{\Sigma}}_{qq}$ in the block ES method also has a (sub)block structure

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

We shall use it in the proof of Lemma 6.6 in Appendix A.2.

Denote by $Q = \sum_{w=1}^{q} N_w$ the total number of equations (or variables) in the first q blocks of $\mathbf{\Sigma}$. Using a valid offset pair $(\mathbf{c}; \mathbf{d})$ of $\mathbf{\Sigma}$, we construct two $(n+s)$-vectors $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$, defined as

$$\widetilde{c}_i = \begin{cases} c_i & \text{if } i = 1:Q \\ \overline{c} & \text{if } i = Q+1:Q+s \\ c_{i-s} & \text{if } i = Q+s+1:n+s, \end{cases} \qquad \widetilde{d}_j = \begin{cases} d_j & \text{if } j = 1:Q \\ \overline{c} & \text{if } j = Q+1:Q+s \\ d_{j-s} & \text{if } j = Q+s+1:n+s. \end{cases} \tag{6.17}$$

Then we have the following lemma.

Lemma 6.6 *In the block structure of* $\overline{\mathbf{\Sigma}}$ *in Figure 6.1,*

$$\widetilde{d}_j - \widetilde{c}_i \begin{cases} > \overline{\sigma}_{ij} & \text{if } (i,j) \text{ is in a below diagonal block, and} \\ \geq \overline{\sigma}_{ij} & \text{otherwise.} \end{cases}$$

The proof of this lemma is in Appendix A.2. Using this lemma, we can now prove Theorem 6.4.

Proof. Let $\overline{T}$ be a transversal of $\overline{\mathbf{\Sigma}}$. Using Lemma 6.6 and (6.17), we derive

$$\begin{aligned} \mathrm{Val}(\overline{\mathbf{\Sigma}}) &= \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} \\ &\leq \sum_{(i,j)\in\overline{T}} (\widetilde{d}_j - \widetilde{c}_i) \\ &= \sum_{j=1}^{n+s} \widetilde{d}_j - \sum_{i=1}^{n+s} \widetilde{c}_i \\ &= \left(\sum_{j=1}^{Q} d_j + s\overline{c} + \sum_{j=Q+s+1}^{n+s} d_{j-s} \right) - \left(\sum_{j=1}^{Q} c_i + s\overline{c} + \sum_{i=Q+s+1}^{n+s} c_{i-s} \right) \\ &= \sum_{j=1}^{n} d_j - \sum_{i=1}^{n} c_i = \mathrm{Val}(\mathbf{\Sigma}). \end{aligned} \tag{6.18}$$

Again, as in the proof of Theorem 6.1, we prove $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ by contradiction. Assume $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma})$. Obviously $\mathrm{Val}(\mathbf{\Sigma}) \geq 0$. The vectors $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$ in Lemma 6.6 satisfy conditions *(i)* and *(ii)* of Lemma 2.10. Also, $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \sum_j^{n+s} \widetilde{d}_j - \sum_i^{n+s} \widetilde{c}_i$ satisfies condition *(iii)* of Lemma 2.10. It follows from this lemma that

(a) $(\widetilde{\mathbf{c}}; \widetilde{\mathbf{d}})$ is a valid offset pair of $\overline{\mathbf{\Sigma}}$.

(b) The Jacobian pattern $\overline{\mathbf{S}}_0$, derived from $\overline{\mathbf{\Sigma}}$ and $(\widetilde{\mathbf{c}}; \widetilde{\mathbf{d}})$, is in the $p \times p$ BTF shown in Figure 6.1.

(c) $\overline{T}$ is the union of HVTs $\overline{T}_w$ of all diagonal blocks $\overline{\mathbf{\Sigma}}_{11}, \ldots, \overline{\mathbf{\Sigma}}_{pp}$ of $\overline{\mathbf{\Sigma}}$.

We can consider block q of the original DAE as a sub-DAE, with signature matrix $\mathbf{\Sigma}_{qq}$ and offset pair $(\mathbf{c}_q; \mathbf{d}_q)$—this follows from Lemma 2.11. The ES conversion described in Theorem 6.4 can be regarded as an application of the basic ES method to this sub-DAE, given that the basic ES conditions (4.19) hold due to the block ES conditions (6.13). By Theorem 4.17 for the basic ES method, a conversion results in $\mathrm{Val}(\overline{\mathbf{\Sigma}}_{qq}) < \mathrm{Val}(\mathbf{\Sigma}_{qq})$. Also, since $\overline{\mathbf{\Sigma}}_{ww} = \mathbf{\Sigma}_{ww}$ for $w \neq q$, $\mathrm{Val}(\overline{\mathbf{\Sigma}}_{ww}) = \mathrm{Val}(\mathbf{\Sigma}_{ww})$ when $w \neq q$. Then a contradiction $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ follows, if we apply the same arguments as in (6.10).

Hence, the assumption $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma})$ is erroneous. By (6.18), only $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$ can hold. □

Remark 6.7 For clarification, we revisit the three items (a)—(c) in the above proof of Theorem 6.4. Since

$$\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma}) = \sum_{j=1}^{n+s} \widetilde{d}_j - \sum_{i=1}^{n+s} \widetilde{c}_i$$

by (6.17) and Theorem 6.4, an offset pair $(\widetilde{\mathbf{c}}; \widetilde{\mathbf{d}})$ cannot be valid for $\overline{\mathbf{\Sigma}}$, so (a) *must* not hold. Regarding (b) and (c), it may be possible that a Jacobian pattern $\overline{\mathbf{S}}_0$, derived from $\overline{\mathbf{\Sigma}}$ and its valid offset pair $(\overline{\mathbf{c}}; \overline{\mathbf{d}})$, is in the $p \times p$ BTF shown in Figure 6.1. In this case, (b) holds and then (c) follows from it. However, in general, the block structure in Figure 6.1 is not necessarily a BTF of the resulting DAE.

Chapter 7

Examples of block conversion methods

We illustrate the block conversion methods with two DAE problems originated from electrical circuit analysis [27]. We discuss the transistor amplifier DAE in §7.1, and discuss the ring modulator DAE in §7.2.

We also illustrate in §7.3 how to fix the index overestimation problem on the family of DAEs by Reißig, for which SA produces a nonsingular System Jacobian but overestimates the index. We apply a technique similar to the block LC method, and then SA reports the correct index $\nu_S = 1$.

Lastly, we summarize in §7.4 our treatments for the SA-unfriendly DAEs discussed in this thesis.

7.1 Transistor amplifier DAE

The transistor amplifier DAE is classified as a stiff index-1 DAE in [27]:

$$
\begin{aligned}
0 = f_1 = \quad & C_1(x_1' - x_2') + R_0^{-1}(x_1 - U_e(t)) \\
0 = f_2 = -& C_1(x_1' - x_2') - R_2^{-1}U_b + x_2\left(R_1^{-1} + R_2^{-1}\right) - (\alpha - 1)g(x_2 - x_3) \\
0 = f_3 = \quad & C_2 x_3' - g(x_2 - x_3) + R_3^{-1}x_3 \\
0 = f_4 = \quad & C_3(x_4' - x_5') + R_4^{-1}(x_4 - U_b) + \alpha g(x_2 - x_3) \\
0 = f_5 = -& C_3(x_4' - x_5') - R_0^{-1}U_b + x_5\left(R_5^{-1} + R_6^{-1}\right) - (\alpha - 1)g(x_5 - x_6) \\
0 = f_6 = \quad & C_4 x_6' - g(x_5 - x_6) + R_7^{-1}x_6 \\
0 = f_7 = \quad & C_5(x_7' - x_8') + R_8^{-1}(x_7 - U_b) + \alpha g(x_5 - x_6) \\
0 = f_8 = -& C_5(x_7' - x_8') + R_9^{-1}x_8,
\end{aligned}
\tag{7.1}
$$

where

$$
\begin{aligned}
&g(y) = \beta\left(e^{y/U_F} - 1\right) && U_b = 6.0 && R_0 = 1000 && \\
&\alpha = 0.99 && U_F = 0.026 && R_k = 9000 && k = 1:9 \\
&\beta = 10^{-6} && U_e(t) = 0.1\sin(200\pi t) && C_k = k \times 10^{-6} && k = 1:5.
\end{aligned}
$$

The SA fails since $\det(\mathbf{J}) \equiv 0$. The fine BTF reveals that the three 2×2 sub-Jacobians $\mathbf{J}_{11}$, $\mathbf{J}_{33}$, $\mathbf{J}_{55}$ are identically singular and have a similar structure. Each block receives the same treatment when a conversion method is applied.

$\mathbf{\Sigma} =$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	c_i
f_1	$1^\bullet$	1							0
f_2	1	$1^\bullet$	0						0
f_3		0	$1^\bullet$						0
f_4		0	0	$1^\bullet$	1				0
f_5				1	$1^\bullet$	0			0
f_6					0	$1^\bullet$			0
f_7					0	0	$1^\bullet$	1	0
f_8							1	$1^\bullet$	0
d_j	1	1	1	1	1	1	1	1	

$\mathbf{J} =$

	x'_1	x'_2	x'_3	x'_4	x'_5	x'_6	x'_7	x'_8	c_i
f_1	C_1	$-C_1$							0
f_2	$-C_1$	C_1							0
f_3			C_2						0
f_4				C_3	$-C_3$				0
f_5				$-C_3$	C_3				0
f_6						C_4			0
f_7							C_5	$-C_5$	0
f_8							$-C_5$	C_5	0

LC method. One can easily find $\widehat{\mathbf{u}} = [1,1]^T \in \operatorname{coker}(\mathbf{J}_{11}), \operatorname{coker}(\mathbf{J}_{33}), \operatorname{coker}(\mathbf{J}_{55})$. We perform on each singular block a conversion, and choose to replace the first equation in each such block.

block	replace	by
1	f_1	$\overline{f}_1 = f_1 + f_2$
3	f_4	$\overline{f}_4 = f_4 + f_5$
5	f_7	$\overline{f}_7 = f_7 + f_8$

The new equations in the resulting DAE are

$$\begin{aligned}
0 = \overline{f}_1 &= R_0^{-1}(x_1 - U_e(t)) - R_2^{-1}U_b + x_2\left(R_1^{-1} + R_2^{-1}\right) - (\alpha - 1)g(x_2 - x_3) \\
0 = \overline{f}_4 &= R_4^{-1}(x_4 - U_b) + \alpha g(x_2 - x_3) - R_5^{-1}U_b + x_5\left(R_5^{-1} + R_6^{-1}\right) \\
&\qquad - (\alpha - 1)g(x_5 - x_6) \\
0 = \overline{f}_7 &= R_8^{-1}(x_7 - U_b) + \alpha g(x_5 - x_6) + R_9^{-1}x_8.
\end{aligned}$$

The SA still reports index 1, and succeeds with a nonzero constant $\det(\overline{\mathbf{J}})$:

$$\det(\overline{\mathbf{J}}) = C_1C_2C_3C_4C_5\left(R_0^{-1} + R_1^{-1} + R_2^{-1}\right)\left(R_4^{-1} + R_5^{-1} + R_6^{-1}\right)\left(R_8^{-1} + R_9^{-1}\right) \neq 0.$$

Now $\operatorname{Val}(\overline{\mathbf{\Sigma}}) = 5 < 8 = \operatorname{Val}(\mathbf{\Sigma})$.

$\overline{\mathbf{\Sigma}} =$

	x_7	x_8	x_4	x_5	x_6	x_1	x_2	x_3	c_i
$\overline{f}_7$	$0^\bullet$	0		0	0				1
f_8	1	$1^\bullet$							0
$\overline{f}_4$			$0^\bullet$	0	0		0	0	1
f_5			1	$1^\bullet$	0				0
f_6				0	$1^\bullet$				0
$\overline{f}_1$						$0^\bullet$	0	0	1
f_2						1	$1^\bullet$	0	0
f_3							0	$1^\bullet$	0
d_j	1	1	1	1	1	1	1	1	

$\overline{\mathbf{J}} =$

	x'_7	x'_8	x'_4	x'_5	x'_6	x'_1	x'_2	x'_3
$\overline{f}'_7$	R_8^{-1}	R_9^{-1}		$\frac{\partial \overline{f}_7}{\partial x_5}$	$\frac{\partial \overline{f}_7}{\partial x_6}$			
f_8	$-C_5$	C_5						
$\overline{f}'_4$			R_4^{-1}	$R_5^{-1} + R_6^{-1}$	$\frac{\partial \overline{f}_4}{\partial x_6}$		$\frac{\partial \overline{f}_4}{\partial x_2}$	$\frac{\partial \overline{f}_4}{\partial x_3}$
f_5			$-C_3$	C_3				
f_6					C_4			
$\overline{f}'_1$						R_0^{-1}	$R_1^{-1} + R_2^{-1}$	$\frac{\partial \overline{f}_1}{\partial x_3}$
f_2						$-C_1$	C_1	
f_3								C_2

ES method. We can take $\widehat{\mathbf{v}} = [1,1]^T \in \ker(\mathbf{J}_{11}), \ker(\mathbf{J}_{33}), \ker(\mathbf{J}_{55})$. We show how to perform a conversion on block 1; block 3 and block 5 can be treated in the same way.

For block 1, we construct the corresponding $\mathbf{v} = [1, 1, \mathbf{0}_8^T]^T$. Using (6.12), we have

$$J = \overline{J} = \{\, j \mid v_j \not\equiv 0 \,\} = \{\, 1, 2 \,\}, \quad s = |J| = 2, \quad M = \{\, 1, 2 \,\}, \quad \text{and} \quad \overline{c} = 0.$$

We choose $l = 1 \in \overline{J}$, introduce for x_2 a new variable $y_2 = x_2^{(d_2-\overline{c})} - \frac{v_2}{v_1} \cdot x_1^{(d_1-\overline{c})} = x_2' - x_1'$, and append correspondingly the equation $0 = h_2 = -y_2 + x_2' - x_1'$. Then we replace x_2' by $y_2 + x_1'$ in f_1, f_2.

After we complete similar conversions on block 3 and block 5, the resulting DAE has equations f_3, f_6 and the following equations:

$$\begin{aligned}
0 = \overline{f}_1 &= -C_1 y_2 + R_0^{-1}(x_1 - U_e(t)) \\
0 = \overline{f}_2 &= \quad C_1 y_2 - R_2^{-1} U_b + x_2 \left(R_1^{-1} + R_2^{-1}\right) - (\alpha - 1) g(x_2 - x_3) \\
0 = h_2 &= -y_2 + x_2' - x_1' \\
0 = \overline{f}_4 &= -C_3 y_5 + R_4^{-1}(x_4 - U_b) + \alpha g(x_2 - x_3) \\
0 = \overline{f}_5 &= \quad C_3 y_5 - R_5^{-1} U_b + x_5 \left(R_5^{-1} + R_6^{-1}\right) - (\alpha - 1) g(x_5 - x_6) \\
0 = h_5 &= -y_5 + x_5' - x_4' \\
0 = \overline{f}_7 &= -C_5 y_8 + R_8^{-1}(x_7 - U_b) + \alpha g(x_5 - x_6) \\
0 = \overline{f}_8 &= \quad C_5 y_8 + R_9^{-1} x_8 \\
0 = h_8 &= -y_8 + x_8' - x_7'.
\end{aligned}$$

The SA succeeds with a nonzero constant $\det(\overline{\mathbf{J}})$ and $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 5 < 8 = \mathrm{Val}(\mathbf{\Sigma})$.

$\overline{\mathbf{\Sigma}} =$

	x_7	x_8	y_8	x_4	x_5	y_5	x_6	x_1	x_2	y_2	x_3	c_i
h_8	$1^\bullet$	1	0									0
$\overline{f}_8$		$0^\bullet$	0									1
$\overline{f}_7$	0		$0^\bullet$		0		0					1
h_5				$1^\bullet$	1	0						0
$\overline{f}_5$					$0^\bullet$	0	0					1
$\overline{f}_4$				0		$0^\bullet$			0		0	1
f_6					0		$1^\bullet$					0
h_2								$1^\bullet$	1	0		0
$\overline{f}_2$									$0^\bullet$	0	0	1
$\overline{f}_1$								0		$0^\bullet$		1
f_3									0		$1^\bullet$	0
d_j	1	1	1	1	1	1	1	1	1	1	1	

$\overline{\mathbf{J}} =$

	x_7'	x_8'	y_8'	x_4'	x_5'	y_5'	x_6'	x_1'	x_2'	y_2'	x_3'
h_8	-1	1									
$\overline{f}_8'$		R_9^{-1}	C_5								
$\overline{f}_7'$	R_8^{-1}		$-C_5$		$\frac{\partial \overline{f}_7}{\partial x_5}$		$\frac{\partial \overline{f}_7}{\partial x_6}$				
h_5				-1	1						
$\overline{f}_5'$					$\frac{\partial \overline{f}_5}{\partial x_5}$	C_3	$\frac{\partial \overline{f}_5}{\partial x_6}$				
$\overline{f}_4'$				R_4^{-1}		$-C_3$			$\frac{\partial \overline{f}_4}{\partial x_2}$		$\frac{\partial \overline{f}_4}{\partial x_3}$
f_6							C_4				
h_2								-1	1		
$\overline{f}_2'$									$\frac{\partial \overline{f}_2}{\partial x_2}$	C_1	$\frac{\partial \overline{f}_2}{\partial x_3}$
$\overline{f}_1'$								R_0^{-1}		$-C_1$	
f_3											C_3

7.2 Ring modulator DAE

We study the ring modulator problem from [27]. When $C_s \neq 0$, it is a stiff ODE system of 15 nonlinear equations. Setting $C_s = 0$ gives a DAE of differentiation index 2, which consists of 11 differential and 4 algebraic equations:

$$
\begin{aligned}
0 = f_1 &= -x_1' + C^{-1}\big(x_8 - 0.5x_{10} + 0.5x_{11} + x_{14} - R^{-1}x_1\big)\\
0 = f_2 &= -x_2' + C^{-1}\big(x_9 - 0.5x_{11} + 0.5x_{12} + x_{15} - R^{-1}x_2\big)\\
0 = f_3 &= \quad x_{10} - q(U_{D1}) + q(U_{D4})\\
0 = f_4 &= -x_{11} + q(U_{D2}) - q(U_{D3})\\
0 = f_5 &= \quad x_{12} + q(U_{D1}) - q(U_{D3})\\
0 = f_6 &= -x_{13} - q(U_{D2}) + q(U_{D4})\\
0 = f_7 &= -x_7' + C_p^{-1}\big(-R_p^{-1}x_7 + q(U_{D1}) + q(U_{D2}) - q(U_{D3}) - q(U_{D4})\big)\\
0 = f_8 &= -x_8' + -L_h^{-1}x_1\\
0 = f_9 &= -x_9' + -L_h^{-1}x_2\\
0 = f_{10} &= -x_{10}' + L_{s2}^{-1}(0.5x_1 - x_3 - R_{g2}x_{10})\\
0 = f_{11} &= -x_{11}' + L_{s3}^{-1}(-0.5x_1 + x_4 - R_{g3}x_{11})\\
0 = f_{12} &= -x_{12}' + L_{s2}^{-1}(0.5x_2 - x_5 - R_{g2}x_{12})\\
0 = f_{13} &= -x_{13}' + L_{s3}^{-1}(-0.5x_2 + x_6 - R_{g3}x_{13})\\
0 = f_{14} &= -x_{14}' + L_{s1}^{-1}(-x_1 + U_{in1}(t) - (R_i + R_{g1})x_{14})\\
0 = f_{15} &= -x_{15}' + L_{s1}^{-1}(-x_2 - (R_c + R_{g1})x_{15}).
\end{aligned}
\tag{7.2}
$$

$\boldsymbol{\Sigma} =$

	x_1	x_2	x_7	x_{13}	x_{11}	x_{12}	x_{10}	x_3	x_4	x_5	x_6	x_8	x_9	x_{14}	x_{15}	c_i
f_1	$1^\bullet$				0		0					0		0		0
f_2		$1^\bullet$		0		0							0		0	0
f_7			$1^\bullet$					0	0	0	0					0
f_{13}		0		$1^\bullet$							0					0
f_{11}	0				$1^\bullet$				0							0
f_{12}		0				$1^\bullet$				0						0
f_{10}	0						$1^\bullet$	0								0
f_3			0				0	$0^\bullet$		0	0					0
f_4			0		0				$0^\bullet$	0	0					0
f_5			0			0		0	0	$0^\bullet$						0
f_6			0	0				0	0		$0^\bullet$					0
f_8	0											$1^\bullet$				0
f_9		0											$1^\bullet$			0
f_{14}	0													$1^\bullet$		0
f_{15}		0													$1^\bullet$	0
d_j	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	

The functions are

$$U_{D1} = \quad x_3 - x_5 - x_7 - U_{in2}(t) \qquad q(U) = \gamma(e^{\delta U} - 1)$$

$$U_{D2} = -x_4 + x_6 - x_7 - U_{in2}(t) \qquad U_{in1}(t) = 0.5\sin(2000\pi t)$$

$$U_{D3} = \quad x_4 + x_5 + x_7 + U_{in2}(t) \qquad U_{in2}(t) = 2\sin(20000\pi t)$$

$$U_{D4} = -x_3 - x_6 + x_7 + U_{in2}(t).$$

The parameters are

$$\begin{array}{lll} C = 1.6 \times 10^{-8} & L_h = 4.45 & R_{g1} = 36.3 \\ C_p = 10^{-8} & L_{s1} = 2 \times 10^{-3} & R_{g2} = 17.3 \\ R = 25 \times 10^3 & L_{s2} = 5 \times 10^{-4} & R_{g3} = 17.3 \\ R_p = 50 & L_{s3} = 5 \times 10^{-4} & R_i = 5 \times 10 \\ R_c = 6 \times 10^2 & \gamma = 40.67286402 \times 10^{-9} & \delta = 17.7493332. \end{array}$$

Each 1×1 block has a nonsingular Jacobian: $\mathbf{J}_{qq} = -1$ for $q = 1\!:\!7, 9\!:\!12$, or equivalently $\partial f_i / \partial x_i' = -1$ for $i = 1, 2, 7\!:\!15$. SA fails with $\det(\mathbf{J}) \equiv 0$ because block 8 has an identically singular sub-Jacobian

$$\mathbf{J}_{88} = \begin{array}{c} \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{array} \begin{array}{c} \begin{array}{cccc} x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} -s_1 - s_4 & & s_1 & -s_4 \\ & -s_2 - s_3 & -s_3 & s_2 \\ s_1 & -s_3 & -s_1 - s_3 & \\ -s_4 & s_2 & & -s_2 - s_4 \end{bmatrix} \end{array}, \quad \text{where} \quad s_i = \gamma \delta e^{\delta U_{Di}}.$$

This is a nonlinear block, since variables x_3, x_4, x_5, x_6 do not occur jointly linearly in equations f_3, f_4, f_5, f_6. One can also see these variables appear in $\mathbf{J}_{88}$.

LC method. We find a constant vector $\widehat{\mathbf{u}} = [1, -1, 1, -1]^T \in \operatorname{coker}(\mathbf{J}_{88})$, which satisfies the block LC condition (6.4). Then $\mathbf{u} = [\mathbf{0}_7^T, 1, -1, 1, -1, \mathbf{0}_4^T]^T$. We use (6.3) to derive

$$I = \{\, i \mid u_i \not\equiv 0 \,\} = \{\, 8, 9, 10, 11 \,\}, \quad \underline{c} = 0, \quad \text{and} \quad L = \overline{L} = \{\, 8, 9, 10, 11 \,\}.$$

The row indices in $\overline{L}$ correspond to the equations f_3, f_4, f_5, f_6. We can pick any one of them and replace it by

$$\overline{f} = u_1 f_3 + u_2 f_4 + u_3 f_5 + u_4 f_6 = f_3 - f_4 + f_5 - f_6 = x_{10} + x_{11} + x_{12} + x_{13}.$$

We choose f_3 and replace it by $\overline{f}_3 = \overline{f}$. The resulting DAE has the following $\overline{\mathbf{\Sigma}}$ with $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 10 < 11 = \mathrm{Val}(\mathbf{\Sigma})$.

$\overline{\mathbf{\Sigma}} =$

	x_1	x_2	x_7	x_3	x_4	x_5	x_6	x_{10}	x_{11}	x_{12}	x_{13}	x_8	x_9	x_{14}	x_{15}	c_i
f_1	$1^\bullet$							0	0			0		0		0
f_2		$1^\bullet$								0	0		0		0	0
f_7			$1^\bullet$	0	0	0	0									0
f_{10}	0			$0^\bullet$				1								0
f_5			0	0	$0^\bullet$	0				0						0
f_4			0		0	$0^\bullet$	0		0							0
f_6			0	0	0		$0^\bullet$				0					0
$\overline{f}_3$								$0^\bullet$	0	0	0					1
f_{11}	0				0				$1^\bullet$							0
f_{12}		0				0				$1^\bullet$						0
f_{13}		0					0				$1^\bullet$					0
f_8	0											$1^\bullet$				0
f_9		0											$1^\bullet$			0
f_{14}	0													$1^\bullet$		0
f_{15}		0													$1^\bullet$	0
d_j	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	

Again, each 1×1 block has a nonsingular Jacobian:

$$\partial f_i / \partial x_i' = -1 \quad \text{for } i = 1, 2, 7, 8, 9, 14, 15.$$

The sub-Jacobian of block 4 in the resulting DAE is

$$\overline{\mathbf{J}}_{44} = \begin{array}{c} \\ f_{10} \\ f_5 \\ f_4 \\ f_6 \\ \overline{f}_3' \\ f_{11} \\ f_{12} \\ f_{13} \end{array} \begin{array}{c} \begin{array}{cccccccc} x_3 & x_4 & x_5 & x_6 & x_{10}' & x_{11}' & x_{12}' & x_{13}' \end{array} \\ \begin{bmatrix} -L_{s2}^{-1} & & & & -1 & & & \\ s_1 & -s_3 & -s_1 - s_3 & & & & & \\ & -s_2 - s_3 & -s_3 & s_2 & & & & \\ -s_4 & s_2 & & -s_2 - s_4 & & & & \\ & & & & 1 & 1 & 1 & 1 \\ & L_{s3}^{-1} & & & & -1 & & \\ & & -L_{s2}^{-1} & & & & -1 & \\ & & & L_{s3}^{-1} & & & & -1 \end{bmatrix} \end{array},$$

whose determinant is

$$\det(\overline{\mathbf{J}}_{44}) = 2s_1s_2s_3s_4(s_1^{-1} + s_2^{-1} + s_3^{-1} + s_4^{-1})(L_{s2}^{-1} + L_{s3}^{-1}).$$

The SA succeeds at any point where $\det(\overline{\mathbf{J}}_{44}) \neq 0$, and the DAE is of index 2.

The initial values given for solving this DAE are

$$x_i = 0 \quad \text{for } i = 1:15, \quad \text{and} \quad x_i' = 0 \quad \text{for } i = 1, 2, 7:15.$$

They satisfy all f_i and f_3', and hence are consistent. At this consistent point, $\det(\overline{\mathbf{J}}_{44}) = 1.2040 \times 10^{-14}$ and $\text{cond}(\overline{\mathbf{J}}_{44}) = 4.9451 \times 10^{12}$. This large condition number is due to equations f_4, f_5, f_6 not being scaled properly. If we multiply each of these equations by 10^7, then the determinant of the resulting sub-Jacobian is 1.2040×10^7 and the condition number becomes 4.9456×10^5.

ES method. Take $\widehat{\mathbf{v}} = [-1, 1, -1, 1]^T \in \ker(\mathbf{J}_{88})$. Then $\mathbf{v} = [\mathbf{0}_7^T, -1, 1, -1, 1, \mathbf{0}_4^T]^T$. We use (6.12) to derive

$$J = \overline{J} = \{\, j \mid v_j \not\equiv 0 \,\} = \{\, 8, 9, 10, 11 \,\}, \quad s = |J| = 4, \quad M = J, \quad \text{and} \quad \overline{c} = 0.$$

We choose column index $l = 8 \in \overline{J}$ in the permuted $\mathbf{\Sigma}$. The variable of this column is x_3. The other variables in block 8 are x_4, x_5, x_6, so we introduce for them, respectively,

$$y_4 = x_4 - \frac{v_9}{v_8} \cdot x_3, \quad y_5 = x_5 - \frac{v_{10}}{v_8} \cdot x_3, \quad \text{and} \quad y_6 = x_6 - \frac{v_{11}}{v_8} \cdot x_3.$$

Then we append the equations corresponding to these variables

$$0 = g_4 = -y_4 + x_4 + x_3, \quad 0 = g_5 = -y_5 + x_5 - x_3, \quad \text{and} \quad 0 = g_6 = -y_6 + x_6 + x_3.$$

The equations in block 8 are f_3, f_4, f_5, f_6. In these equations, we perform the following substitutions.

replace	by	in
x_4	$y_4 - x_3$	f_4, f_5, f_6
x_5	$y_5 + x_3$	f_3, f_4, f_5
x_6	$y_6 - x_3$	f_3, f_4, f_6

The resulting index-2 DAE is of size 18, and we do not show its equations and SA results. It has $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = 10 < 11 = \mathrm{Val}(\mathbf{\Sigma})$ and

$$\det(\overline{\mathbf{J}}) = -2s_1s_2s_3s_4(s_1^{-1} + s_2^{-1} + s_3^{-1} + s_4^{-1})(L_{s2}^{-1} + L_{s3}^{-1}).$$

The largest fine block is of size 12, and the other six fine blocks are of size 1. The SA succeeds at any point where $\det(\overline{\mathbf{J}}) \neq 0$.

7.3 A family of DAEs by Reißig

Reißig *et al.* construct a family of DAEs of differentiation index 1 for which SA finds an arbitrarily high structural index $\nu_S > 1$ [49]. They are linear constant coefficient DAEs, and each has the form

$$\mathbf{A} \cdot \mathbf{x}'(t) + \mathbf{B} \cdot \mathbf{x}(t) - \mathbf{q}(t) = 0. \tag{7.3}$$

The problem is of size $n = 2k + 1$ where $k \in \mathbb{N}^+$. The $n \times n$ mass matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & & & & & & \\ & 1 & 1 & 0 & & & & & \\ & & 0 & 1 & 1 & & & & \\ & & & 1 & 1 & 0 & & & \\ & & & & 0 & \ddots & \ddots & & \\ & & & & & \ddots & \ddots & 0 & \\ & & & & & & 0 & 1 & 1 \\ & & & & & & & 1 & 1 \\ & & & & & & & & 0 \end{bmatrix},$$

and $\mathbf{B}$ is the $n \times n$ identity matrix. In other words, for $i = 1 : k$, equations f_{2i-1} and f_{2i} have the common expression $x'_{2i} + x'_{2i+1}$.

In [42], Pryce applies the Σ-method on (7.3) with $n = 5$ and $k = 2$:

$$\begin{aligned}
0 = f_1 &= x_2' + x_3' + x_1 - q_1(t) \\
0 = f_2 &= x_2' + x_3' + x_2 - q_2(t) \\
0 = f_3 &= x_4' + x_5' + x_3 - q_3(t) \\
0 = f_4 &= x_4' + x_5' + x_4 - q_4(t) \\
0 = f_5 &= \qquad\quad x_5 - q_5(t).
\end{aligned} \tag{7.4}$$

$$\boldsymbol{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left[\begin{array}{ccccc} 0^{\bullet} & 1 & 1 & & \\ & 1^{\bullet} & 1 & & \\ & & 0^{\bullet} & 1 & 1 \\ & & & 1^{\bullet} & 1 \\ & & & & 0^{\bullet} \end{array}\right] \\ \begin{array}{ccccc} 0 & 1 & 1 & 2 & 2 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ \mathrm{Val}(\boldsymbol{\Sigma}) = 2 \end{array}$$

$$\mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3' \\ f_4' \\ f_5'' \\ \det(\mathbf{J}) = 1 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2' & x_3' & x_4'' & x_5'' \end{array} \\ \left[\begin{array}{ccccc} 1 & 1 & 1 & & \\ & 1 & 1 & & \\ & & 1 & 1 & 1 \\ & & & 1 & 1 \\ & & & & 1 \end{array}\right] \end{array}$$

The method succeeds, but reports $\nu_S = 3$ greater than $\nu_d = 1$. Equivalent to the Σ-method, Pantelides's algorithm reports the same structural index as well. This index overestimation is not favoured, as SA exaggerates the numerical difficulty of solving (7.3). We illustrate below how to fix the index overestimation problem on (7.4).

Although matrix $\mathbf{A}$ is structurally singular—that is, every transversal contains an identically zero entry—we can permute $\mathbf{A}$ into the following structure

$$\begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_2 & x_3 & x_4 & x_5 & x_1 \end{array} \\ \left[\begin{array}{ccc|cc} 1 & 1 & & & \\ 1 & 1 & & & \\ \hline & & 1 & 1 & \\ & & 1 & 1 & \\ \hline & & & & \end{array}\right] \end{array}.$$

Note that this is *not* a BTF of $\mathbf{A}$, as the last diagonal block is empty. However, by observing this structure, we can replace f_1 by $\overline{f}_1 = f_1 - f_2$, and replace f_3 by $\overline{f}_3 = f_3 - f_4$. The converted DAE has new equations

$$0 = \overline{f}_1 = x_1 - x_2 - q_1(t) + q_2(t)$$

$$0 = \overline{f}_3 = x_3 - x_4 - q_3(t) + q_4(t)$$

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ \overline{f}_3 \\ f_4 \\ f_5 \\ d_j \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left[\begin{array}{ccccc} 0^{\bullet} & 0 & & & \\ & 1^{\bullet} & 1 & & \\ & & 0^{\bullet} & 0 & \\ & & & 1^{\bullet} & 1 \\ & & & & 0^{\bullet} \end{array}\right] \\ \begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \end{array} \end{array} \begin{array}{c} c_i \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = 2 \end{array} \qquad \overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}'_1 \\ f_2 \\ \overline{f}'_3 \\ f_4 \\ f'_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x'_1 & x'_2 & x'_3 & x'_4 & x'_5 \end{array} \\ \left[\begin{array}{ccccc} 1 & -1 & & & \\ & 1 & 1 & & \\ & & 1 & -1 & \\ & & & 1 & 1 \\ & & & & 1 \end{array}\right] \end{array} \quad \det(\overline{\mathbf{J}}) = 1$$

Since no d_j equals 0, SA reports index $\nu_S = \max_i c_i = 1$. Notice that here we *do*

not choose the canonical offset pair

$$(\mathbf{c};\mathbf{d}) = (0,0,1,0,1;0,1,1,1,1),$$

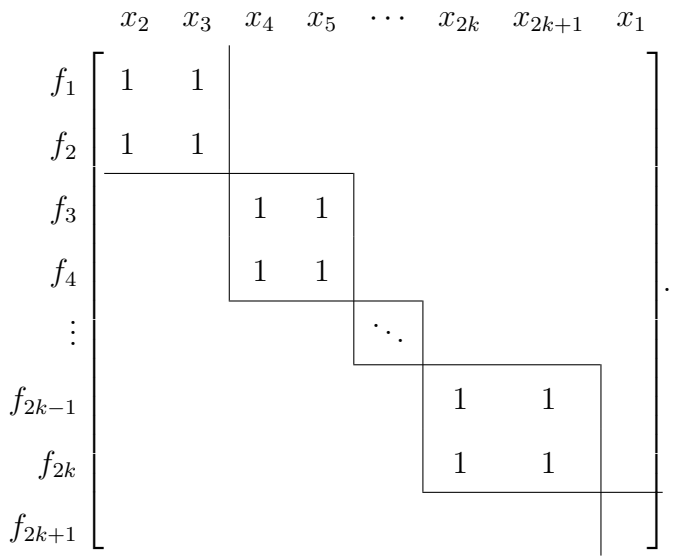
which still give an overestimated structural index $\nu_S = 2$ as $d_1 = 0$ and $c_3 = c_5 = 1$.

Consider for general case $k \geq 1$. The DAE is

$$\begin{aligned}
0 = f_{2i-1} &= x'_{2i} + x'_{2i+1} + x_{2i-1} - q_{2i-1}(t) && i = 1:k \\
0 = \quad f_{2i} &= x'_{2i} + x'_{2i+1} + \quad x_{2i} - q_{2i}(t) && i = 1:k \\
0 = f_{2k+1} &= \qquad\qquad x_{2k+1} - q_{2k+1}(t).
\end{aligned}$$

SA succeeds but reports structural index $\nu_S = k + 1$.

To remedy this index overestimation, we repeat the same strategy used above. We permute $\mathbf{A}$ into the structure



$$\mathbf{\Sigma} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ \vdots \\ f_{2k-1} \\ f_{2k} \\ f_{2k+1} \\ d_j \end{array} \begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{2k} & x_{2k+1} & c_i \end{array} \\ \left[\begin{array}{ccccccccc} 0^{\bullet} & 1 & 1 & & & & & & \\ & 1^{\bullet} & 1 & & & & & & \\ & & 0^{\bullet} & 1 & 1 & & & & \\ & & & 1^{\bullet} & 1 & & & & \\ & & & & 0^{\bullet} & \ddots & & & \\ & & & & & \ddots & 1 & 1 & \\ & & & & & & 1^{\bullet} & 1 & \\ & & & & & & & 0^{\bullet} & \end{array}\right] \\ \begin{array}{ccccccccc} 0 & 1 & 1 & 2 & 2 & \cdots & k-1 & k & \end{array} \end{array} \begin{array}{l} \\ 0 \\ 0 \\ 1 \\ 1 \\ \vdots \\ k-1 \\ k-1 \\ k \\ \mathrm{Val}(\mathbf{\Sigma}) = k \end{array}$$

$$\mathbf{J} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ \vdots \\ f_{2k-1} \\ f_{2k} \\ f_{2k+1} \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{2k} & x_{2k+1} \end{array} \\ \left[\begin{array}{cccccccc} 1 & 1 & 1 & & & & & \\ & 1 & 1 & & & & & \\ & & 1 & 1 & 1 & & & \\ & & & 1 & 1 & & & \\ & & & & \ddots & \ddots & & \\ & & & & & 1 & 1 & 1 \\ & & & & & & 1 & 1 \\ & & & & & & & 1 \end{array}\right] \end{array}$$

$$\det(\mathbf{J}) = 1$$

Then, for $i = 1:k$, we replace f_{2i-1} by $\overline{f}_{2i-1} = f_{2i-1} - f_{2i}$. The converted DAE is

$$\begin{aligned} 0 =& \overline{f}_{2i-1} = x_{2i-1} - x_{2i} - q_{2i-1}(t) + q_{2i}(t) && i = 1:k \\ 0 =& \quad f_{2i} = x'_{2i} + x'_{2i+1} + x_{2i} - q_{2i}(t) && i = 1:k \\ 0 =& f_{2k+1} = x_{2k+1} - q_{2k+1}(t). \end{aligned} \tag{7.5}$$

Now SA reports the correct $\nu_S = 1$ on the converted DAE (7.5). Again, we use a non-canonical offset pair of $\overline{\mathbf{\Sigma}}$, while in the canonical case we would have $d_1 = c_1 = 0$ and the structural index 2.

$$\overline{\mathbf{\Sigma}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ \overline{f}_3 \\ f_4 \\ \vdots \\ \overline{f}_{2k-1} \\ f_{2k} \\ f_{2k+1} \\ d_j \end{array} \begin{array}{c} \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{2k} & x_{2k+1} & c_i \end{array} \\ \left[\begin{array}{cccccccc} 0^\bullet & 0 & & & & & & \\ & 1^\bullet & 1 & & & & & \\ & & 0^\bullet & 0 & & & & \\ & & & 1^\bullet & 1 & & & \\ & & & & 0^\bullet & \ddots & & \\ & & & & & \ddots & 0 & \\ & & & & & & 1^\bullet & 1 \\ & & & & & & & 0^\bullet \end{array}\right] \\ \begin{array}{cccccccccc} 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{array} \end{array} \begin{array}{c} \\ 1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 1 \\ \mathrm{Val}(\overline{\mathbf{\Sigma}}) = k \end{array}$$

$$\overline{\mathbf{J}} = \begin{array}{c} \\ \overline{f}_1 \\ f_2 \\ \overline{f}_3 \\ f_4 \\ \vdots \\ \overline{f}_{2k-1} \\ f_{2k} \\ f_{2k+1} \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & x_{2k} & x_{2k+1} \end{array} \\ \begin{bmatrix} 1 & -1 & & & & & & \\ & 1 & 1 & & & & & \\ & & 1 & -1 & & & & \\ & & & 1 & 1 & & & \\ & & & & \ddots & \ddots & & \\ & & & & & 1 & -1 & \\ & & & & & & 1 & 1 \\ & & & & & & & 1 \end{bmatrix} \end{array}$$

$$\det(\overline{\mathbf{J}}) = 1$$

7.4 Summary of examples

We summarize in Table 7.1 the comparison between the basic conversion methods and the block methods on five SA-unfriendly DAEs in this thesis: the introductory example (6.2), the Campbell-Griepentrog robot arm DAE (6.11), the transistor amplifier DAE (7.1), the ring modulator (7.2), and the Scholz-Steinbrecher linear constant coefficient DAE (3.10).

We give several remarks on Table 7.1.

- The basic LC method and the block LC method give the same conversion(s) on (3.10), (7.1), and (7.2). The basic ES method and the block ES method give the same conversion on (6.2) and (7.1).

- For (6.11) and (7.2), both the basic ES method and the block ES method find $\overline{J} \neq \emptyset$, so they give desirable conversions. However, the basic ES method introduces more variables and appends more equations for fixing these DAEs.

- For (6.2), the basic LC method is not applicable, while the block LC method finds a desirable conversion. For (6.11), the basic LC method finds a conversion that is not desirable, while the block LC method finds a desirable conversion.

In Table 7.2, we summarize the comparisons between the SA-unfriendly DAEs presented in this thesis and their SA-friendly formulations, in terms of differentiation index ν_d, structural index ν_S, and value of signature matrix.

DAE	Method	Result
Introductory example (6.2)	basic LC	$L = \emptyset$, method not applicable
	block LC	$\overline{L} \neq \emptyset$, desirable conversion
	basic ES	Same conversion with $\overline{J} \neq \emptyset$
	block ES	
Campbell-Griepentrog robot arm (6.11)	basic LC	$L \neq \emptyset, \overline{L} = \emptyset$; available conversion not desirable
	block LC	$\overline{L} \neq \emptyset$; available conversion is desirable
	basic ES	$\overline{J} \neq \emptyset$, size increased by 2
	block ES	$\overline{J} \neq \emptyset$, size increased by 1
Ring modulator (7.2)	basic LC	Same conversion with $\overline{L} \neq \emptyset$
	block LC	
	basic ES	$\overline{J} \neq \emptyset$, size increased by 7
	block ES	$\overline{J} \neq \emptyset$, size increased by 3
Transistor amplifier (7.1)	basic LC	Same conversion with $\overline{L} \neq \emptyset$
	block LC	
	basic ES	Same conversion with $\overline{J} \neq \emptyset$
	block ES	
Scholz-Steinbrecher linear constant coefficient DAE (3.10)	basic LC	Same conversions with $\overline{L} \neq \emptyset$
	block LC	
	basic ES	Both become inapplicable in the second iteration
	block ES	

Table 7.1: Comparison between the basic conversion methods and the block methods on several DAEs presented in this thesis.

DAE	SA-friendly / SA-unfriendly formulation		
	ν_d	ν_S	Val($\overline{\mathbf{\Sigma}}$) / Val($\mathbf{\Sigma}$)
Example (4.6) for LC method	2 / 3	2 / 2	1 / 2
Modified pendulum A (4.13)	3 / 6	3 / 3	2 / 9
Example (4.14) for ES method	2 / 2	2 / 1	1 / 2
Scholz-Steinbrecher DAE (5.1)	2 / 3	2 / 1	0 / 2
Modified pendulum B (5.4)	3 / 3	3 / 2	2 / 4
Example (6.2) for block methods	1 / 2	1 / 2	1 / 2
Robot arm (6.11)	5 / 5	5 / 3	0 / 2
Transistor amplifier (7.1)	1 / 1	1 / 0	5 / 8
Ring modulator (7.2)	1 / 1	1 / 1	10 / 11
$*$DAE (4.23)	1 / 1	1 / 0	1 / 2
$*$Family of DAEs by Reißig	1 / 1	1 / $\frac{n+1}{2}$	$\frac{n+1}{2}$ / $\frac{n+1}{2}$

Table 7.2: Some characteristics (differentiation index, structural index, and value of signature matrix) of the SA-unfriendly DAEs analyzed in this thesis, and the structural data of their corresponding SA-friendly formulations obtained by conversion methods. Two items with "$*$" are special cases: for (4.23) in Example 4.24, neither of our conversion methods is applicable; for the differentiation index-1 DAEs by Reißig, SA does not fail but gives a structural index linear in the problem size; see §7.3 and the last paragraph in §1.3.

Chapter 8

Conclusions

We identified in Chapter 3 two types of SA's failure on a structurally well-posed DAE. In the first type, the System Jacobian is structurally (and hence identically) singular. The failure may attribute to hidden symbolic cancellations, which lead to overestimations of signature entries and thence more identically zero entries in the System Jacobian. Therefore, to handle this type of failure, we do symbolic simplifications on some equations before performing SA.

We focused on dealing with the second type of failure, where a remedy for a failure is less obvious. When SA fails on an SA-unfriendly DAE, the Jacobian is identically but not structurally singular. In Chapter 4, we proposed two conversion methods, the LC method and the ES method, which are the main contribution of this thesis. They reformulate an SA-unfriendly DAE, with finite Val($\mathbf{\Sigma}$) and an identically singular System Jacobian, into an equivalent DAE that is more likely to be SA-friendly with a nonsingular System Jacobian. Our conversion methods enable SA to recognize better the true structure of a DAE, and thus SA is more likely to succeed and report correct structural information. Moreover, our methods provide insights into reasons for SA's

failures, which were not well understood before.

We summarize the ideas of these methods here. The LC method replaces an existing equation by a linear combination of some existing equations and derivatives of them. The ES method appends new equations that define some newly introduced variables, and replaces in the original equations some existing derivatives by a linear combination of new variables and other existing derivatives. The conditions for applying these methods can be checked automatically, and the main result of a conversion is $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$, where $\overline{\mathbf{\Sigma}}$ is the signature matrix of the resulting DAE.

A conversion by either method guarantees that the original DAE and the converted one are equivalent, that is, they have the same solution (if any) over some time interval. Considering the equivalence and simplicity of performing a conversion, we presented in Table 4.1 our rationale for choosing the desirable conversion method between the two.

In Chapter 6, we combined the block triangularization with the (simple) conversion methods, and developed our block conversion methods. When $\mathbf{J}$ is identically singular and the DAE has a nontrivial BTF, we can identify which diagonal blocks of the Jacobian are identically singular, and then perform a conversion on each such block. We base this strategy on the view that each diagonal block can be regarded as a sub-DAE.

The computational cost of performing a conversion depends on the size of the DAE, its sparsity, and intricacy of the equations involved in a conversion. In general, this cost cannot be determined in advance, since deciding whether an expression is identical to zero is unsolvable [51]. For example, given a solvable SA-friendly DAE of equations $\mathbf{f} = 0$, one can make the task of fixing $\mathbf{Mf} = 0$ arbitrarily difficult by

constructing any generically nonsingular dense $n \times n$ matrix $\mathbf{M}$ that contains any expressions comprising derivatives of the x_j's, typically lower than the d_jth.

Fortunately, the Σ-method already provably succeeds on many DAEs, and fixing those failure cases so far encountered in practice does not require much computational effort. Also, since the block conversion methods work on a singular block only, which can have a significantly smaller size compared to the whole DAE, they can reduce the computational cost and improve the efficiency of finding a useful conversion for fixing SA's failure.

We combined MATLAB's Symbolic Math Toolbox [56] with our structural analysis software DAESA [37, 46], and have built a prototype code that automates the conversion process. We aim to incorporate them in a future version of DAESA.

With our prototype code, we have applied our methods on numerous DAEs. They are either arbitrarily constructed to be SA-failure cases for our investigations, or borrowed from the existing literature. We have shown how to fix Scholz and Steinbrecher's linear constant coefficient DAE in §5.1, the Campbell-Griepentrog robot arm DAE [5] in Examples 6.3 and 6.5, the transistor amplifier DAE in §7.1, and the ring modulator DAE in §7.2. Our conversion methods succeed in fixing all these solvable but SA-unfriendly DAEs. We believe that our assumptions and conditions are reasonable for practical problems, and that these methods can help to make the Σ-method more reliable.

We briefly discussed another limitation of SA, the index overestimation problem. On some DAEs, typically Reißig's family of DAEs of differentiation index 1, SA produces a nonsingular Jacobian (hence succeeds) but an unnecessarily high structural index, while the differentiation index is low. This scenario is not favoured, as the

numerical difficulty of solving these DAEs is exaggerated. In §7.3, we resolved the index overestimation problem on Reißig's DAEs, but did not develop a theory such that the fix can be generalized in a systematic way. We conjecture that this problem is due to the structure of the mass matrix $\partial\mathbf{f}/\partial\mathbf{y}'$ and thus may be resolved by simply taking a linear combination of the differential equations so that the mass matrix of the remaining differential equations has full row rank.

We have shown in the example in §6.1 that the basic LC method does not apply successfully, but the corresponding block method leads to a successful conversion. However, we have not found an example where a basic method works but the corresponding block method does not. Investigating further the connections between the basic and block methods is left for future research.

Another research direction is combining the dummy derivative index reduction method [26] with our conversion methods. When the conditions for applying them are violated, appending some differentiated equations and replacing some "genuine" derivatives by dummy derivatives (which are algebraic) may make a conversion possible.

We end this thesis with our main conjecture related to SA's failure. In all our experiments, when we transform an SA-unfriendly DAE to an equivalent SA-friendly DAE, the value of a signature matrix always decreases. As Pryce points out in [41], the solvability of a DAE lies within its inherent nature, not the way it is formulated nor the method that analyzes it. Hence, we conjecture that an SA-friendly DAE should always have a reasonable but never overestimated $\mathrm{Val}(\mathbf{\Sigma})$ that can be interpreted as the DOF of this DAE; see (2.9). In other words, a DAE should not be formulated to exhibit more degrees of freedom than the underlying problem has. However, based on

our current knowledge, it appears difficult to show why overestimating the number of degrees of freedom leads to an identically singular System Jacobian.

Appendix A

Proofs for expression substitution methods

A.1 Preliminary results and proof of Lemma 4.19

Let the notation be as at the start of §4.2. We give two preliminary lemmas prior to the main proof of Lemma 4.19.

Lemma A.1 *Let* $r \in J \setminus \{ l \}$ *and*

$$\omega_1 = y_r + \frac{v_r}{v_l} \cdot x_l^{(d_l - \overline{c})}.$$

Then

$$\sigma(x_j, \omega_1) = \begin{cases} < d_j - \overline{c} & \text{if } j \in J \setminus \{ l \} \\ \leq d_j - \overline{c} & \text{otherwise.} \end{cases} \tag{A.1}$$

Proof. Consider the case $j = l \in J$. Obviously $\sigma(x_l, \omega_1) = d_l - \overline{c}$.

Now consider the case $j \neq l$. Since x_j can occur only in v_r and v_l in ω_1, we have $\sigma(x_j, \omega_1) \leq \sigma(x_j, \mathbf{v})$. It follows from (4.19) and the case $j = l$ that (A.1) holds. □

Lemma A.2 *Let $r \in J \setminus \{ l \}$, $i \in M$, and*

$$\omega_2 = \omega_1^{(\overline{c}-c_i)} = \left(y_r + \frac{v_r}{v_l} \cdot x_l^{(d_l - \overline{c})} \right)^{(\overline{c}-c_i)}. \tag{A.2}$$

Then

$$\sigma(x_j, \omega_2) = \begin{cases} < d_j - c_i & \text{if } j \in J \setminus \{ l \} \\ \leq d_j - c_i & \text{otherwise.} \end{cases} \tag{A.3}$$

Proof. Since $\overline{c} = \max_{i \in M} c_i$, $\overline{c} - c_i \geq 0$ holds for all $i \in M$. From (A.2), connecting $\sigma(x_j, \omega_2) = \sigma(x_j, \omega_1) + (\overline{c} - c_i)$ to (A.1) immediately yields (A.3). □

Using the two assumptions before Lemma 4.19, we prove it below.

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

$$\overline{\mathbf{\Sigma}} = \left[\begin{array}{c|c|c} \overline{\mathbf{\Sigma}}_{11} & \overline{\mathbf{\Sigma}}_{12} & \overline{\mathbf{\Sigma}}_{13} \\ \hline \overline{\mathbf{\Sigma}}_{21} & \overline{\mathbf{\Sigma}}_{22} & \overline{\mathbf{\Sigma}}_{23} \end{array} \right]. \tag{A.4}$$

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

$$
\text{(a)}\quad \overline{\mathbf{\Sigma}}_{11} = \begin{array}{c} \\ \overline{f}_1 \\ \vdots \\ \overline{f}_n \\ \widetilde{d}_j \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & \cdots & x_{l-1} & x_l & x_{l+1} & \cdots & x_s & \widetilde{c}_i \end{array} \\ \left[\begin{array}{cccccccc} & & & & & & & \\ & & & & < & & & \\ & & & & & & & \end{array}\right] \\ \begin{array}{cccccccc} d_1 & \cdots & d_{l-1} & d_l & d_{l+1} & \cdots & d_s & \end{array} \end{array} \begin{array}{c} \\ c_1 \\ \vdots \\ c_n \\ \\ \end{array}
$$

Consider $j, r \in J \setminus \{ l \}$. By (4.17), we substitute ω_2 in (A.2) for every $x_r^{(d_r - c_i)}$ in f_i for all $i = 1\!:\!n$. Using (A.3) gives $\sigma(x_j, \omega_2) < d_j - c_i$ for all $i \in M$. So these expression substitutions do not introduce $x_r^{(d_r - c_i)}$, $r \in J \setminus \{ l \}$, in $\overline{f}_i$. Because of M in (4.15), we have $d_j - c_i > \sigma_{ij}$ for all $i \notin M$ and $j \in J$. Hence

$$
\sigma\left(x_j, \overline{f}_i\right) < d_j - c_i \qquad \text{for } j \in J \setminus \{ l \},\ i = 1\!:\!n. \tag{A.5}
$$

What remains to show is the case $j = l$. From (4.16), $x_r^{(d_r - \overline{c})} = y_r + \frac{v_r}{v_l} \cdot x_l^{(d_l - \overline{c})}$. Taking the partial derivatives of both sides with respect to $x_l^{(d_l - \overline{c})}$ and applying Griewank's Lemma (2.7) with $w = x_r^{(d_r - \overline{c})}$ and $q = \overline{c} - c_i \geq 0$ for all $i \in M$ gives

$$
\frac{v_r}{v_l} = \frac{\partial x_r^{(d_r - \overline{c})}}{\partial x_l^{(d_l - \overline{c})}} = \frac{\partial x_r^{(d_r - \overline{c} + \overline{c} - c_i)}}{\partial x_l^{(d_l - \overline{c} + \overline{c} - c_i)}} = \frac{\partial x_r^{(d_r - c_i)}}{\partial x_l^{(d_l - c_i)}}, \quad \text{and then} \tag{A.6}
$$

$$
\begin{aligned}
\frac{\partial \overline{f}_i}{\partial x_l^{(d_l - c_i)}} &= \frac{\partial f_i}{\partial x_l^{(d_l - c_i)}} + \sum_{r \in J \setminus \{l\}} \frac{\partial f_i}{\partial x_r^{(d_r - c_i)}} \cdot \frac{\partial x_r^{(d_r - c_i)}}{\partial x_l^{(d_l - c_i)}} && \text{by the chain rule} \\
&= J_{il} + \sum_{r \in J \setminus \{l\}} J_{ir} \cdot \frac{v_r}{v_l} && \text{by (A.6)} \\
&= \frac{1}{v_l} \sum_{r \in J} J_{ir} v_r = \frac{1}{v_l} (\mathbf{J}\mathbf{v})_i = 0 && \text{by } \mathbf{J}\mathbf{v} = \mathbf{0}.
\end{aligned}
$$

This gives $\sigma\left(x_l, \overline{f}_i\right) < d_l - c_i$ for all $i = 1\!:\!n$. Together with (A.5) we have proved the "$<$" part in $\overline{\mathbf{\Sigma}}_{11}$.

$$\text{(b)}\quad \overline{\mathbf{\Sigma}}_{12} = \begin{array}{c} \\ \overline{f}_1 \\ \vdots \\ \overline{f}_n \\ \widetilde{d}_j \end{array} \begin{array}{c} \begin{array}{cccc} x_{s+1} & \cdots & x_n & \widetilde{c}_i \end{array} \\ \left[\begin{array}{ccc} & & \\ & \leq & \\ & & \end{array}\right] \\ \begin{array}{ccc} d_{s+1} & \cdots & d_n \end{array} \end{array} \begin{array}{c} \\ c_1 \\ \vdots \\ c_n \\ \end{array}$$

The substitutions do not affect x_j, for all $j \notin L$. By (A.3), such an x_j occurs in every ω_2 of order $\leq d_j - c_i$, $i \in M$. Hence also

$$\sigma\left(x_j, \overline{f}_i\right) \leq d_j - c_i \quad \text{for all } i = 1\!:\!n \text{ and } j \notin L.$$

$$\text{(c)}\quad \overline{\mathbf{\Sigma}}_{13} = \begin{array}{c} \\ \overline{f}_1 \\ \vdots \\ \overline{f}_n \\ \widetilde{d}_j \end{array} \begin{array}{c} \begin{array}{ccccccc} y_1 & \cdots & y_{l-1} & y_l & y_{l+1} & \cdots & y_s & \widetilde{c}_i \end{array} \\ \left[\begin{array}{ccccccc} & & & -\infty & & & & \\ & \leq & & \vdots & & \leq & & \\ & & & -\infty & & & & \end{array}\right] \\ \begin{array}{ccccccc} \overline{c} & \cdots & \overline{c} & \overline{c} & \overline{c} & \cdots & \overline{c} \end{array} \end{array} \begin{array}{c} \\ c_1 \\ \vdots \\ c_n \\ \end{array}$$

Consider $r \in J \setminus \{ l \}$. For an $i \in M$, y_r occurs of order $\overline{c} - c_i$ in ω_2 in (A.2). For all $i = 1\!:\!n$, if a substitution occurs for an $x_r^{(d_r - c_i)}$ in f_i, then $\sigma\left(y_r, \overline{f}_i\right) = \overline{c} - c_i$; otherwise $\sigma\left(y_r, \overline{f}_i\right) = -\infty$. In either case $\sigma\left(y_r, \overline{f}_i\right) \leq \overline{c} - c_i$.

$$\text{(d)}\quad \overline{\mathbf{\Sigma}}_{21} = \begin{array}{c} \\ g_1 \\ \vdots \\ g_l \\ \vdots \\ g_s \\ \widetilde{d}_j \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & \cdots & x_{l-1} & x_l & x_{l+1} & \cdots & x_s & \widetilde{c}_i \end{array} \\ \begin{bmatrix} = & & & = & & & & \\ & \ddots^{<} & & \vdots & & < & & \\ & & & = & & & & \\ & < & & \vdots & < & \ddots & & \\ & & & = & & & = & \end{bmatrix} \\ \begin{array}{cccccccc} d_1 & \cdots & d_{l-1} & d_l & d_{l+1} & \cdots & d_s & \end{array} \end{array} \begin{array}{c} \\ \overline{c} \\ \vdots \\ \overline{c} \\ \vdots \\ \overline{c} \\ \end{array}$$

Equalities hold on the diagonal and in the lth column, as $y_r^{(d_r-\overline{c})}$ and $y_l^{(d_l-\overline{c})}$ 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 (4.16) and (4.19),

$$\sigma\left(x_j, g_r\right) = \sigma\left(x_j, y_r - x_r^{(d_r-\overline{c})} + \frac{v_r}{v_l} \cdot x_l^{(d_l-\overline{c})}\right) \leq \sigma\left(x_j, \mathbf{v}\right) < d_j - \overline{c}. \tag{A.7}$$

$$\text{(e)}\quad \overline{\mathbf{\Sigma}}_{22} = \begin{array}{c} \\ g_1 \\ \vdots \\ g_l \\ \vdots \\ g_s \\ \widetilde{d}_j \end{array} \begin{array}{c} \begin{array}{ccccc} x_{s+1} & \cdots & x_n & \widetilde{c}_i \end{array} \\ \begin{bmatrix} & \leq & & \\ & & & \\ -\infty & \cdots & -\infty & \\ & & & \\ & \leq & & \end{bmatrix} \\ \begin{array}{ccccc} d_{s+1} & \cdots & d_n & \end{array} \end{array} \begin{array}{c} \\ \overline{c} \\ \vdots \\ \overline{c} \\ \vdots \\ \overline{c} \\ \end{array}$$

Assume again that j, r, l are distinct, where $r \in J$ and $j = s+1 : n$ Then replacing the "$<$" in (A.7) by "$\leq$" proves the "$\leq$" part in $\overline{\mathbf{\Sigma}}_{22}$.

$$
\text{(f)}\quad \overline{\mathbf{\Sigma}}_{23} = \begin{array}{cc} & \begin{matrix} y_1 & \cdots & y_{l-1} & y_l & y_{l+1} & \cdots & y_s \end{matrix} \\ \begin{matrix} \overline{f}_{n+1} \\ \vdots \\ \overline{f}_{n+l} \\ \vdots \\ \overline{f}_{n+s} \end{matrix} & \begin{matrix} g_1 \\ \vdots \\ g_l \\ \vdots \\ g_s \end{matrix} \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & -\infty & & \\ & & & 0 & & & \\ & -\infty & & & \ddots & & \\ & & & & & & 0 \end{bmatrix} \begin{matrix} \overline{c} \\ \vdots \\ \overline{c} \\ \vdots \\ \overline{c} \end{matrix} \\ & \begin{matrix} \widetilde{d}_j \end{matrix} \quad \begin{matrix} \overline{c} & \cdots & \overline{c} & \overline{c} & \overline{c} & \cdots & \overline{c} \end{matrix} \end{array}
$$

Consider $r, j \in J$. By $0 = g_l = -y_l + x_l^{(d_l - \overline{c})}$ and (4.16), y_j occurs in g_r only if $j = r$, and $\sigma(y_j, g_j) = 0$. 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 $\sigma_{n+l,l} = d_l - \overline{c}$.

Recall (4.20) for the formulas of $\widetilde{c}_i$ and $\widetilde{d}_j$ of $\overline{\mathbf{\Sigma}}$. The above verifies the relations between $\overline{\sigma}_{ij}$ and $\widetilde{d}_j - \widetilde{c}_i$, for all $i, j = 1 : n + s$, in $\overline{\mathbf{\Sigma}}$ in Figure 4.1. □

A.2 Proof of Lemma 6.6

For $\overline{\mathbf{\Sigma}} = (\overline{\sigma}_{ij})$ in the block structure in Figure 6.1, we write the block sizes in the array

$$\overline{N} = (N_1, N_2, \ldots, N_{q-1}, N_q + s, N_{q+1}, \ldots, N_p), \tag{A.8}$$

and also write the block sizes of $\mathbf{\Sigma}$ in the array

$$N = (N_1, N_2, \ldots, N_{q-1}, N_q, N_{q+1}, \ldots, N_p). \tag{A.9}$$

Let $\overline{\text{blockOf}}(i)$ denote the block number of a row or column index i in $\overline{\mathbf{\Sigma}}$. From (A.8) and (A.9), it is not difficult to show that

$$\overline{\text{blockOf}}(j) < q \;\Leftrightarrow\; 1 \le j \le \sum_{w=1}^{q-1} N_w \;\Leftrightarrow\; \text{blockOf}(j) < q \quad \text{and} \tag{A.10}$$

$$\begin{aligned} \overline{\text{blockOf}}(j+s) > q \;&\Leftrightarrow\; \sum_{w=1}^{q} N_w + s + 1 \le j + s \\ &\Leftrightarrow\; \sum_{w=1}^{q} N_w + 1 \le j \le n \;\Leftrightarrow\; \text{blockOf}(j) > q. \end{aligned} \tag{A.11}$$

Recall the construction of the two vectors $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$ in (6.17):

$$\widetilde{c}_i = \begin{cases} c_i & \text{if } i = 1\!:\!Q \\ \overline{c} & \text{if } i = Q+1\!:\!Q+s \\ c_{i-s} & \text{if } i = Q+s+1\!:\!n+s, \end{cases} \qquad \widetilde{d}_j = \begin{cases} d_j & \text{if } j = 1\!:\!Q \\ \overline{c} & \text{if } j = Q+1\!:\!Q+s \\ d_{j-s} & \text{if } j = Q+s+1\!:\!n+s, \end{cases} \tag{A.12}$$

where $Q = \sum_{w=1}^{q} N_w$. From this construction, each variable x_j for $j = 1\!:\!n$ has the same "variable offset" in $\overline{\mathbf{\Sigma}}$ as x_j has in $\mathbf{\Sigma}$. Also, each equation $\overline{f}_i$ for $i = 1\!:\!n$ has the same "equation offset" in $\overline{\mathbf{\Sigma}}$ as f_i has in $\mathbf{\Sigma}$. Quotation marks are used here because $(\widetilde{\mathbf{c}}; \widetilde{\mathbf{d}})$ is *not* a valid offset pair of $\overline{\mathbf{\Sigma}}$; this vector pair is merely used for proving $\text{Val}(\overline{\mathbf{\Sigma}}) < \text{Val}(\mathbf{\Sigma})$ in Theorem 6.4.

We aim to show that

$$\widetilde{d}_j - \widetilde{c}_i \begin{cases} > \overline{\sigma}_{ij} & \text{if } \overline{\text{blockOf}}(j) < \overline{\text{blockOf}}(i) \\ \ge \overline{\sigma}_{ij} & \text{if } \overline{\text{blockOf}}(j) \ge \overline{\text{blockOf}}(i). \end{cases} \tag{A.13}$$

For the block structure of $\overline{\mathbf{\Sigma}}$ in Figure 6.1, we have shown on page 112 that

$$\overline{\mathbf{\Sigma}}_{w_1 w_2} = \begin{cases} \mathbf{\Sigma}_{w_1 w_2} & \text{if } w_1 \neq q \text{ and } w_2 \neq q \\ \begin{bmatrix} \mathbf{\Sigma}_{w_1 q} & -\boldsymbol{\infty}_{N_{w_1} \times s} \end{bmatrix} & \text{if } w_1 \neq q \text{ and } w_2 = q. \end{cases} \tag{A.14}$$

Hence, provided $w_1 \neq q$, $\overline{\mathbf{\Sigma}}_{w_1 w_2}$ is below [resp. above] the block diagonal of $\overline{\mathbf{\Sigma}}$, if $\mathbf{\Sigma}_{w_1 w_2}$ is below [resp. above] the block diagonal of $\mathbf{\Sigma}$. By (6.1), the inequalities in (A.13) hold for i with $\overline{\text{blockOf}}(i) \neq q$.

What remains to show is the inequalities in (A.13) for i with $\overline{\text{blockOf}}(i) = q$. These inequalities are for the signature entries in $\overline{\mathbf{\Sigma}}_{q w_2}$, the blocks that are affected by the expression substitutions. We consider three cases for $\overline{\mathbf{\Sigma}}_{q w_2}$: it is (a) below the block diagonal, with $w_2 < q$, (b) above the block diagonal, with $w_2 > q$, or (c) the diagonal block $\overline{\mathbf{\Sigma}}_{qq}$, with $w_2 = q$.

(a) $\overline{\mathbf{\Sigma}}_{q w_2}$ with $w_2 < q$. An entry (i, j) in this block satisfies $\overline{\text{blockOf}}(j) < \overline{\text{blockOf}}(i) = q$. By (A.10), $\text{blockOf}(j) < q$ and hence $j \notin B_q$.

Recall from (6.15) that, in each f_i with $i \in M \subseteq B_q$, we

$$\begin{aligned} &\text{replace each} \quad x_r^{(\sigma_{ir})} \quad \text{with} \quad d_r - c_i = \sigma_{ir} \quad \text{and} \quad r \in J \setminus \{\, l \,\} \subset B_q \\ &\qquad\qquad \text{by} \quad \left(y_r + \frac{v_r}{v_l} \cdot x_l^{(d_l - \overline{c})} \right)^{(\overline{c} - c_i)}. \end{aligned}$$

For a $j \notin B_q \supset J \setminus \{\, l \,\}$, the corresponding derivatives $x_j^{(d_j - c_i)}$ are not replaced in the ES conversion, and for $r \in J \setminus \{\, l \,\}$ (so j, r, l are distinct),

$$\sigma\left(x_j, \left(y_r + \frac{v_r}{v_l} \cdot x_l^{(d_l - \overline{c})} \right)^{(\overline{c} - c_i)} \right) = \sigma\left(x_j, \left(\frac{v_r}{v_l} \right)^{(\overline{c} - c_i)} \right) \leq \sigma\left(x_j, \mathbf{v}^{(\overline{c} - c_i)} \right). \tag{A.15}$$

By (6.13), $\sigma(x_j, \mathbf{v}) < d_j - \overline{c}$. Using (6.1) and (A.15), we derive

$$\begin{aligned}
\sigma\left(x_j, \overline{f}_i\right) &\le \max\left\{\sigma\left(x_j, f_i\right), \max_{r\in J\setminus\{l\}} \sigma\left(x_j, \left(y_r + \frac{v_r}{v_l}\cdot x_l^{(d_l-\overline{c})}\right)^{(\overline{c}-c_i)}\right)\right\} \\
&\le \max\left\{\sigma\left(x_j, f_i\right),\ \sigma\left(x_j, \mathbf{v}^{(\overline{c}-c_i)}\right)\right\} \\
&= \max\left\{\sigma_{ij},\ \sigma\left(x_j, \mathbf{v}\right) + (\overline{c} - c_i)\right\} \\
&< \max\left\{d_j - c_i,\ (d_j - \overline{c}) + (\overline{c} - c_i)\right\} \\
&= d_j - c_i \qquad\qquad \text{for } i \in M \subseteq B_q. \qquad\qquad \text{(A.16)}
\end{aligned}$$

From the ES conversion described in Theorem 6.4, we have

$$\sigma\left(x_j, \overline{f}_i\right) = \sigma\left(x_j, f_i\right) < d_j - c_i \qquad \text{for } i \in B_q \setminus M \text{ and} \qquad \text{(A.17)}$$

$$\sigma\left(x_j, g_r\right) \le \sigma\left(x_j, v\right) < d_j - \overline{c} \qquad \text{for } r \in J. \qquad \text{(A.18)}$$

Since blocks $\overline{\mathbf{\Sigma}}_{qw_2}$ with $w_2 < q$ contain signature entries $\overline{\sigma}_{ij}$ for equations $\overline{f}_i$ and g_r, where $i \in B_q$ and $r \in J$, in variables x_j with $\text{blockOf}(j) < q$, by taking together the inequalities in (A.16)-(A.18), we have

$$\overline{\sigma}_{ij} < \begin{cases} d_j - c_i & \text{if } \text{blockOf}(j) < q \text{ and } i \in B_q \\ d_j - \overline{c} & \text{if } \text{blockOf}(j) < q \text{ and } i \in Q+1 : Q+s; \end{cases}$$

recall $Q = \sum_{w=1}^{q} N_w$. Using (A.10) and the construction of $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$ in (A.12), we have

$$\overline{\sigma}_{ij} < \widetilde{d}_j - \widetilde{c}_i \quad \text{for} \quad \overline{\text{blockOf}}(j) < \overline{\text{blockOf}}(i) = q. \qquad \text{(A.19)}$$

(q is the block number of both the original and enlarged diagonal blocks.)

(b) $\overline{\mathbf{\Sigma}}_{qw_2}$ with $w_2 > q$. An entry $(i, j+s)$ in this block satisfies $\overline{\text{blockOf}}(j+s) > \overline{\text{blockOf}}(i) = q$. By (A.11), $\text{blockOf}(j) > q$ and hence $j \notin B_q \supset J \setminus \{\, l \,\}$. By the same

arguments as in (a), the corresponding derivatives $x_j^{(d_j - c_i)}$ are not replaced in the ES conversion.

By (6.13), $\sigma(x_j, \mathbf{v}) \leq d_j - \overline{c}$. Then by the same derivations as (A.16–A.18) in (a), we have

$$\sigma\left(x_j, \overline{f}_i\right) \leq d_j - c_i \qquad \text{for } i \in M \subseteq B_q, \tag{A.20}$$

$$\sigma\left(x_j, \overline{f}_i\right) = \sigma(x_j, f_i) \leq d_j - c_i \qquad \text{for } i \in B_q \setminus M, \text{ and} \tag{A.21}$$

$$\sigma(x_j, g_r) \leq \sigma(x_j, \mathbf{v}) \leq d_j - \overline{c} \qquad \text{for } r \in J. \tag{A.22}$$

Since blocks $\overline{\mathbf{\Sigma}}_{qw_2}$ with $w_2 > q$ contain signature entries $\overline{\sigma}_{i,j+s}$ for equations $\overline{f}_i$ and g_r, where $i \in B_q$ and $r \in J$, in variables x_j with $\text{blockOf}(j) > q$, the inequalities (A.20–A.22) yield

$$\overline{\sigma}_{i,j+s} \leq \begin{cases} d_j - c_i & \text{if } \text{blockOf}(j) > q \text{ and } i \in B_q \\ d_j - \overline{c} & \text{if } \text{blockOf}(j) > q \text{ and } i \in Q+1 : Q+s. \end{cases}$$

Using (A.11) and the construction of $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{d}}$ in (A.12), we have

$$\overline{\sigma}_{i,j+s} \leq \widetilde{d}_{j+s} - \widetilde{c}_i \quad \text{for} \quad \overline{\text{blockOf}}(j+s) > \overline{\text{blockOf}}(i) = q,$$

with $j = Q+1 : n$. We can rewrite this inequality as

$$\overline{\sigma}_{ij} \leq \widetilde{d}_j - \widetilde{c}_i \quad \text{for} \quad \overline{\text{blockOf}}(j) > \overline{\text{blockOf}}(i) = q, \tag{A.23}$$

with $j = Q+1+s : n+s$.

For the inequalities in (A.13), so far we have proved the case $\overline{\text{blockOf}}(i) \neq q$ in (A.14) and the case $\overline{\text{blockOf}}(j) \neq \overline{\text{blockOf}}(i) = q$ in (A.19) and (A.23). For the

last case $\overline{\text{blockOf}}(j) = \overline{\text{blockOf}}(i) = q$, we use the results from the ES method in Appendix A.1.

(c) $\overline{\mathbf{\Sigma}}_{qw_2}$ is $\overline{\mathbf{\Sigma}}_{qq}$, with $w_2 = q$. An entry (i, j) in $\overline{\mathbf{\Sigma}}_{qq}$ satisfies $\overline{\text{blockOf}}(j) = \overline{\text{blockOf}}(i) = q$. We view block q of the original DAE as a sub-DAE, with a signature matrix $\mathbf{\Sigma}_{qq}$ of size N_q and an offset pair $(\mathbf{c}_q; \mathbf{d}_q)$. Given that the ES conditions are satisfied by (6.13), performing the ES conversion as described in Theorem 6.4 is equivalent to applying the basic ES method to this sub-DAE. After a conversion, the resulting enlarged signature matrix $\overline{\mathbf{\Sigma}}_{qq}$ of size $N_q + s$ has the form

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

cf. (A.4) in Appendix A.1, Figure 4.1, and Figure 6.1. The two block rows of $\overline{\mathbf{\Sigma}}_{qq}$ correspond to $\overline{f}_i$ for $i \in B_q$ and g_j for $j \in J$, respectively. The three block columns of $\overline{\mathbf{\Sigma}}_{qq}$ correspond to x_j for $j \in J$, x_j for $j \in B_q \setminus J$, and y_j for $j \in J$, respectively. If we apply the same arguments in the proof of Lemma 4.19 (in Appendix A.1) for the basic ES method, then we have $\widetilde{d}_j - \widetilde{c}_i \geq \overline{\sigma}_{ij}$ for all entries in $\overline{\mathbf{\Sigma}}_{qq}$. □

Appendix B

Alternative proof of Theorem 6.1

Let the notation be as at the start of §6.2. This proof is based on the following lemma.

Lemma B.1 *Assume that* $\mathbf{\Sigma}$ *has a finite* $\mathrm{Val}(\mathbf{\Sigma})$, $(\mathbf{c};\mathbf{d})$ *is a valid offset pair, and* $\mathbf{S}_0$, *derived from* $\mathbf{\Sigma}$ *and* $(\mathbf{c};\mathbf{d})$, *is in a* $p \times p$ *BTF. Given a row index* $l \in B_q$, *where* $q \in 1:p$, *if we replace the entries* σ_{lj} *in this row by*

$$\overline{\sigma}_{lj} \begin{cases} < d_j - c_l & \text{if } \mathit{blockOf}(j) \le q \\ \le d_j - c_l & \text{if } \mathit{blockOf}(j) > q, \end{cases} \tag{B.1}$$

then $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$, *where* $\overline{\mathbf{\Sigma}} = (\overline{\sigma}_{ij})$ *is the resulting signature matrix.*

Proof. Let T be a HVT of $\mathbf{\Sigma}$, and let $\overline{T}$ be a HVT of $\overline{\mathbf{\Sigma}}$. By (B.1),

$$\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} \le \sum_{(i,j)\in\overline{T}} (d_j - c_i) = \sum_{j=1}^{n} d_j - \sum_{i=1}^{n} c_i = \mathrm{Val}(\mathbf{\Sigma}). \tag{B.2}$$

We show by contradiction that an equality in "$\le$" of (B.2) cannot be achieved. We

first outline the steps of the proof. (a) Assuming $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma})$, we show that a valid $(\mathbf{c};\mathbf{d})$ of $\mathbf{\Sigma}$ is also valid for $\overline{\mathbf{\Sigma}}$. (b) Using Lemma 2.10, we show that the Jacobian patterns $\mathbf{S}_0$, derived from $\mathbf{\Sigma}$ and $(\mathbf{c};\mathbf{d})$, and $\overline{\mathbf{S}}_0$, derived from $\overline{\mathbf{\Sigma}}$ and $(\mathbf{c};\mathbf{d})$, are in the same $p \times p$ BTF. Then we can write $\overline{T} = \overline{T}_1 \cup \cdots \cup \overline{T}_p$, where $\overline{T}_w$ is an HVT of a diagonal block $\overline{\mathbf{\Sigma}}_{ww}$, $w = 1:p$. (c) Finally we show that a contradiction occurs at the intersection of row l in $\overline{\mathbf{\Sigma}}$ with the HVT $\overline{T}_q$ of diagonal block $\overline{\mathbf{\Sigma}}_{qq}$.

Now we elaborate on each step.

(a) We start off by assuming $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \sum_{(i,j)\in\overline{T}} \overline{\sigma}_{ij} = \mathrm{Val}(\mathbf{\Sigma}) = \sum_j d_j - \sum_i c_i \geq 0$. In this case, each entry $\overline{\sigma}_{ij}$ on $\overline{T}$ is finite. Also, since $d_j - c_i \geq \overline{\sigma}_{ij}$ holds everywhere by the construction of $\overline{\mathbf{\Sigma}}$, an equality $d_j - c_i = \overline{\sigma}_{ij}$ holds for each $(i,j) \in \overline{T}$. Hence $(\mathbf{c};\mathbf{d})$ is a valid offset pair of $\overline{\mathbf{\Sigma}}$.

(b) Since $\mathbf{S}_0$ is in a BTF, $d_j - c_l > \sigma_{lj}$ if $\mathrm{blockOf}(j) < q$ and $d_j - c_l \geq \sigma_{lj}$ if $\mathrm{blockOf}(j) \geq q$. Also by (B.1), $d_j - c_i > \overline{\sigma}_{ij}$ holds in the below diagonal blocks of $\overline{\mathbf{\Sigma}}$. Given that $d_j - c_i \geq \overline{\sigma}_{ij}$ holds elsewhere and that $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \sum_j d_j - \sum_i c_i$, by Lemma 2.10, the Jacobian pattern $\overline{\mathbf{S}}_0$, derived from $\overline{\mathbf{\Sigma}}$ and $(\mathbf{c};\mathbf{d})$, is in the same $p \times p$ BTF as is $\mathbf{S}_0$, derived from $\mathbf{\Sigma}$ and $(\mathbf{c};\mathbf{d})$. Hence, $\overline{T}$ is the union of HVTs $\overline{T}_w$ of diagonal blocks $\overline{\mathbf{\Sigma}}_{ww}$ of $\overline{\mathbf{\Sigma}}$, $w = 1:p$.

(c) If we intersect row l in $\overline{\mathbf{\Sigma}}$ with an HVT $\overline{T}_q$ of $\overline{\mathbf{\Sigma}}_{qq}$, then we obtain a position $(l,r) \in \overline{T}_q \subseteq \overline{T}$ with $l \in B_q$, $r \in B_q$, and $d_r - c_l = \overline{\sigma}_{lr}$. However, this equality contradicts (B.1) that requires $d_r - c_l > \overline{\sigma}_{lr}$. That is, the assumption $\mathrm{Val}(\overline{\mathbf{\Sigma}}) = \mathrm{Val}(\mathbf{\Sigma})$ leads to a contradiction, so $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$. □

Using Lemma B.1, we give the alternative proof of Theorem 6.1.

Proof. We verify below that $\overline{\mathbf{\Sigma}} = (\overline{\sigma}_{ij})$ satisfies (B.1). Then by Lemma B.1, $\mathrm{Val}(\overline{\mathbf{\Sigma}}) < \mathrm{Val}(\mathbf{\Sigma})$.

Obviously, we only replace f_l by $\overline{f}_l = \overline{f}$ in an LC conversion, so $\overline{\sigma}_{ij} = \sigma_{ij}$ for all $i \neq l$ and all j. We consider three cases: (l, j) is (a) below the block diagonal, (b) above the block diagonal, or (c) in diagonal block q.

Recall that, in the proof of Lemma 6.2, we have proved the cases (a) and (b) in (B.1). What remains to show is $\overline{\sigma}_{lj} < d_j - c_l$ for $\text{blockOf}(j) = q$.

For $j \in B_q$, we derive

$$\begin{aligned}
\frac{\partial \overline{f}_l}{\partial x_j^{(d_j - \underline{c})}} &= \frac{\partial \left(\sum_{i \in I} u_i f_i^{(c_i - \underline{c})} \right)}{\partial x_j^{(d_j - \underline{c})}} = \sum_{i \in I} u_i \frac{\partial f_i^{(c_i - \underline{c})}}{\partial x_j^{(d_j - \underline{c})}} && \text{using (6.4) and (6.5)} \\
&= \sum_{i \in I} u_i \frac{\partial f_i}{\partial x_j^{(d_j - c_i)}} = \sum_{i \in I} u_i J_{ij} && \text{using Lemma 4.2} \\
&= 0 && \text{using } \widehat{\mathbf{u}} \in \text{coker}(\mathbf{J}_{qq}).
\end{aligned}$$

Hence $\overline{\sigma}_{lj} = \sigma\left(x_j, \overline{f}_l\right) < d_j - \underline{c} = d_j - c_l$ for all $j \in B_q$. □

Bibliography

[1] Ascher, U.M., Petzold, L.R.: Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM, Philadelphia (1998)

[2] Barrio, R.: Performance of the Taylor series method for ODEs/DAEs. Appl. Math. Comp. **163**, 525–545 (2005)

[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., Gear, C.: The index of general nonlinear DAEs. Numerische Mathematik **72**, 173–196 (1995)

[5] Campbell, S.L., Griepentrog, E.: Solvability of general differential-algebraic equations. SIAM Journal on Scientific Computing **16**(2), 257–270 (1995)

[6] 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)

[7] 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)

[8] Corless, R.M., Ilie, S.: Polynomial cost for solving IVP for high-index DAE. BIT Numerical Mathematics **48**(1), 29–49 (2008). DOI 10.1007/s10543-008-0163-2. URL http://dx.doi.org/10.1007/s10543-008-0163-2

[9] Duff, I., Erisman, A., Reid, J.: Direct Methods for Sparse Matrices. Oxford Science Publications. Clarendon Press, Oxford (1986)

[10] Duff, I., Gear, C.: Computing the structural index. SIAM Journal on Algebraic and Discrete Methods **7**, 594ê603 (1986)

[11] ESI ITI GmbH: SimulationX (2016). https://www.simulationx.com/

[12] Estvez Schwarz, D., Lamour, R.: Diagnosis of singular points of properly stated DAEs using automatic differentiation. Numerical Algorithms pp. 1–29 (2015). DOI 10.1007/s11075-015-9973-x. URL http://dx.doi.org/10.1007/s11075-015-9973-x

[13] Gear, C.W.: Differential-algebraic equation index transformations. SIAM Journal on Scientific and Statistical Computing **9**(1), 39–47 (1988)

[14] Gear, C.W.: Differential algebraic equations, indices, and integral algebraic equations. SIAM Journal on Numerical Analysis **27**(6), 1527–1534 (1990)

[15] Griepentrog, E., März, R.: Differential-algebraic equations and their numerical treatment. In: Teubner-Texte zur Mathematik [Teubner Texts in Mathematics],

88. BSB B. G. Teubner Verlagsgesellschaft, Leipzig (1986). With German, French and Russian summaries

[16] Griewank, A., Walther, A.: On the efficient generation of Taylor expansions for DAE solutions by automatic differentiation. In: Applied Parallel Computing. State of the Art in Scientific Computing, pp. 1089–1098. Springer (2006)

[17] Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II. Stiff and Differential– Algebraic Problems. Springer Verlag, Berlin (1991)

[18] Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: Sundials: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans. Math. Softw. **31**(3), 363–396 (2005). DOI 10.1145/1089014.1089020. URL http://doi.acm.org/10.1145/1089014.1089020

[19] Hoefkens, J.: Rigorous numerical analysis with high-order taylor models. Ph.D. thesis, Department of Mathematics and Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824 (2001)

[20] JModelica.org: The JModelica.org Platform: Jmodelica web page. http://www.jmodelica.org

[21] 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)

[22] Kunkel, P., Mehrmann, V.L.: Differential-algebraic equations: analysis and numerical solution. European Mathematical Society, Zürich, Switzerland (2006)

[23] Lamour, R., März, R.: Detecting structures in differential-algebraic equations: Computational aspects. Journal of Computational and Applied Mathematics **236**(16), 4055 – 4066 (2012). DOI http://dx.doi.org/10.1016/j.cam.2012.03.009. URL http://www.sciencedirect.com/science/article/pii/S0377042712001288

[24] MapleSim: Technological superiority in multi-domain physical modeling and simulation (2012). http://www.maplesoft.com/view.aspx?sf=7032

[25] Maplesoft: Maplesim web page. http://www.maplesoft.com/products/maplesim/

[26] Mattsson, S.E., Söderlind, G.: Index reduction in differential-algebraic equations using dummy derivatives. SIAM J. Sci. Comput. **14**(3), 677–692 (1993)

[27] 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/

[28] McKenzie, R.: Structural analysis based dummy derivative selection for differential-algebraic equations. Tech. rep., Cardiff University, UK (2015). Submitted to BIT Numerical Mathematics, October 2015

[29] McKenzie, R.: Reducing the index of differential-algebraic equations by exploiting underlying structures. PhD Thesis, School of Mathematics, Cardiff University, Senghennydd Road, Cardiff CF24 4AG, UK (2016)

[30] McKenzie, R., Nedialkov, N., Pryce, J., Tan, G.: DAESA user guide. Tech. rep., Department of Computing and Software, McMaster University, Hamilton, ON,

L8S 4K1, Canada (2013). 47 pages, DAESA is available at http://www.cas.mcmaster.ca/~nedialk/daesa

[31] McKenzie, R., Pryce, J.D.: Structural analysis and dummy derivatives: Some relations. In: G.M. Cojocaru, S.I. Kotsireas, N.R. Makarov, N.R.V. Melnik, H. Shodiev (eds.) Interdisciplinary Topics in Applied Mathematics, Modeling and Computational Science, pp. 293–299. Springer International Publishing, Cham (2015). DOI 10.1007/978-3-319-12307-3_42. URL http://dx.doi.org/10.1007/978-3-319-12307-3_42

[32] McKenzie, R., Pryce, J.D.: Solving differential-algebraic equations by selecting universal dummy derivatives. In: J. Bélair, A.I. Frigaard, H. Kunze, R. Makarov, R. Melnik, J.R. Spiteri (eds.) Mathematical and Computational Approaches in Advancing Modern Science and Engineering, pp. 665–676. Springer International Publishing, Cham (2016). DOI 10.1007/978-3-319-30379-6_60. URL http://dx.doi.org/10.1007/978-3-319-30379-6_60

[33] 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

[34] 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)

[35] Nedialkov, N.S., Pryce, J.D.: Solving differential-algebraic equations by Taylor series (II): Computing the system Jacobian. BIT Numerical Mathematics **47**(1), 121–135 (2007)

[36] 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)

[37] 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)

[38] Nedialkov, N.S., Tan, G., Pryce, J.D.: Exploiting fine block triangularization and quasilinearity in differential-algebraic equation systems (2016). McMaster University, Cardiff University. In preparation

[39] OpenModelica: OpenModelica web page. http://www.openmodelica.org

[40] Pantelides, C.C.: The consistent initialization of differential-algebraic systems. SIAM J. Sci. Stat. Comput. **9**, 213–231 (1988)

[41] Pryce, J.D.: Solving high-index DAEs by Taylor Series. Numerical Algorithms **19**, 195–211 (1998)

[42] Pryce, J.D.: A simple structural analysis method for DAEs. BIT Numerical Mathematics **41**(2), 364–394 (2001)

[43] Pryce, J.D.: A simple approach to Dummy Derivatives for DAEs. Tech. rep., Cardiff University (2015). In preparation

[44] Pryce, J.D., McKenzie, R.: A new look at dummy derivatives for differential-algebraic equations. In: J. Bélair, A.I. Frigaard, H. Kunze, R. Makarov, R. Melnik, J.R. Spiteri (eds.) Mathematical and Computational Approaches in Advancing Modern Science and Engineering, pp. 713–723. Springer International Publishing, Cham (2016). DOI 10.1007/978-3-319-30379-6_64. URL http://dx.doi.org/10.1007/978-3-319-30379-6_64

[45] Pryce, J.D., Nedialkov, N.S.: How automatic differentiation can help solve differential-algebraic equations (2016). Abstract

[46] 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)

[47] Pryce, J.D., Nedialkov, N.S., Tan, G.: Fine block triangular structure of DAEs and its uses (2016). Cardiff University, McMaster University. In preparation

[48] Rabier, P.J., Rheinboldt, W.C.: A general existence and uniqueness theory for implicit differential-algebraic equations. Differential Integral Equations **4**(3), 563–582 (1991). URL http://projecteuclid.org/euclid.die/1372700430

[49] 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)

[50] Rheinboldt, W.C.: Differential-algebraic systems as differential equations on manifolds. Mathematics of computation **43**(168), 473–482 (1984)

[51] Richardson, D.: Some undecidable problems involving elementary functions of a real variable. J. Symbolic Logic **33**(4), 514–520 (1968)

[52] 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)

[53] 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)

[54] Soares, R.d.P., Secchi, A.R.: Structural analysis for static and dynamic models. Mathematical and Computer Modelling **55**(3), 1051–1067 (2012)

[55] The MathWorks, Inc.: Matlab (2016). http://www.mathworks.com/products/matlab/

[56] The MathWorks, Inc.: Matlab Symbolic Math Toolbox (2016). http://www.mathworks.com/products/symbolic/

[57] The MathWorks, Inc.: Simulink (2016). http://www.mathworks.com/products/simulink/

[58] The Modelica Organization: Modelica web page. http://www.modelica.org

[59] Unger, J., Kröner, A., Marquardt, W.: Structural analysis of differential-algebraic equation systems – theory and applications. Computers & Chemical Engineering **19**(8), 867–882 (1995)