MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Index Reduction via Unimodular Transformations

Satoru IWATA and Mizuyo TAKAMATSU

METR 2017–05 February 2017

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Index Reduction via Unimodular Transformations*

Satoru Iwata[†] Mizuyo Takamatsu[‡]

February 2017

Abstract

This paper presents an algorithm for transforming a matrix pencil $A(s)$ into another matrix pencil $U(s)A(s)$ with a unimodular matrix $U(s)$ so that the resulting Kronecker index is at most one. The algorithm is based on the framework of combinatorial relaxation, which combines graph-algorithmic techniques and matrix computation. Our algorithm works for index reduction of linear differential-algebraic equations, including those for which the existing index reduction methods based on Pantelides' algorithm are known to fail.

1 Introduction

A matrix pencil is a polynomial matrix in which the degree of each entry is at most one. By a strict equivalence transformation, each matrix pencil can be brought into its Kronecker canonical form (KCF). Numerically stable computation of KCF is a challenging problem, which has required enormous efforts [2, 4, 5, 9, 21].

Let $A(s)$ be an $n \times n$ matrix pencil. The Kronecker index $\nu(A)$ of $A(s)$ is defined in terms of the KCF of $A(s)$. Previous work given in [7, 8, 14, 19] aims at finding $\nu(A)$ without obtaining the KCF. They utilize the following combinatorial characterization:

$$\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1. \tag{1}$$

Here, $\delta_k(A)$ denotes the maximum degree of minors of order k in $A(s)$, i.e.,

$$\delta_k(A) = \max\{\deg \det A(s)[I, J] \mid |I| = |J| = k\}, \tag{2}$$

where $\deg a(s)$ designates the degree of a polynomial $a(s)$ and $A(s)[I, J]$ denotes the submatrix with row set I and column set J.

While the previous work [7, 8, 14, 19] deals with the index computation, this paper focuses on the index reduction of a matrix pencil. Our aim is to transform $A(s)$ into another matrix pencil with the Kronecker index at most one. More precisely, we present an algorithm for finding a unimodular polynomial matrix $U(s)$ such that $U(s)A(s)$ is a matrix pencil with $\nu(UA) \leq 1$.

*This research is supported by CREST, JST.

†Department of Mathematical Informatics, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan. E-mail: iwata@mist.i.u-tokyo.ac.jp

‡Department of Information and System Engineering, Chuo University, Kasuga 1-13-27, Bunkyo-ku, Tokyo 112-8551, Japan. This author's research is supported in part by JSPS KAKENHI Grant Number 25730009. E-mail: takamatsu@ise.chuo-u.ac.jp

Once the KCF of $A(s)$ is obtained together with the transformation matrices, it is straightforward to construct such a unimodular matrix $U(s)$. Since numerical difficulty is inherent in the computation of KCF, we aim at finding $U(s)$ more directly without relying on the KCF. Instead of computing the KCF, our algorithm makes use of (1).

Our motivation comes from the study of differential-algebraic equations (DAEs) [1, 3, 6, 11, 18]. Consider a linear DAE

$$F\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} + H\boldsymbol{z}(t) = \boldsymbol{g}(t) \tag{3}$$

with an initial condition $\boldsymbol{z}(0) = \boldsymbol{z}_0$, where F and H are constant matrices. By the Laplace transformation, we obtain

$$A(s)\tilde{\boldsymbol{z}}(s) = \tilde{\boldsymbol{g}}(s) + F\boldsymbol{z}_0$$

with the matrix pencil $A(s) = sF + H$. The numerical difficulty of the DAE (3) is measured by the Kronecker index $\nu(A)$.

A common approach for solving a high index DAE is to transform it into an equivalent DAE with index at most one, which can be solved easily by numerical methods including the backward differentiation formulas (BDF). This motivates a variety of index reduction algorithms, in which we are allowed to differentiate a certain equation and add it to another equation. Such an operation corresponds to equivalence row transformations with unimodular polynomial matrix $U(s)$. The Laplace transform of the resulting DAE is in the form of

$$U(s)A(s)\tilde{\boldsymbol{z}}(s) = U(s)(\tilde{\boldsymbol{g}}(s) + F\boldsymbol{z}_0).$$

If $U(s)A(s)$ is a matrix pencil and $\nu(UA) \leq 1$ holds, the index of the DAE is now reduced to at most one.

The modeling and simulation software for dynamical systems, such as Dymola, OpenModelica, and MapleSim, is equipped with the index reduction methods based on Pantelides' algorithm [16], the dummy derivative approach [12], or the signature method [17]. These algorithms adopt a structural approach, which extracts zero/nonzero pattern of coefficients in equations, ignoring the numerical values. Such algorithms are efficient, because they exploit graph-algorithmic techniques. However, the discard of numerical information can cause a failure even for linear DAEs. In contrast, our algorithm always works for any instances of linear DAEs.

The algorithms for computing $\delta_k(A)$ given in [7, 8, 14, 19] are based on the framework of "combinatorial relaxation," which combines graph-algorithmic techniques and matrix computation. In combinatorial relaxation algorithms, we find an estimate $\hat{\delta}_k(A)$ of $\delta_k(A)$ by solving a matching problem and check if $\hat{\delta}_k(A) = \delta_k(A)$ by constant matrix computation. If $\hat{\delta}_k(A) \neq \delta_k(A)$, then we modify $A(s)$ to improve $\hat{\delta}_k(A)$ without changing $\delta_k(A)$. After a finite number of iterations, the algorithms terminate with $\hat{\delta}_k(A) = \delta_k(A)$. They mainly rely on fast combinatorial algorithms and perform numerical computation only when necessary.

Our index reduction algorithm, which consists of two phases, inherits the idea of combinatorial relaxation. In the first phase, we transform $A(s)$ into another matrix pencil $\tilde{A}(s)$ such that an estimate of $\nu(\tilde{A})$ is at most one. In the second phase, we determine if the estimate is correct. If not, we further transform $\tilde{A}(s)$ into another matrix pencil $\hat{A}(s)$ with $\nu(\hat{A}) \leq 1$. In the both phases, we exploit a feasible dual solution of the matching problem, which was also used by Pryce [17] in the interpretation of Pantelides' algorithm [16].

The rest of this paper is organized as follows. In Section 2, we explain the bipartite matching problems associated with matrix pencils. We present an index reduction algorithm in Section 3. Section 4 gives numerical examples, and Section 5 concludes this paper.

2 Matrix Pencils and Matching Problems

For a polynomial $a(s)$, we denote the degree of $a(s)$ by $\deg a$, where $\deg 0 = -\infty$ by convention. A polynomial matrix $A(s) = (a_{ij}(s))$ with $\deg a_{ij} \leq 1$ for all (i,j) is called a *matrix pencil*. A matrix pencil $A(s)$ is said to be *regular* if $A(s)$ is square and $\det A(s)$ is a nonvanishing polynomial.

Let us denote by block-diag$(D_1, \ldots, D_b)$ the block-diagonal matrix pencil with diagonal blocks $D_1, \ldots, D_b$. By a strict equivalence transformation, a regular matrix pencil $A(s)$ can be brought into its Kronecker canonical form block-diag$(sI_{\mu_0} + J_{\mu_0}, N_{\mu_1}, \ldots, N_{\mu_d})$, where I_{μ_0} is a $\mu_0 \times \mu_0$ identity matrix, J_{μ_0} is a $\mu_0 \times \mu_0$ constant matrix, and N_μ is a $\mu \times \mu$ matrix pencil defined by

$$N_\mu = \begin{pmatrix} 1 & s & 0 & \cdots & 0 \\ 0 & 1 & s & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & s \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

The matrices $N_{\mu_1}, \ldots, N_{\mu_d}$ are called the *nilpotent blocks*.

For a matrix pencil $A(s)$, the Kronecker index $\nu(A)$ is defined to be the maximum size of the nilpotent blocks in the Kronecker canonical form of $A(s)$, i.e., $\max_{1 \leq i \leq d} \mu_i$. It is known [15, Theorem 5.1.8] that $\nu(A)$ is expressed by (1).

A polynomial matrix is called *unimodular* if it is square and its determinant is a nonvanishing constant. This implies that a square polynomial matrix is unimodular if and only if its inverse is a polynomial matrix.

Let $A(s)$ be an $n \times n$ regular matrix pencil with row set R and column set C. We construct a bipartite graph $G(A) = (R, C; E(A))$ with $E(A) = \{(i,j) \mid i \in R, j \in C, A_{ij}(s) \neq 0\}$. The weight c_e of an edge $e = (i,j)$ is given by $c_e = c_{ij} = \deg A_{ij}(s)$. We remark that c_e is 0 or 1 for each $e \in E(A)$ because $A(s)$ is a matrix pencil. A subset M of $E(A)$ is called a *matching* if every pair of edges in M are disjoint. A matching M is called a *perfect matching* if M covers all the vertices.

Consider the following maximum-weight perfect matching problem $\mathrm{P}(A)$:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{e \in M} c_e \\ \text{subject to} & M \text{ is a perfect matching.} \end{array}$$

Since $A(s)$ is regular, $G(A)$ has a perfect matching. The maximum weight of a perfect matching in $G(A)$, denoted by $\hat{\delta}_n(A)$, is an upper bound on $\delta_n(A)$.

The dual problem D(A) of P(A) is given by

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in R} p_i - \sum_{j\in C} q_j \\
\text{subject to} \quad & p_i - q_j \geq c_e \quad (e=(i,j)\in E(A)), \\
& p_i \in \mathbb{Z} \quad (i \in R), \\
& q_j \in \mathbb{Z} \quad (j \in C).
\end{aligned}$$

We denote the objective function of D(A) by $\Delta_n(p,q)$.

We construct an optimal solution (p,q) of D(A) as follows. Let M be a maximum-weight perfect matching in $G(A)=(R,C;E(A))$. The reorientation of $a\in E(A)$ is denoted by $\bar{a}$. Consider an auxiliary graph $\check{G}_M=(\check{V},\check{E})$ with $\check{V}=R\cup C\cup\{r\}$ and $\check{E}=\bar{E}\cup M\cup W$, where r is a new vertex, $\bar{E}=\{\bar{a}\mid a\in E(A)\}$, and $W=\{(r,i)\mid i\in R\}$. We define the arc length $\gamma:\check{E}\to\mathbb{Z}$ by

$$\gamma(a)=\begin{cases}-c_{\bar{a}} & (a\in\bar{E})\\ c_a & (a\in M)\\ 0 & (a\in W)\end{cases}.$$

Let $d(i,j)$ be the shortest distance from $i\in\check{V}$ to $j\in\check{V}$ with respect to the arc length γ in $\check{G}_M$. We define

$$p_i = -d(r,i)+\max_{\ell\in C} d(r,\ell) \quad (i\in R), \tag{4}$$

$$q_j = -d(r,j)+\max_{\ell\in C} d(r,\ell) \quad (j\in C). \tag{5}$$

Lemma 2.1. Suppose that (p,q) is defined by (4) and (5). Then (p,q) is an optimal solution of D(A) satisfying

$$\min_{i\in R} p_i \geq 0, \quad \min_{j\in C} q_j = 0, \quad \max_{j\in C} q_j \leq n. \tag{6}$$

Proof. By the definition of (p,q), $p_i\in\mathbb{Z}$ $(i\in R)$ and $q_j\in\mathbb{Z}$ $(j\in C)$ clearly hold. For $e=(i,j)\in E(A)$, we have $d(r,i)\leq d(r,j)-c_e$. Hence

$$p_i - q_j = -d(r,i)+d(r,j) \geq c_e \tag{7}$$

holds. Thus (p,q) is a feasible solution of D(A).

Since $\check{G}_M$ has both arcs (i,j) and (j,i) for $e=(i,j)\in M$, we obtain

$$p_i - q_j = -d(r,i)+d(r,j) = c_e. \tag{8}$$

It follows from $|R|=|C|$ and (8) that

$$\sum_{i\in R} p_i - \sum_{j\in C} q_j = -\sum_{i\in R} d(r,i) + \sum_{j\in C} d(r,j) = \sum_{(i,j)\in M} (-d(r,i)+d(r,j)) = \sum_{e\in M} c_e,$$

which implies that (p,q) is optimal to D(A).

Finally, we show that (p, q) satisfies (6). The second condition follows from the definition of q_j. Since $G(A)$ has a perfect matching, each $i \in R$ is incident to at least one vertex $j \in C$. Hence we have $p_i \geq q_j + c_{ij} \geq 0$ by (7), $q_j \geq 0$, and $c_{ij} \geq 0$. This implies $\min_{i \in R} p_i \geq 0$. Let P_j and P_ℓ denote the shortest paths from r to j and ℓ, respectively. Let v be the last common vertex in P_j and P_ℓ. Then $d(r,\ell) - d(r,j) = d(v,\ell) - d(v,j)$. Note that $d(v,\ell)$ is at most the number of arcs in M between v and ℓ along P_ℓ, whereas $-d(v,j)$ is at most the number of arcs in $\bar{E}$ between v and j along P_j. The sum of these upper bounds is at most n. Thus we obtain $q_j \leq n$ for every $j \in C$. □

Next, consider the following matching problem corresponding to $\delta_{n-1}(A)$.

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{e \in M} c_e \\ \text{subject to} & M \text{ is a matching,} \\ & |M| = n-1. \end{array}$$

The optimal value is denoted by $\hat{\delta}_{n-1}(A)$, which is an upper bound on $\delta_{n-1}(A)$.

For a feasible solution (p, q) of D(A), we define

$$\Delta_{n-1}(p,q) = \Delta_n(p,q) - \min_{i \in R} p_i + \max_{j \in C} q_j.$$

The following lemma gives upper bounds on $\hat{\delta}_n(A)$ and $\hat{\delta}_{n-1}(A)$.

Lemma 2.2. For a feasible solution (p, q) of D(A), we have

$$\hat{\delta}_n(A) \leq \Delta_n(p,q), \quad \hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p,q).$$

Proof. Let M_n^* denote a maximum-weight matching of size n. Since (p, q) is a feasible solution of D(A), we have $p_i - q_j \geq c_e$ for $e = (i,j) \in E(A)$. The former follows from the weak duality for the maximum-weight perfect matching problem:

$$\hat{\delta}_n(A) = \sum_{e \in M_n^*} c_e \leq \sum_{(i,j) \in M_n^*} (p_i - q_j) = \sum_{i \in R} p_i - \sum_{j \in C} q_j = \Delta_n(p,q).$$

We now prove the latter. Let M_{n-1}^* denote a maximum-weight matching of size $n-1$. Let ∂M_{n-1}^* denote the set of vertices incident to M_{n-1}^*. Then we have

$$\begin{aligned} \hat{\delta}_{n-1}(A) = \sum_{e \in M_{n-1}^*} c_e &\leq \sum_{(i,j) \in M_{n-1}^*} (p_i - q_j) = \sum_{i \in R \cap \partial M_{n-1}^*} p_i - \sum_{j \in C \cap \partial M_{n-1}^*} q_j \\ &\leq \sum_{i \in R} p_i - \min_{i \in R} p_i - \sum_{j \in C} q_j + \max_{j \in C} q_j = \Delta_{n-1}(p,q). \end{aligned}$$

□

3 Index Reduction Algorithm

3.1 Outline of Algorithm

Let $A(s)$ be an $n \times n$ regular matrix pencil, and (p, q) be a feasible solution of $\mathrm{D}(A)$ satisfying (6). By Lemma 2.2, we have

$$\delta_n(A) \le \hat{\delta}_n(A) \le \Delta_n(p, q), \tag{9}$$

$$\delta_{n-1}(A) \le \hat{\delta}_{n-1}(A) \le \Delta_{n-1}(p, q). \tag{10}$$

Our aim is to find a unimodular matrix $U(s)$ such that $\bar{A}(s) = U(s)A(s)$ is a matrix pencil with index $\nu(\bar{A}) \le 1$. The following algorithm updates a matrix pencil $A(s)$ and a feasible solution (p, q). The upper bounds $\Delta_n(p, q)$ and $\Delta_{n-1}(p, q)$ are non-increasing. The resulting matrix pencil $\bar{A}(s)$ and its feasible solution $(\bar{p}, \bar{q})$ satisfy

$$\delta_n(\bar{A}) = \hat{\delta}_n(\bar{A}) = \Delta_n(\bar{p}, \bar{q}), \quad \delta_{n-1}(\bar{A}) = \hat{\delta}_{n-1}(\bar{A}) = \Delta_{n-1}(\bar{p}, \bar{q}), \tag{11}$$

$$\bar{p}_i \in \{0, 1\} \quad (i \in R), \qquad \bar{q}_j = 0 \quad (j \in C). \tag{12}$$

We describe the outline of the index reduction algorithm. The algorithm consists of two phases. In the first phase, we make use of

$$\hat{\nu}(p, q) := \Delta_{n-1}(p, q) - \Delta_n(p, q) + 1$$

as an estimate of $\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1$. At the end of the first phase, we obtain an updated matrix pencil $A(s)$ and a feasible solution (p, q) with $\hat{\nu}(p, q) \le 1$. It should be remarked that this does not imply $\nu(A) \le 1$, because $\hat{\nu}(p, q)$ is not an upper bound on $\nu(A)$.

In the second phase, we check if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ hold without computing $\delta_n(A)$ and $\delta_{n-1}(A)$ directly. If these equations hold, we obtain

$$\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1 = \Delta_{n-1}(p, q) - \Delta_n(p, q) + 1 = \hat{\nu}(p, q) \le 1.$$

If not, we further update $A(s)$ to another matrix pencil. A formal description is as follows.

Outline of Index Reduction Algorithm

Step 1: Construct an optimal solution (p, q) of $\mathrm{D}(A)$ satisfying (6).

Step 2: If $q_j = 0$ for every $j \in C$, then go to Step 4.

Step 3: Bring $A(s)$ into another matrix pencil $\tilde{A}(s)$ by a unimodular transformation, and construct a feasible solution $(\tilde{p}, \tilde{q})$ of $\mathrm{D}(\tilde{A})$ from (p, q). Set $A(s) \leftarrow \tilde{A}(s)$ and $(p, q) \leftarrow (\tilde{p}, \tilde{q})$. Go back to Step 2.

Step 4: If both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ hold, then terminate.

Step 5: Bring $A(s)$ into another matrix pencil $\hat{A}(s)$ by a unimodular transformation, and construct a feasible solution $(\hat{p}, \hat{q})$ of $\mathrm{D}(\hat{A})$ from (p, q). Set $A(s) \leftarrow \hat{A}(s)$ and $(p, q) \leftarrow (\hat{p}, \hat{q})$. Go back to Step 4.

Phase 1 corresponds to Steps 1–3, while Phase 2 corresponds to Steps 4–5. In Steps 1–3, we aim at constructing a feasible solution (p, q) satisfying (12), which implies $\hat{\nu}(p, q) \leq 1$. Then we further update p to obtain a feasible solution satisfying (11) in Steps 4–5. The details of Steps 3–5 are given in Sections 3.2–3.4, respectively.

3.2 Unimodular Transformations in Step 3

We describe how to construct $(\tilde{p}, \tilde{q})$ from a feasible solution (p, q) of $\mathrm{D}(A)$ satisfying (6) in Step 3. For nonnegative integer h, we define

$$R_h = \{i \in R \mid p_i = h\}, \quad C_h = \{j \in C \mid q_j = h\}.$$

Then $A(s)$ is expressed as

$$A(s) = \begin{array}{c} \\ R_\eta \\ R_{\eta-1} \\ \vdots \\ \vdots \\ R_1 \\ R_0 \end{array} \begin{array}{c} \begin{array}{cccccc} C_\eta & C_{\eta-1} & C_{\eta-2} & \cdots & C_1 & C_0 \end{array} \\ \begin{pmatrix} * & ** & ** & \cdots & \cdots & ** \\ O & * & ** & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & ** \\ O & \cdots & \cdots & O & * & ** \\ O & \cdots & \cdots & O & O & * \end{pmatrix} \end{array}$$

for some η, where $*$ and $**$ denote a constant matrix and a matrix pencil, respectively. Since $A(s)$ is regular, the submatrix $A[R_0, C_0]$ is of full-row rank, and hence we can express it as $\begin{pmatrix} * & H_0 \end{pmatrix}$ with a nonsingular constant matrix H_0.

Next, consider the submatrix

$$\begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} C_0 \\ \begin{pmatrix} ** & sF_1 + H_1 \\ * & H_0 \end{pmatrix} \end{array}$$

with constant matrices F_1 and H_1. By multiplying a unimodular matrix $\begin{pmatrix} I & -sF_1H_0^{-1} \\ O & I \end{pmatrix}$ from the left, we obtain

$$\begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} C_0 \\ \begin{pmatrix} sF_2 + H_2 & H_1 \\ * & H_0 \end{pmatrix} \end{array}$$

with constant matrices F_2 and H_2. Since $A[R_0, C_1] = O$, this transformation does not change $A[R_1, C_1]$.

Then consider the submatrix $\left(\begin{array}{c|c} sF_2 + H_2 & H_1 \end{array} \right)$, which can be transformed into

$$\left(\begin{array}{cc|c} sF_3 + H_3 & ** & * \\ * & * & * \end{array} \right)$$

by row transformations, so that the lower part does not contain s with nonsingular constant matrix F_3 and constant matrix H_3.

As a result, we obtain another matrix pencil $\tilde{A}(s)$ satisfying the following conditions.

- It holds that

$$\tilde{A}(s)[R_1 \cup R_0, C_1 \cup C_0] = \left(\begin{array}{c|ccc|c} * & sF_3 + H_3 & ** & & * \\ * & * & * & & * \\ \hline O & * & * & & H_0 \end{array}\right), \tag{13}$$

where the first two row sets correspond to R_1, the last row set corresponds to R_0, the first column set corresponds to C_1, and the last three column sets correspond to C_0.

- The other entries coincide with the corresponding entries of $A(s)$.

Let us denote the first row set of (13) by S. We construct $(\tilde{p}, \tilde{q})$ from (p, q) by

$$\begin{aligned} \tilde{p}_i &= p_i - 1 \quad (i \in R \setminus (R_0 \cup S)), & \tilde{p}_i &= p_i \quad (i \in R_0 \cup S), \\ \tilde{q}_j &= q_j - 1 \quad (j \in C \setminus C_0), & \tilde{q}_j &= q_j = 0 \quad (j \in C_0). \end{aligned}$$

The following lemma ensures that $(\tilde{p}, \tilde{q})$ is a feasible solution of $\mathrm{D}(\tilde{A})$.

Lemma 3.1. Let (p, q) be a feasible solution of $\mathrm{D}(A)$ satisfying (6). Then $(\tilde{p}, \tilde{q})$ is a feasible solution of $\mathrm{D}(\tilde{A})$ satisfying (6).

Proof. By the construction rule of $\tilde{A}(s)$, we have $p_i - q_j \geq \tilde{c}_{ij}$. If $\tilde{p}_i - \tilde{q}_j \geq p_i - q_j$ holds, then $\tilde{p}_i - \tilde{q}_j \geq p_i - q_j \geq \tilde{c}_{ij}$ also holds.

Consider the case with $\tilde{p}_i - \tilde{q}_j < p_i - q_j$. This implies that $i \in R \setminus (R_0 \cup S)$ and $j \in C_0$. Then we have $\tilde{p}_i - \tilde{q}_j = p_i - 1$. If $i \notin R_1$ holds, it follows from $p_i \geq 2$ that $\tilde{p}_i - \tilde{q}_j = p_i - 1 \geq 1 \geq \tilde{c}_{ij}$. Next, suppose $i \in R_1 \setminus S$. Then we have $p_i = 1$ and $\tilde{c}_{ij} = 0$ for $(i, j) \in E(\tilde{A})$ by (13). Hence $\tilde{p}_i - \tilde{q}_j = p_i - 1 = 0 \geq \tilde{c}_{ij}$ holds. Moreover, $(\tilde{p}, \tilde{q})$ satisfies (6) by the construction rule. □

The following lemma shows that the values of the right-hand sides in (9) and (10) decrease or remain the same when we update (p, q) to $(\tilde{p}, \tilde{q})$.

Lemma 3.2. Let (p, q) be a feasible solution of $\mathrm{D}(A)$ satisfying (6). The dual solution $(\tilde{p}, \tilde{q})$ obtained by the above procedure satisfies

$$\Delta_n(p, q) \geq \Delta_n(\tilde{p}, \tilde{q}), \quad \Delta_{n-1}(p, q) \geq \Delta_{n-1}(\tilde{p}, \tilde{q}).$$

Proof. By the definition of $\tilde{p}_i$ and $\tilde{q}_j$, we have

$$\sum_{i \in R} \tilde{p}_i - \sum_{j \in C} \tilde{q}_j = \sum_{i \in R} p_i - |R \setminus (R_0 \cup S)| - \sum_{j \in C} q_j + |C \setminus C_0|.$$

Since F_3 and H_0 in (13) are nonsingular, $\tilde{A}[R_0 \cup S, C_0]$ is of full-row rank. Hence we have $|R_0 \cup S| \leq |C_0|$, which implies that

$$|R \setminus (R_0 \cup S)| \geq |C \setminus C_0|. \tag{14}$$

Thus the first inequality holds.

By the definition of $\tilde{p}$, the value of $\min_{i \in R} \tilde{p}_i$ is equal to $\min_{i \in R} p_i$ or $\min_{i \in R} p_i - 1$. Since $\sum_{i \in R} \tilde{p}_i = \sum_{i \in R} p_i - |R \setminus (R_0 \cup S)|$ holds, we have

$$\sum_{i \in R} \tilde{p}_i - \min_{i \in R} \tilde{p}_i \leq \sum_{i \in R} \tilde{p}_i - \min_{i \in R} p_i + 1 = \sum_{i \in R} p_i - \min_{i \in R} p_i - |R \setminus (R_0 \cup S)| + 1.$$

Now $C \neq C_0$ holds, because the condition in Step 2 is not fulfilled. Hence

$$\sum_{j \in C} \tilde{q}_j - \max_{j \in C} \tilde{q}_j = \sum_{j \in C} q_j - \max_{j \in C} q_j - (|C \setminus C_0| - 1)$$

follows. Thus we obtain

$$\begin{aligned}\Delta_{n-1}(\tilde{p}, \tilde{q}) &\leq \sum_{i \in R} p_i - \min_{i \in R} p_i - |R \setminus (R_0 \cup S)| - \sum_{j \in C} q_j + \max_{j \in C} q_j + |C \setminus C_0| \\ &\leq \sum_{i \in R} p_i - \min_{i \in R} p_i - \sum_{j \in C} q_j + \max_{j \in C} q_j \\ &= \Delta_{n-1}(p, q),\end{aligned}$$

where the second inequality is due to (14). □

By executing Steps 1–3, we obtain a matrix pencil $A(s)$ and its feasible solution (p, q) with the following property.

Lemma 3.3. At the end of Phase 1, we obtain (p, q) such that $p_i \in \{0, 1\}$ for every $i \in R$ and $q_j = 0$ for every $j \in C$. Moreover, the number of iterations in Phase 1 is at most n.

Proof. Step 2 ensures that $q_j = 0$ for every $j \in C$. Since $c_{ij} = 0$ or 1, this implies $p_i \in \{0, 1\}$ for each $i \in R$. At each iteration, $\max_{j \in C} q_j$ decreases by one. Lemma 2.1 ensures that $\max_{j \in C} q_j \leq n$ holds for an initial solution (p, q), which indicates that the number of iterations is at most n. □

Lemma 3.3 leads to the following corollary.

Corollary 3.4. At the end of Phase 1, we have $\hat{\nu}(p, q) \leq 1$.

Proof. By Lemma 3.3, $p_i \in \{0, 1\}$ holds for every $i \in R$ and $q_j = 0$ holds for every $j \in C$. Let m denote the number of rows with $p_i = 1$. Then we have

$$\Delta_n(p, q) = m, \quad \Delta_{n-1}(p, q) = \begin{cases} m & (m < n), \\ m - 1 & (m = n). \end{cases} \tag{15}$$

Hence it holds that

$$\hat{\nu}(p, q) = \Delta_{n-1}(p, q) - \Delta_n(p, q) + 1 = \begin{cases} 1 & (m < n), \\ 0 & (m = n). \end{cases}$$

□

3.3 Test for Tightness in Step 4

In this section, we present how to check if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ hold in Step 4.

Suppose that we have a feasible solution (p, q) of $\mathrm{D}(A)$ such that $p_i \in \{0, 1\}$ for every $i \in R$ and $q_j = 0$ for every $j \in C$. The tight coefficient matrix of $A(s)$ is defined to be the constant matrix $A^\# = (A^\#_{ij})$ with $A^\#_{ij}$ being the coefficient of $s^{p_i - q_j}$ in $A_{ij}(s)$. The following lemma enables us to check $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ efficiently.

Lemma 3.5. The tight coefficient matrix $A^\#$ is nonsingular if and only if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p,q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p,q)$ hold.

Proof. Note that $\det A(s) = s^{\Delta_n(p,q)}\{\det A^\# + \mathrm{o}(1)\}$ holds. Therefore, if $\delta_n(A) = \Delta_n(p,q)$, then $A^\#$ must be nonsingular. Conversely, if $A^\#$ is nonsingular, then $\delta_n(A) = \Delta_n(p,q)$, which together with (9) implies $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p,q)$. The nonsingularity of $A^\#$ further implies that there exists a nonsingular submatrix $A^\#[I,J]$ such that $|I| = |J| = n-1$ and $I \supseteq R^*$, where $R^* = \{i \in R \mid p_i > \min_{\ell \in R} p_\ell\}$. Since $\det A(s)[I,J] = s^{\Delta_{n-1}(p,q)}\{\det A^\#[I,J] + \mathrm{o}(1)\}$, we have $\delta_{n-1}(A) \geq \Delta_{n-1}(p,q)$, which together with (10) implies $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p,q)$. □

By Lemma 3.5, we can perform Step 4 by checking the nonsingularity of $A^\#$.

3.4 Unimodular Transformations in Step 5

Let $A(s)$ be a matrix pencil in Step 5. The algorithm has detected that the condition in Step 4 is not fulfilled, i.e., the tight coefficient matrix $A^\#$ is singular. Hence there exists a nonzero row vector $\boldsymbol{u} = (u_i \mid i \in R)$ such that

$$\boldsymbol{u}A^\# = \boldsymbol{0}.$$

By executing the Gaussian elimination on $A^\#$ with column transformations, we can find $\boldsymbol{u}$ such that $\operatorname{supp} \boldsymbol{u} := \{i \in R \mid u_i \neq 0\}$ is minimal with respect to set inclusion.

By the definition of $A^\#$, we have $A^\#[R_0, C] = A(s)[R_0, C]$. Since $A(s)$ is regular, $A^\#[R_0, C]$ is of full-row rank. This implies that there exists $l \in \operatorname{supp} \boldsymbol{u}$ with $p_l = 1$.

We now define U by

$$U_{ik} = \begin{cases} u_k/u_l & (i = l), \\ \delta_{ik} & (i \neq l), \end{cases}$$

where δ_{ik} denotes Kronecker's delta. We remark that the row set and the column set of U correspond to $R_1 \cup R_0$ and $U[R_0, R_1] = O$. We denote by $\operatorname{diag}(s;p)$ the square diagonal matrix with each (i,i) entry being s^{p_i}. Then the polynomial matrix $U(s) = \operatorname{diag}(s;p) \cdot U \cdot \operatorname{diag}(s;-p)$ is unimodular.

Since $A(s)$ can be expressed as

$$A(s) = \operatorname{diag}(s;p) \cdot \left(A^\# + \frac{1}{s} \begin{pmatrix} A(0)[R_1, C] \\ O \end{pmatrix} \right),$$

it holds that

$$\begin{aligned} U(s)A(s) &= \operatorname{diag}(s;p) \cdot U \cdot \left(A^\# + \frac{1}{s} \begin{pmatrix} * \\ O \end{pmatrix} \right) = \operatorname{diag}(s;p) \cdot \left(UA^\# + \frac{1}{s} \begin{pmatrix} * & * \\ O & * \end{pmatrix} \begin{pmatrix} * \\ O \end{pmatrix} \right) \\ &= \operatorname{diag}(s;p) \cdot \left(UA^\# + \frac{1}{s} \begin{pmatrix} * \\ O \end{pmatrix} \right) = \operatorname{diag}(s;p) \cdot UA^\# + \begin{pmatrix} * \\ O \end{pmatrix}, \end{aligned}$$

where $*$ denotes a constant matrix. Hence $U(s)A(s)$ remains to be a matrix pencil. Since the lth row vector of $UA^\#$ is zero, $U(s)A(s)$ does not contain s in the lth row. Hence we can decrease $p_l = 1$ by one. By setting

$$\hat{A}(s) := U(s)A(s), \quad \hat{p}_i := \begin{cases} 0 & (i = l), \\ p_i & (i \neq l), \end{cases} \quad \hat{q} := q,$$

we obtain another matrix pencil $\hat{A}(s)$ and its feasible solution $(\hat{p}, \hat{q})$.

Lemma 3.6. The number of iterations in Phase 2 is at most n.

Proof. At each iteration, the number of rows with $p_i = 0$ increases by one. □

At the end of the index reduction algorithm, we obtain a matrix pencil with index at most one.

Theorem 3.7. The algorithm finds a matrix pencil with the Kronecker index at most one in $\mathrm{O}(n^4)$ time.

Proof. When the algorithm terminates, we obtain $\bar{A}(s)$ and its optimal solution $(\bar{p}, \bar{q})$ satisfying (11) and (12) by Lemmas 3.3 and 3.5. Let m denote the number of rows with $p_i = 1$. Then we have (15) for $(\bar{p}, \bar{q})$. Hence the Kronecker index $\nu(\bar{A})$ is given by

$$\nu(\bar{A}) = \delta_{n-1}(\bar{A}) - \delta_n(\bar{A}) + 1 = \Delta_{n-1}(\bar{p}, \bar{q}) - \Delta_n(\bar{p}, \bar{q}) + 1 = \begin{cases} 1 & (m < n), \\ 0 & (m = n). \end{cases}$$

Thus the index of the obtained matrix pencil is at most one.

In Step 1, we solve a maximum-weighted perfect matching problem. This can be performed in $\mathrm{O}(n^3)$ time by the Hungarian method [10, 13, 20]. Steps 3 and 5 require the Gaussian elimination, which costs $\mathrm{O}(n^3)$ time at each iteration. Since the number of iterations of Steps 3 and 5 is $\mathrm{O}(n)$ by Lemmas 3.3 and 3.6, the total time complexity is $\mathrm{O}(n^4)$. □

4 Examples

We give two examples below. The first one is a famous example for which Pantelides' algorithm does not work:

$$\begin{aligned} z_1 - \dot{z}_1 + 2z_2 + 3z_3 &= 0, \\ z_1 + z_2 + z_3 + 1 &= 0, \\ 2z_1 + z_2 + z_3 &= 0. \end{aligned}$$

The corresponding matrix pencil $A(s)$ is expressed as

$$A(s) = \begin{pmatrix} -s+1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}.$$

By $\delta_2(A) = 1$ and $\delta_3(A) = 0$, the index $\nu(A)$ is equal to 2. However, when we apply Pantelides' algorithm [16] to $A(s)$, the algorithm terminates without detecting equations to be differentiated. Pantelides' algorithm is adopted in the MATLAB function called `reduceDAEIndex`. In fact, this function does not work for the DAE.

Let us apply our algorithm to $A(s)$. In Step 1, we find an optimal solution $p = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$ of $\mathrm{D}(A)$. In Step 3, we obtain another solution $p = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ without changing $A(s)$. Then we go to Step 4 by $q = \mathbf{0}$. The tight coefficient matrix $A^{\#} = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}$ is singular. In Step 5, we have $\boldsymbol{u} = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix}$ and $U(s) = \begin{pmatrix} 1 & -s & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. The matrix pencil $A(s)$ is transformed into $U(s)A(s) = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}$ with $p = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$. Then we obtain $\nu(UA) = 1$.

Next, consider another matrix pencil

$$A(s) = \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & s \end{pmatrix}.$$

It follows from $\delta_3(A) = 2$ and $\delta_4(A) = 0$ that $\nu(A) = 3$. We apply the algorithm described in Section 3 to $A(s)$.

In Step 1, we find an optimal solution $p = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$ and $q = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$ of $\mathrm{D}(A)$. Then we go to Step 3 by $q \neq \mathbf{0}$. In Step 3, we delete s in the last row by row transformations and obtain a feasible dual solution $p' = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$ and $q' = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$ as follows:

$$A(s) = \begin{matrix} \\ R_1 \\ \\ \end{matrix} \overset{\begin{matrix} C_1 & C_0 \end{matrix}}{\begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & s \end{pmatrix}} \longrightarrow A'(s) = U^{\circ}(s)A(s) = \begin{matrix} R_1 \\ \\ R_0 \\ \end{matrix} \overset{C_0}{\begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}},$$

where $U^{\circ}(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$. We return to Step 2 and then go to Step 4 by $q' \neq \mathbf{0}$. The tight coefficient matrix $A^{\#} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ is singular in Step 4, and we have $\boldsymbol{u}' = \begin{pmatrix} 0 & 1 & -1 & 1 \end{pmatrix}$

and $U'(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -s & s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ in Step 5. The matrix pencil $A'(s)$ is transformed into

$$A''(s) = U'(s)A'(s) = \begin{array}{c} R_1 \\ \\ R_0 \\ \\ \end{array} \overset{\displaystyle C_0}{\begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}}$$

with $p'' = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$ and $q'' = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$.

Returning to Step 4, the tight coefficient matrix $A^{\#} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ is also singular. In Step 5, we have $\boldsymbol{u}'' = \begin{pmatrix} 1 & -1 & 0 & 0 \end{pmatrix}$ and $U''(s) = \begin{pmatrix} 1 & -s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$. The matrix pencil $A''(s)$ is transformed into

$$\bar{A}(s) = U''(s)A''(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

with $\bar{p} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$ and $\bar{q} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$. Returning to Step 4, the tight coefficient matrix $A^{\#} = \bar{A}(s)$ is nonsingular and hence we terminate the algorithm.

As a result, we obtain a unimodular matrix $U(s)$ and a matrix pencil $\bar{A}(s)$ with $\nu(\bar{A}) = 1$ expressed as

$$U(s) = U''(s)U'(s)U^{\circ}(s) = \begin{pmatrix} 1 & s^2 - s & s^2 & -s^2 \\ 0 & -s+1 & -s & s \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}, \quad \bar{A}(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

5 Conclusion

We have presented a new index reduction algorithm of matrix pencils which makes use of unimodular transformations. The algorithm is based on the framework of combinatorial relaxation, which combines graph-algorithmic techniques and matrix computation. Our algorithm can be used as an index reduction method for linear DAEs. It works correctly for any linear DAEs including those for which Pantelides' algorithm is known to fail. An extension of our algorithm to index reduction of nonlinear DAEs is left for future investigation.

References

[1] U. M. Ascher and L. R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, 1998.

[2] T. Beelen and P. Van Dooren, *An improved algorithm for the computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 105 (1988), pp. 9–65.

[3] K. E. Brenan, S. L. Campbell and L. R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, SIAM, Philadelphia, 2nd edition, 1996.

[4] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil A-λB: Robust software with error bounds and applications. Part I: Theory and algorithms*, ACM Trans. Math. Softw., 19 (1993), pp. 160–174.

[5] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil A-λB: Robust software with error bounds and applications. Part II: Software and applications*, ACM Trans. Math. Softw., 19 (1993), pp. 175–201.

[6] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, Springer-Verlag, Berlin, 2nd edition, 1996.

[7] S. Iwata, *Computing the maximum degree of minors in matrix pencils via combinatorial relaxation*, Algorithmica, 36 (2003), pp. 331–341.

[8] S. Iwata, K. Murota, and I. Sakuta, *Primal-dual combinatorial relaxation algorithms for the maximum degree of subdeterminants*, SIAM J. Sci. Comput., 17 (1996), pp. 993–1012.

[9] B. Kågström, *RGSVD—an algorithm for computing the Kronecker structure and reducing subspaces of singular $A - \lambda B$ pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.

[10] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly, 2 (1955), pp. 83–97.

[11] P. Kunkel and V. Mehrmann, Differential-Algebraic Equations: Analysis and Numerical Solutions, European Mathematical Society, Zürich, 2006.

[12] S. E. Mattsson and G. Söderlind, *Index reduction in differential-algebraic equations using dummy derivatives*, SIAM J. Sci. Comput., 14 (1993), pp. 677–692.

[13] J. Munkres *Algorithms for the assignment and transportation problems*, J. SIAM, 5 (1957), pp. 32–38.

[14] K. Murota, *Combinatorial relaxation algorithm for the maximum degree of subdeterminants: Computing Smith-McMillan form at infinity and structural indices in Kronecker form*, Appl. Algebra Engrg. Comm. Comput., 6 (1995), pp. 251–273.

[15] K. Murota, Matrices and Matroids for Systems Analysis, Springer-Verlag, Berlin, 2000.

[16] C. C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 213–231.

[17] J. D. Pryce, *A Simple Structural Analysis Method for DAEs*, BIT, 41 (2001), pp. 364–394.

[18] R. Riaza, Differential-Algebraic Systems: Analytical Aspects and Circuit Applications, World Scientific Publishing Company, Singapore, 2008.

[19] S. Sato, *Combinatorial relaxation algorithm for the entire sequence of the maximum degree of minors*, Algorithmica, 77 (2017), pp. 815–835.

[20] N. Tomizawa, *On some techniques useful for solution of transportation network problems*, Networks, 1 (1971), pp. 173–194.

[21] P. Van Dooren, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–140.