# An Improved Algorithm for the Computation of Kronecker's Canonical Form of a Singular Pencil

Th. Beelen
*PICOS Glass*
*Philips Eindhoven*

and

P. Van Dooren
*Philips Research Laboratory*
*Av. Van Becelaere 2, Box 8*
*B-1170 Brussels, Belgium*

Submitted by Hans Schneider

ABSTRACT

We give an $O(m^2n)$ algorithm for computing the Kronecker structure of an arbitrary $m \times n$ pencil $\lambda E - A$. The algorithm is shown to be numerically stable, because only unitary transformations are used. The improved speed over earlier unitary methods is due to the efficient use of condensed forms which are maintained throughout the recursions of the algorithm.

## 1. INTRODUCTION

The problem of determining the Kronecker canonical form (KCF) of a singular pencil has received considerable attention over the last few years [7, 23, 18, 24, 9, 12, 8]. Part of this is certainly due to the relevance of the KCF in a number of applications found in the area of systems and control theory [10, 19, 16]. In these various papers different aspects of the computation of the KCF have been looked at, such as the computational complexity and the numerical stability of some algorithm, or the sensitivity of the KCF.

In this paper we mainly focus on the complexity of the problem. We present a new algorithm which is shown to require only $O(m^2n)$ operations



© Elsevier Science Publishing Co., Inc., 1988
52 Vanderbilt Ave., New York, NY 10017 0024-3795/88/$3.50

for computing the so-called generalized Schur decomposition of an arbitrary $m \times n$ pencil $\lambda E - A$. The transformations required for this decomposition are unitary. This fact is also shown to ensure the backward stability of the algorithm. Moreover, the KCF is easily derived from this Schur form without any additional computations [18]. We believe that this $O(m^2 n)$ algorithm is an important improvement, since earlier methods based on unitary transformations only [18, 12, 8] have a complexity which is $O(n^4)$ in the worst case.

In Section 2 we develop some preliminary material needed in the rest of the paper. In Section 3 we then give the new algorithm, which in fact consists of two subalgorithms. The first one separates the infinite elementary divisors and right Kronecker indices from the finite elementary divisors and left Kronecker indices. The second one performs the further separation between the right Kronecker indices and the infinite elementary divisors. In this section we also give an operation count of these two algorithms and discuss their numerical stability. In Section 4 we then finally mention a few applications of this new algorithm in the area of systems and control theory.

## 2. PRELIMINARIES

Throughout this report we use uppercase for matrices and lowercase for vectors and scalars. The identity matrix is denoted by $I$. All matrices, vectors and functions considered are defined over $\mathbb{C}$. By $A^T$ we denote the transpose of $A$. The pertranspose of $A$ (reflection in the second diagonal) is indicated by $A^P$. By $\operatorname{diag}\{M_1, \dots, M_k\}$ we denote a block diagonal matrix with diagonal blocks $M_i$ that are not necessarily square.

Consider the pencil $\lambda E - A$ with $A$ and $E$ arbitrary constant matrices of equal dimensions. The pencil is said to be *regular* if $\lambda E - A$ is square *and* $\det(\lambda E - A) \not\equiv 0$. Otherwise, it is called *singular*. Two pencils $\lambda E_1 - A_1$ and $\lambda E_2 - A_2$ of dimensions $m \times n$ are termed *strictly equivalent* when there exist constant (independent of $\lambda$) invertible matrices $P$ and $Q$ of orders $m$ and $n$, respectively, such that

$$P(\lambda E_1 - A_1)Q = \lambda E_2 - A_2. \tag{2.1}$$

We denote this equivalence relation by $\sim$. When $P$ and $Q$ are, moreover, unitary, these pencils are said to be *unitarily equivalent*, which is denoted by $\overset{\mathrm{U}}{\sim}$.

Kronecker (see [5]) has shown that any pencil is strictly equivalent to a canonical block diagonal form

$$\lambda E - A \sim \operatorname{diag}\left\{ L_{\epsilon_1}, \dots, L_{\epsilon_p}, L^T_{\eta_1}, \dots, L^T_{\eta_q}, \lambda N - I, \lambda I - J \right\}, \tag{2.2}$$

where

(1) $L_\epsilon$ is the $\epsilon \times (\epsilon+1)$ bidiagonal pencil

$$\begin{bmatrix} \lambda & -1 & & \\ & \ddots & \ddots & \\ & & \lambda & -1 \end{bmatrix},$$

(2) $L_\eta^T$ is the $(\eta+1)\times\eta$ bidiagonal pencil

$$\begin{bmatrix} \lambda & & \\ -1 & \ddots & \\ & \ddots & \lambda \\ & & -1 \end{bmatrix},$$

(3) $N$ is a nilpotent Jordan matrix, and
(4) $J$ is in Jordan canonical form.

The matrix $\lambda I - J$ contains the *finite elementary divisors*, and $\lambda N - I$ the *infinite elementary divisors* of $\lambda E - A$. The blocks $L_{\epsilon_i}$ and $L_{\eta_j}^T$ contain the *singularity* of the pencil. In fact, for $L_\epsilon$ there exists a polynomial column vector such that

$$L_\epsilon [1 \quad \lambda \quad \cdots \quad \lambda^\epsilon]^T = [0 \quad \cdots \quad 0]^T, \tag{2.3}$$

while for $L_\eta^T$ there exists a polynomial row vector such that

$$[1 \quad \lambda \quad \cdots \quad \lambda^\eta] L_\eta^T = [0 \quad \cdots \quad 0]. \tag{2.4}$$

The sizes of these blocks characterize them completely. Therefore they are given special names [5]. The $\epsilon_i$ and $\eta_j$ are called Kronecker *column* and *row* indices, respectively ($\epsilon_i$ and $\eta_j$ are also called the *right* and *left* Kronecker indices.) Notice that the indices may be zero, corresponding to constant (degree zero) nulling vectors. The pencil $\lambda N - I$ is completely determined by the degrees $\delta_i$ of the infinite elementary divisors, and $\lambda I - J$ by the finite elementary divisors $(\lambda - \alpha_i)^j$.

In [18] it is explained that in order to compute the Kronecker canonical form, it is recommended from a numerical point of view to compute instead the following quasitriangular form, also called the *generalized Schur form*:

$$\lambda E - A^{\mathrm{U}} \left[\begin{array}{c|c|c|c} \lambda E_\epsilon - A_\epsilon & X & X & X \\ \hline & \lambda E_\infty - A_\infty & X & X \\ \hline & & \lambda E_f - A_f & X \\ \hline & & & \lambda E_\eta - A_\eta \end{array}\right], \tag{2.5}$$

which can be obtained under unitary transformations, and where

(1) $\lambda E_f - A_f$ is a square regular pencil containing the finite elementary divisors of $\lambda E - A$;

(2) $\lambda E_\infty - A_\infty$ is a square regular pencil containing the infinite elementary divisors of $\lambda E - A$;

(3) $\lambda E_\eta - A_\eta$ and $\lambda E_\epsilon - A_\epsilon$ are singular pencils containing the Kronecker row and column structure, respectively.

Moreover, the finer details of each diagonal block (i.e. the so-called staircase structure) in (2.5) completely reveal the structural elements of the Kronecker canonical form [18].

In order to obtain this decomposition we repeatedly use unitary row transformations to reduce an arbitrary $m \times n$ matrix $A$ to the form

$$PA = \begin{bmatrix} A_r \\ 0 \end{bmatrix} \begin{matrix} \}\rho \\ \phantom{0} \end{matrix}, \tag{2.6}$$

where $A_r$ has $\rho$ linearly independent rows (thus $\rho$ is the rank of $A$). Such a transformation is called a *row compression* of matrix $A$. Analogously we use the name *column compression* for the unitary column transformation

$$AQ = [0 \mid \underbrace{A_c}_{\rho}], \tag{2.7}$$

where the columns of $A_c$ are linearly independent. The resulting matrices $A_r$ and $A_c$ are said to have full row rank and full column rank, respectively.

There are a number of methods available for computing these expressions in a numerically reliable way [6]. Among them we mention the singular value decomposition and $QR$ factorization with pivoting [6]. In this paper we mainly use methods quite similar to the latter, which we briefly recall here. Let $A$ be an $m \times n$ matrix of rank $\rho$. Then there exist a *unitary transforma-*

*tion* $P$ and a *permutation* $Q$ such that

$$\hat{A} \doteq PAQ = \begin{array}{cc} \left[\begin{array}{c|c} \begin{matrix} \mathrm{x} & x & \cdots & x \\ & \ddots & & \vdots \\ & & \mathrm{x} & x \\ & & & \mathrm{x} \end{matrix} & \begin{matrix} x & \cdots & x \\ \vdots & & \vdots \\ x & & x \\ x & \cdots & x \end{matrix} \\ \hline 0 & 0 \end{array}\right] & \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}\rho \\ \}\, m-\rho \end{matrix} \\ \begin{matrix} \underbrace{\hphantom{xxxxxxxxxx}}_{\rho} & \underbrace{\hphantom{xxxxxxx}}_{n-\rho} \end{matrix} & \end{array} \tag{2.8}$$

where the x's are nonzero and hence $\rho = \operatorname{rank}(A)$. From this, one easily obtains (2.6) by postmultiplying $\hat{A}$ by $Q$. A dual decomposition with $P$ a permutation and $Q$ a unitary transformation also leads to the result requested in (2.7).

If one wants to avoid the (intermediate) permutation altogether, then one can make use of an echelon form of $A$, which we give now below for the case of column transformations. There always exists a unitary transformation $Q$ such that

$$\hat{A} \doteq AQ = \begin{array}{cc} \left[\begin{array}{c|c} \begin{matrix} \\ \\ \\ 0 \\ \\ \\ \\ \end{matrix} & \begin{matrix} x & x & \cdots & x \\ \vdots & & & \cdot \\ \mathrm{x} & x & & \cdot \\ & \vdots & & \cdot \\ & \mathrm{x} & & \cdot \\ & & \ddots & \cdot \\ & 0 & & \mathrm{x} \end{matrix} \end{array}\right] & \begin{matrix} \\ \\ \leftarrow i_1 \\ \\ \leftarrow i_2 \\ \vdots \\ \leftarrow i_\rho \end{matrix} \\ \begin{matrix} \underbrace{\hphantom{xxxx}}_{n-\rho} & \underbrace{\hphantom{xxxxxxxxx}}_{\rho} \end{matrix} & \end{array} \tag{2.9}$$

where the x's are the last nonzero elements in each column of $\hat{A}$ and have *increasing* row indices $1 \leqslant i_1 < i_2 < \cdots < i_\rho \leqslant m$. This *column echelon form* directly yields (2.7), and there also exists a dual *row echelon form* corresponding to (2.6).

Remark 2.1. Due to roundoff in a computer, one should always expect to find full rank matrices (i.e. $\rho = \min\{m, n\}$) unless a *threshold* $\delta$ is chosen below which elements of $\hat{A}$ will be considered zero. Since with unitary decompositions of the type (2.8), (2.9) roundoff errors are of the order $\epsilon\|A\|_F$ [22] (where $\epsilon$ is the relative precision of the computer used), one must choose

$\delta > \epsilon \|A\|_F$. When elements below the threshold $\delta$ are set to zero, one has then that (2.8) and (2.9) are in fact decompositions of a "nearby" matrix $A + E$ with $\|E\|_F < \delta$ [22]. Moreover, the x's in (2.8) and (2.9) are then all larger in magnitude than $\delta$. It should also be noted here that the above decompositions do *not* guarantee that one finds the smallest possible rank of $A + E$ for some $\delta$-perturbation $E$ (the so-called $\delta$-rank), but counterexamples [13] are not likely to occur frequently.

REMARK 2.2. If one uses Givens rotations to perform the $QR$ decomposition (2.8), then one needs

$$\rho m - \frac{\rho(\rho+1)}{2} \text{ rotations} \tag{2.10}$$

and

$$\rho n - \frac{\rho(\rho+1)}{2} \text{ rotations} \tag{2.11}$$

for the dual decomposition. The number of "flops" involved (one flop is the work needed for one addition and one multiplication) is roughly equal to

$$4\rho mn - 2\rho^2\left(m + n - \tfrac{2}{3}\rho\right) \text{ flops} \tag{2.12}$$

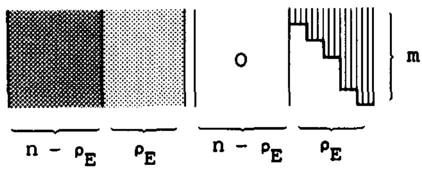
for both (2.8) and the dual decomposition (we have neglected lower order terms here). Moreover, the above numbers of rotations and flops are also the *maximum* numbers required by the corresponding echelon forms.

## 3. A NEW ALGORITHM FOR THE GENERALIZED SCHUR FORM

### 3.1. *The Basic Step*

In the following subsections we propose an algorithm which is related to the "staircase algorithm" described in [18], in the sense that it constructs the same decomposition (2.5) and also uses unitary transformations only. The most important difference is that a preliminary transformation of $E$ to a "condensed form" (namely the echelon form) is performed and that this form is then exploited in the subsequent staircase reduction. It turns out that keeping $E$ in condensed form can be done very efficiently. Roughly speaking, our new algorithm requires $O(n^3)$ operations, in contrast to $O(n^4)$ in the worst case for the staircase algorithm of [18] (here $n$ is supposed to be the maximum dimension of $\lambda E - A$). The reduction of the complexity from $O(n^4)$ to $O(n^3)$ is completely due to the following decomposition of a pair of $m \times n$ matrices $(A, E)$, where $E$ has rank $\rho_E$ and is in column echelon form:

$$[A \,\|\, E] \doteq [A_1 \mid A_2 \,\|\, 0 \mid E_2] =$$

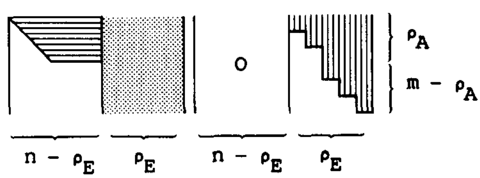



(3.1.1)

Notice here the special notation $[A \| E]$ we introduce for a pencil $\lambda E - A$ (the constant term $A$ is always on the left) in order to display better the fine structure of both $A$ and $E$ in the development of our algorithms. The factors appearing left and right of $[A \| E]$ are thus applied to both $A$ and $E$. We now show that there exist unitary transformations $P$ and $Q_2$ and a permutation $Q_1$ such that

$$P[A_1 \mid A_2 \,\|\, 0 \mid E_2]\operatorname{diag}\{Q_1, Q_2\}$$

$$=$$




(3.1.2)

i.e., where $A_1$ is row compressed and $E_2$ is maintained in column echelon form (though not necessarily with the same pivot indices). For this decomposition we use a product of Givens transformations $P_{i-1,i}$ operating only on two *successive* rows $i-1$ and $i$ in order to form the transformation matrix $P$. This is in fact classical when performing the $QR$ decomposition with pivoting (accumulated in $Q_1$)—in this case of the matrix $A_1$—via Givens transformations [6]. Each time such a Givens rotation affects the echelon form of $E_2$, this is restored via an appropriate column transformation which also turns out to be a Givens rotation. Together, these then constitute the column transformation $Q_2$. The details are now discussed by looking at the different cases that may occur when a Givens rotation $P_{i-1,i}$ operates on rows $e_{i-1}$ and $e_i$ of a matrix $E$ in echelon form. Let $G_{i-1,i}$ be the $2\times 2$ Givens rotation embedded in $P_{i-1,i}$. We can distinguish four different cases of pairs $\left[\frac{e_{i-1}}{e_i}\right]$ as shown in Figure 1 (the "nonzero" elements are marked by x).

We then have

$$G_{i-1,i}\left[\frac{e_{i-1}}{e_i}\right]=$$

$$G_{i-1,i}\left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\ 0 & \cdots & 0 & 0 & \mathrm{x} & \cdots & x \end{array}\right] = \left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & x & x & \cdots & x \\ 0 & \cdots & 0 & x & x & \cdots & x \end{array}\right] \quad (1),$$

$$G_{i-1,i}\left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\ 0 & \cdots & 0 & 0 & x & \cdots & x \end{array}\right] = \begin{cases} \left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\ 0 & \cdots & 0 & 0 & x & \cdots & x \end{array}\right] & (2a) \\ \text{or} & \\ \left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\ 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \end{array}\right] & (2b) \end{cases},$$

$$G_{i-1,i}\left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & x & x & \cdots & x \\ 0 & \cdots & 0 & x & x & \cdots & x \end{array}\right] = \left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & x & x & \cdots & x \\ 0 & \cdots & 0 & x & x & \cdots & x \end{array}\right] \quad (3),$$

$$G_{i-1,i}\left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & x & x & \cdots & x \\ 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \end{array}\right] = \begin{cases} \left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\ 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \end{array}\right] & (4a) \\ \text{or} & \\ \left[\begin{array}{ccc|cccc} 0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\ 0 & \cdots & 0 & 0 & x & \cdots & x \end{array}\right] & (4b) \end{cases}. \quad (3.1.3)$$

$$
\begin{array}{llllllll}
 & \vdots & & & & & & \\
(1) & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & \cdots & \mathrm{x} \\
(2) & & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & & \\
(3) & & & \mathrm{x} & \mathrm{x} & \mathrm{x} & & \\
(3) & & & \mathrm{x} & \mathrm{x} & \mathrm{x} & & \\
(4) & & & \mathrm{x} & \mathrm{x} & \mathrm{x} & & \\
(1) & & & \mathrm{x} & \mathrm{x} & \mathrm{x} & & \\
 & & & & \mathrm{x} & \mathrm{x} & \cdots & \mathrm{x} \\
 & & & & & \vdots & &
\end{array}
$$

FIG. 1. The four different boundary cases for $\left[\frac{e_{i-1}}{e_i}\right]$.

We see that in each of the four cases the rows $e_{i-1}$ and $e_i$ are transformed to a pair of rows also belonging to an echelon form. Moreover, only in the cases (1), (2b), and (4b) do the transformed rows have a different from than the original pair. It is easily seen that only in case (1) is the (complete) transformed matrix not guaranteed to be in echelon form any longer. However, in this case the "leading" nonzero $2\times 2$ matrix of $\left[\frac{e_{i-1}}{e_i}\right]$ has rank 2. Therefore, we can find a Givens transformation $Q_{j,j-1}$ affecting columns $j-1$ and $j$ (where $j$ is determined by the "boundary" of the echelon form) such that

$$
G_{i-1,i}\left[\frac{e_{i-1}}{e_i}\right]Q_{j,j-1}=
\begin{array}{c}
\begin{array}{ccccccc} & & & j-1 & j & & \\ & & & \downarrow & \downarrow & & \end{array}\\
\left[\begin{array}{ccc|cccc}
0 & \cdots & 0 & \mathrm{x} & x & \cdots & x \\
0 & \cdots & 0 & 0 & \mathrm{x} & \cdots & x
\end{array}\right].
\end{array}
\qquad (3.1.4)
$$

Thus, in case (1) we recover an echelon form by an extra column transformation $Q_{j,j-1}$. In all other cases we can take $Q_{j,j-1}=I_n$.

REMARK 3.1.1. The cases (a) and (b) in (2) and (4) are distinguished via a test on the bottom element of the relevant column: if this element is smaller in magnitude than a threshold $\delta$, then it is set equal to 0. The same comments hold here as in Remark 2.1.

Before giving the complete algorithm we first illustrate in Figure 2 these different cases in the elimination of the $k$th column of $A_1$ (stage $k$). At the beginning of stage $k$ we have a $(k-1)\times(k-1)$ triangular submatrix in $A_1$. In substeps (1) and (2) zeros (denoted by [0]) are created in $A_1$ by row

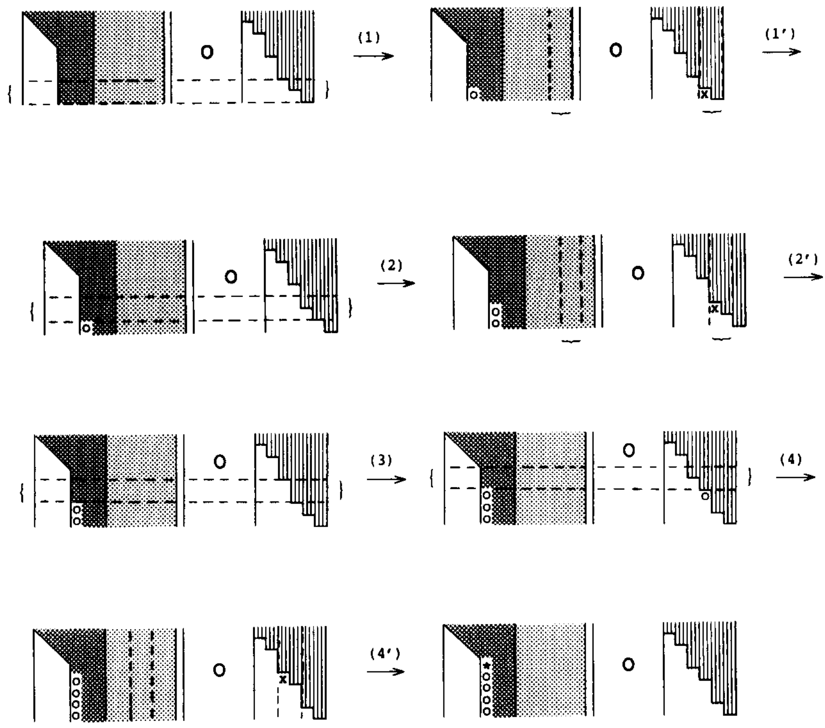


FIG. 2. Stage $k$ in the reduction process.

rotations, and the nonzero elements just introduced in $E_2$ (denoted by $[\times]$) are annihilated subsequently in steps (1′) and (2′) by column rotations. In step (3) a third zero is created in $A_1$. Hereby we have assumed that this rotation also introduces a zero in $E_2$ [see case (4b) in (3.1.3)]. Hence no restoration is done on $E_2$. At this moment we have found a *new* echelon form of $E_2$. We proceed in step (4) with creating a zero in $A_1$ plus introducing a nonzero element in $E_2$. After restoration of $E_1$ in step (4′) we finally obtain a $k \times k$ triangular submatrix in $A_1$ while $E_2$ is in echelon form (different from the original one). Repeating such a procedure for all columns of $A_1$, one finds the rank $\rho_A$ of this matrix and the decomposition (3.1.2). This is now summarized in the following algorithm written in Algol-like notation.

ALGORITHM 3.1.1 (Row compression of $A_1$ while keeping $E_2$ in echelon form).

**comment** Initialization. $P$ and $Q_1, Q_2$ are the row and column transformations;
$k := 1$, *zerotest* := *false*; $\rho_A := 0$; $P := I_m$; $Q_1 := I_{n-\rho_E}$; $Q_2 := I_{\rho_E}$;
**comment** Iteration process;
**while** $k \leqslant \min(m, n-\rho_E)$ **and** *zerotest* = *false* **do**
**begin**
  $a_j := A_1(k\colon m; j)$ $(j = k, \ldots, n-\rho_E)$;
  Determine smallest index $l$ such that $\|a_l\| = \max\{\|a_j\| \mid j = k, \ldots, n - \rho_E\}$;
  **if** $\|a_l\| \leqslant tol$ **then begin** $A_1(k\colon m, k\colon n-\rho_E) := 0$; *zerotest* := *true* **end else**
  **begin**
    **if** $l \neq k$ **then** Interchange columns $k$ and $l$ of $A_1$ and update $Q_1$;
    **for** $i = m$ **step** $-1$ **until** $k+1$ **do**
    **begin**
      **comment** Annihilate $A_1(i, k)$ by row rotations $P^{(k)}_{i-1,i}$.
      Also apply these to $A_2$ and $E_2$;
      $A_1 := P^{(k)}_{i-1,i}A_1$; $A_2 := P^{(k)}_{i-1,i}A_2$; $E_2 := P^{(k)}_{i-1,i}E_2$; **result** $A_1(i,k) = 0$;
      Determine boundary type of transformed rows $i-1$ and $i$ of $E_2$;
      **if** type $\neq 1$ **then** $Q^{(k)}_{j(i),j(i)-1} := I_{\rho_E}$ **else**
      **begin**
        **comment** Annihilate $E_2(i, j(i)-1)$ by column rotations $Q^{(k)}_{j(i),j(i)-1}$.
        Also apply these to $A_2$;
        $E_2 := E_2Q^{(k)}_{j(i),j(i)-1}$; $A_2 := A_2Q^{(k)}_{j(i),j(i)-1}$; **result** $E_2(i, j(i)-1) = 0$;
      **end**;
      **comment** Update matrices $P$ and $Q_2$;
      $P := P^{(k)}_{i-1,i}P$; $Q_2 := Q_2Q^{(k)}_{j(i),j(i)-1}$;
    **end**;
    $\rho_A := \rho_A + 1$;
  **end**;
  $k := k+1$;
**end**;
**comment** End of Algorithm 3.1.1;

REMARK 3.1.2. If requested, one could restore the original order of the columns in $A_1$. In this case $Q_1$ in (3.1.2) equals $I_{n-\rho_E}$ and $P_1A_1$ in (3.1.2) is still row compressed but without particular zero pattern, as was commented in (2.6), (2.8). Since this is not essential in the sequel, we shall hereafter denote the column transformation by $Q$ $(\doteq \operatorname{diag}\{Q_1, Q_2\})$ and leave open whether or not $Q_1 = I_{n-\rho_E}$.

### 3.2. *The Recursive Algorithm*

Let us now present the new algorithm, which consists of several steps. The starting point is a pencil $\lambda E - A$ where $A$ and $E$ are constant matrices both of dimensions $m \times n$. Let $\rho = \operatorname{rank}(E)$. We perform the following steps.

*Step 0.* Transform $E$ to column echelon form, i.e., determine unitary matrices $P_0$ and $Q_0$ such that

$$E_1 := P_0 E Q_0 = [\,0 \mid \underbrace{E_{1,2}}_{\rho}\,] = \left[\begin{array}{c|cccc} & x & x & \cdots & x \\ & \vdots & & & \vdots \\ & x & x & \cdots & x \\ \cline{2-5} 0 & x & x & \cdots & x \\ & & x & \ddots & \vdots \\ & & & \ddots & x \\ & & & & x \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}x\\ \vdots\\ x\end{array}}\right\} m-\rho \\ \left.\vphantom{\begin{array}{c}x\\ x\\ x\\ x\end{array}}\right\} \rho \end{array} \qquad (3.2.1)$$

$$\hphantom{E_1 := P_0 E Q_0 = [\,0 \mid E_{1,2}\,] =}\;\underbrace{\hphantom{000}}_{n-\rho}\;\underbrace{\hphantom{x\;x\;\cdots\;x}}_{\rho}$$

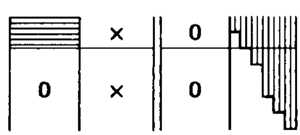
and compute $A_1 := P_0 A Q_0$; set $\rho_1 := \rho$, $\mu_1 := n - \rho_1$.

*Step 1.* Consider the pencil $\lambda E_1 - A_1$. Partition $A_1$ conformally with $E_1$, i.e.,

$$A_1 = [\,\underbrace{A_{1,1}}_{\mu_1} \mid \underbrace{A_{1,2}}_{\rho_1}\,]\}m, \qquad E_1 = [\,\underbrace{0}_{\mu_1} \mid \underbrace{E_{1,2}}_{\rho_1}\,]\}m \qquad (3.2.2)$$

The matrix $A_{1,1}$ is then compressed to full row rank while keeping $E_{1,2}$ in echelon form. This is done by applying Algorithm 3.1.1 given above. Let the accumulated row and column transformations in the reduction process of $(A_1, E_1)$ be represented by the $m \times m$ matrix $P_1$ and $n \times n$ matrix $Q_1$, respectively. Then, at the end of step 1 we have (after reusing block names)

$$P_1[A_1 \,\|\, E_1]Q_1 = P_1[A_{1,1} \mid A_{1,2} \,\|\, 0 \mid E_{1,2}]Q_1 =$$



$$= \left[\begin{array}{c|c||c|c} A_{1,1} & A_{1,2} & 0 & E_{1,2} \\ \hline 0 & A_2 & 0 & E_2 \end{array}\right] \begin{array}{l} \}\nu_1 \\ \}m_2 \end{array} \Bigg\} m \qquad (3.2.3)$$

$$\underbrace{\hphantom{A_{1,1}}}_{\mu_1}\;\underbrace{\hphantom{A_{1,2}}}_{n_2}\;\underbrace{\hphantom{A_{1,1}}}_{\mu_1}\;\underbrace{\hphantom{E_{1,2}}}_{n_2}$$

where $n_2 = \rho_1$, $n = \mu_1 + n_2$ and

(1) $A_{1,1}$ has full row rank $\nu_1$,

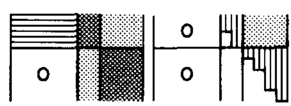
(2) $\left[\begin{array}{c} E_{1,2} \\ \hline E_2 \end{array}\right]$ has full column rank $\rho_1$,

(3) $\left[\begin{array}{c} E_{1,2} \\ \hline E_2 \end{array}\right]$ and $E_2$ are in echelon form.

In (3.2.3) the rank properties just mentioned are visualized by horizontal and vertical lines indicating full row and column rank, respectively.

*Step* 2. We repeat the above procedure for $\lambda E_2 - A_2$. So we start with partitioning $A_2$ conformally with $E_2$ such that the zero and nonzero columns are separated. We then have for the complete matrices $A$ and $E$

$$P_1[A_1 \; \| \; E_1]Q_1 = \left[\begin{array}{c|c||c|c} A_{1,1} & A_{1,2} & 0 & E_{1,2} \\ \hline 0 & A_2 & 0 & E_2 \end{array}\right]$$

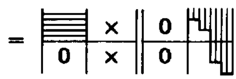
$=$ 


$$= \left[\begin{array}{c|c|c||c|c|c} A_{1,1} & A_{1,2} & A_{1,3} & 0 & E_{1,2} & E_{1,3} \\ \hline 0 & A_{2,2} & A_{2,3} & 0 & 0 & E_{2,3} \end{array}\right]. \qquad (3.2.4)$$

(Note that we have reused the names $A_{1,2}$ and $E_{1,2}$ for submatrices of the original matrices having these names.)

The matrix $A_{2,2}$ is now compressed to full row rank by row rotations plus column pivoting while $E_{2,3}$ is kept in an echelon form by column rotations (see step 1). Now all column transformations are performed on the whole matrices $A$ and $E$ (thus not only on $A_{2,2}$ and $E_{2,3}$). Notice that if column pivoting is needed in $A$ (i.e., in $A_{2,2}$), then applying these transformations to the corresponding columns in $E$ will destroy the echelon form of $E_{1,2}$. If one insists on preserving this echelon form, one has to restore the original order of columns in $A_{1,2}$, $A_{2,2}$, and $E_{1,2}$ (see Remark 3.1.2). However, this is not essential for the sequel. Let the accumulated row and column transformations acting on $A_2$ and $E_2$ be represented by the $m_2 \times m_2$ matrix $\tilde{P}_2$ and $n_2 \times n_2$ matrix $\tilde{Q}_2$, respectively. Then at the end of step 2 we have (after reusing block names)

$$\tilde{P}_2[A_2 \parallel E_2]\tilde{Q}_2 = \tilde{P}_2[A_{2,2} \mid A_{2,3} \parallel 0 \mid E_{2,3}]\tilde{Q}_2$$



$$= \begin{array}{c} \left[\begin{array}{c|c||c|c} A_{2,2} & A_{2,3} & 0 & E_{2,3} \\ \hline 0 & A_3 & 0 & E_3 \end{array}\right] \begin{array}{l} \}\,\nu_2 \\ \}\,m_3 \end{array} \\ \begin{array}{cccc} \underbrace{\qquad}_{\mu_2} & \underbrace{\qquad}_{n_3} & \underbrace{\qquad}_{\mu_2} & \underbrace{\qquad}_{n_3=\rho_2} \end{array} \end{array} \tag{3.2.5}$$

where

(1) $A_{2,2}$ has full row rank,

(2) $\left[\dfrac{E_{2,3}}{E_3}\right]$ has full column rank,

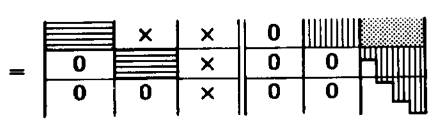
(3) $\left[\dfrac{E_{2,3}}{E_3}\right]$ and $E_3$ are in echelon form.

Defining

$$P_2 = \operatorname{diag}\left\{I_{\nu_1}, \tilde{P}_2\right\}, \qquad Q_2 = \operatorname{diag}\left\{I_{\mu_1}, \tilde{Q}_2\right\}, \tag{3.2.6}$$

we have

$$P_2P_1[A_1 \parallel E_1]Q_1Q_2 = \left[\begin{array}{c|c||c|c} A_{1,1} & A_{1,2}\tilde{Q}_2 & 0 & E_{1,2}\tilde{Q}_2 \\ \hline 0 & \tilde{P}_2A_2\tilde{Q}_2 & 0 & \tilde{P}_2E_2\tilde{Q}_2 \end{array}\right]$$



$$= \begin{array}{c} \left[\begin{array}{c|c|c||c|c|c} A_{1,1} & A_{1,2} & A_{1,3} & 0 & E_{1,2} & E_{1,3} \\ \hline 0 & A_{2,2} & A_{2,3} & 0 & 0 & E_{2,3} \\ \hline 0 & 0 & A_3 & 0 & 0 & E_3 \end{array}\right] \begin{array}{l} \}\,\nu_1 \\ \}\,\nu_2 \\ \}\,m_3 \end{array} \\ \begin{array}{cccccc} \underbrace{\qquad}_{\mu_1} & \underbrace{\qquad}_{\mu_2} & \underbrace{\qquad}_{n_3} & \underbrace{\qquad}_{\mu_1} & \underbrace{\qquad}_{\mu_2} & \underbrace{\qquad}_{n_3} \end{array} \end{array}$$

(3.2.7)

where

(1) $A_{1,1}$ and $A_{2,2}$ have full row rank,

(2) $E_{1,2}$ and $\left[\begin{array}{c} E_{2,3} \\ \hline E_3 \end{array}\right]$ have full column rank,

(3) $\left[\begin{array}{c} E_{2,3} \\ \hline E_3 \end{array}\right]$ and $E_3$ are in echelon form.

*Step j (Induction Step).* Repeat the procedure for the pencil $\lambda E_j - A_j$ until the $m_j \times n_j$ matrix $E_j$ has full column rank by applying Algorithm 3.1.1 to the pair $(A_j, E_j)$.

The procedure for row compression of $A$ while keeping $E$ in an echelon form can be summarized by the following algorithm.

ALGORITHM 3.2.1

**Step 0: comment** Transform $E$ to column echelon form, displaying its rank $\rho$;
**Result** $E_1 := P_0 E Q_0 = [\underbrace{\ 0\ }_{n-\rho} \mid \underbrace{E_{1,2}}_{\rho}]\}m;$

**comment** Initialization. $P$ and $Q$ are the row and column transformations;
$P := P_0;\ Q := Q_0;\ j := 1;\ A_1 := P_0 A Q_0;\ m_1 := m;\ n_1 := n;\ \rho_1 := \rho;\ \mu_1 := n - \rho_1;$
**Step j: comment** Induction step for $j \geqslant 1$;
**comment** Partition $A_j$ conformally with $E_j$;
**Result** $A_j = [\underbrace{A_{j,j}}_{\mu_j} \mid \underbrace{A_{j,j+1}}_{\rho_j}]\}m_j;\ E_j = [\underbrace{\ 0\ }_{\mu_j} \mid \underbrace{E_{j,j+1}}_{\rho_j}]\}m_j;$

**if** $\mu_j = 0$ **then begin** $l := j - 1$; **exit; end;**
**comment** Compress $A_{j,j}$ to full rank $\nu_j$ while keeping $E_j$ in echelon form. The resulting transformation matrices are $\tilde{P}_j$ and $\tilde{Q}_j$;

**Result** $\tilde{P}_j A_j \tilde{Q}_j = \tilde{P}_j [A_{j,j} \mid A_{j,j+1}] \tilde{Q}_j = \begin{array}{cl} \left[\begin{array}{c|c} A_{j,j} & A_{j,j+1} \\ \hline 0 & A_{j+1} \end{array}\right] & \begin{array}{l} \}\nu_j \\ \}m_j - \nu_j; \end{array} \\ \begin{array}{cc} \mu_j & \rho_j \end{array} & \end{array}$

$$\tilde{P}_j E_j \tilde{Q}_j = \tilde{P}_j [0 \mid E_{j,j+1}] \tilde{Q}_j = \left[\begin{array}{c|c} 0 & E_{j,j+1} \\ \hline 0 & E_{j+1} \end{array}\right];$$

**comment** Update and partition blocks with column index $j$;
**for** $i = 1$ **step 1 until** $j - 1$ **do**
**begin** $[A_{i,j} \mid A_{i,j+1}] := A_{i,j} \tilde{Q}_j;\ [E_{i,j} \mid E_{i,j+1}] := E_{i,j} \tilde{Q}_j$ **end;**
Determine $\rho_{j+1} = \text{rank}(E_{j+1})$;
**comment** Update;
$s_j := \Sigma_1^{j-1} \nu_i;\ t_j := \Sigma_1^{j-1} \mu_i;\ P := \text{diag}\{I_{s_j}, \tilde{P}_j\} P;\ Q := Q\,\text{diag}\{I_{t_j}, \tilde{Q}_j\};$
$m_{j+1} := m_j - \nu_j;\ n_{j+1} := n_j - \mu_j;\ j := j + 1;$

**go to step j;**
**comment** End of Algorithm 3.2.1;

Note that the algorithm stops when $E_j$ has full column rank (then $\rho_j = n_j$ and $l = j-1$). Furthermore, since $E_j$ is in echelon form, $E_{j+1}$ is also in such a form. Hence, the determination of $\operatorname{rank}(E_{j+1})$ is trivial. The algorithm reduces $\lambda E - A$ to the following form (for some $X$):

$$P(\lambda E - A)Q$$

$$= \left[\begin{array}{c|c} \lambda E_{\epsilon\infty} - A_{\epsilon\infty} & X \\ \hline 0 & \lambda E_{f\eta} - A_{f\eta} \end{array}\right]$$

$$= \begin{array}{c} \left[\begin{array}{c|c|c|c|c} -A_{1,1} & \lambda E_{1,2} - A_{1,2} & \cdots & \lambda E_{1,l} - A_{1,l} & \lambda E_{1,l+1} - A_{1,l+1} \\ \hline 0 & -A_{2,2} & \cdots & \lambda E_{2,l} - A_{2,l} & \lambda E_{2,l+1} - A_{2,l+1} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline 0 & 0 & \cdots & -A_{l,l} & \lambda E_{l,l+1} - A_{l,l+1} \\ \hline 0 & 0 & \cdots & 0 & \lambda E_{l+1} - A_{l+1} \end{array}\right] \begin{array}{l} \}\nu_1 \\ \}\nu_2 \\ \\ \}\nu_l \\ \}m_{l+1} \end{array} \\ \begin{array}{ccccc} \underbrace{\qquad}_{\mu_1} & \underbrace{\qquad\qquad}_{\mu_2} & & \underbrace{\qquad\qquad}_{\mu_l} & \underbrace{\qquad\qquad\qquad}_{n_{l+1}} \end{array} \end{array} \tag{3.2.8}$$

where $\lambda E_{l+1} - A_{l+1} \doteq \lambda E_{f\eta} - A_{f\eta}$ and

(1) $E_{l+1}$ has full column rank and is in echelon form,
(2) the $A_{i,i}$ have full row rank $\nu_i$ $(i = 1,\ldots,l)$,
(3) the $E_{i-1,i}$ have full column rank $\mu_i$ $(i = 2,\ldots,l)$,
(4) $\left[\begin{array}{c} E_{l,l+1} \\ \hline E_{l+1} \end{array}\right]$ has full column rank and is in echelon form.

From this it follows (putting $\mu_{l+1} = 0$) that

$$e_i \doteq \mu_i - \nu_i \geqslant 0 \qquad \text{for} \quad i = 1,\ldots,l,$$

$$d_i \doteq \nu_i - \mu_{i+1} \geqslant 0 \qquad \text{for} \quad i = 1,\ldots,l. \tag{3.2.9}$$

We note that this form is exactly the same—after permuting block rows and columns—as the one obtained after applying Algorithm 4.1 in [18] to $\lambda E - A$. Hence, the following lemma stated in [18] is also valid here.

LEMMA 3.2.1. *The indices* $\{e_i \mid i=1,\dots,l\}$ *and* $\{d_i \mid i=1,\dots,l\}$ *completely determine the Kronecker column indices* $\{\epsilon_i\}$ *and the infinite elementary divisors with their degrees* $\{\delta_i\}$ *as follows*:

(1) *there are* $d_i$ *infinite elementary divisors of degree* $i$ $(i=1,\dots,l)$;
(2) *there are* $e_i$ *Kronecker blocks* $L_{i-1}$ *of size* $(i-1)\times i$ $(i=1,\dots,l)$.

*Proof.* See proof of Lemma 4.3 and Corollary 4.4 in [18]. ∎

We emphasize that Algorithm 3.2.1 separates recursively the structure elements of $\lambda E - A$ whose coefficient matrix of $\lambda$ has defective *column* rank from the others. Clearly, we can formulate the dual algorithm that acts on the *row* rank of the coefficient of $\lambda$. Thereby the *row* Kronecker indices are detected together with the structure at infinity. The dual algorithm is simply obtained by interchanging the row and column compressions in Algorithm 3.2.1 (see also [18]). In the interest of brevity we do not work this out. We refer to this algorithm as Algorithm 3.2.1-D. It reduces $\lambda E - A$ to the following form [compare this with (3.2.8)]:

$$P(\lambda E - A)Q$$
$$= \begin{array}{cc}
\left[\begin{array}{c|c|c|c|c}
\lambda E_{k+1}-A_{k+1} & \lambda E_{k,k+1}-A_{k,k+1} & \cdots & X & X \\ \hline
0 & -A_{k,k} & \cdots & \lambda E_{2,k}-A_{2,k} & \lambda E_{1,k}-A_{1,k} \\ \hline
\vdots & \vdots & \ddots & \vdots & \vdots \\ \hline
0 & 0 & \cdots & -A_{2,2} & \lambda E_{1,2}-A_{1,2} \\ \hline
0 & 0 & \cdots & 0 & -A_{1,1}
\end{array}\right] &
\begin{array}{l} \}\, m'_{k+1} \\ \}\, \nu'_k \\ \\ \}\, \nu'_2 \\ \}\, \nu'_1 \end{array} \\
\begin{array}{ccccc} \underbrace{\hphantom{\lambda E_{k+1}-A_{k+1}}}_{n'_{k+1}} & \underbrace{\hphantom{\lambda E_{k,k+1}-A_{k,k+1}}}_{\mu'_k} & \hphantom{\cdots} & \underbrace{\hphantom{\lambda E_{2,k}-A_{2,k}}}_{\mu'_2} & \underbrace{\hphantom{\lambda E_{1,k}-A_{1,k}}}_{\mu'_1} \end{array} &
\end{array} \tag{3.2.10}$$

where

(1) $E_{k+1}$ has full row rank and is in row echelon form,
(2) the $A_{i,i}$ have full column rank $\mu'_i$ $(i=1,\dots,k)$,
(3) the $E_{i-1,i}$ have full row rank $\nu'_i$ $(i=2,\dots,k)$,
(4) $[E_{k+1} \mid E_{k,k+1}]$ has full row rank and is in row echelon form.

Let us now return to the pencil having the form (3.2.8), being the result of applying Algorithm 3.2.1 to $\lambda E - A$. In the transformed pencil the bottom block $\lambda E_{l+1} - A_{l+1}$ has only Kronecker row indices and finite elementary divisors as structure elements. Only the first of these two elements has defective row rank in the coefficient of $\lambda$. Hence, applying Algorithm 3.2.1-D to $\lambda E_{f\eta} - A_{f\eta}$ yields the separation of these two structure elements, i.e.,

$$P_1(\lambda E_{f\eta} - A_{f\eta})Q_1 = \left[\begin{array}{c|c} \lambda E_f - A_f & X \\ \hline 0 & \lambda E_\eta - A_\eta \end{array}\right]. \tag{3.2.11}$$

Similarly, the left upper block $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ in (3.2.8) is transformed by Algorithm 3.2.1-D to

$$P_2(\lambda E_{\epsilon\infty} - A_{\epsilon\infty})Q_2 = \left[\begin{array}{c|c} \lambda E_\epsilon - A_\epsilon & X \\ \hline 0 & \lambda E_\infty - A_\infty \end{array}\right]. \tag{3.2.12}$$

Here the infinite elementary divisors are separated form the remaining $\lambda E_\epsilon - A_\epsilon$ (see also [18]). In Section 3.3 we present new algorithms that transform the pencil (3.2.8) into the forms (3.2.11) and (3.2.12) with another algorithm than Algorithm 3.2.1-D. The transformation matrices involved consist again of Givens rotations and permutations but fully exploit the special structure of the submatrices in (3.2.8). These algorithms are also more efficient than Algorithm 3.2.1-D. Finally, we note that in the resulting forms the blocks $\lambda E_f - A_f$ and $\lambda E_\infty - A_\infty$ will be upper triangular, which is not guaranteed by earlier algorithms.

### 3.3. *Refined Algorithms for Further Reduction to Schur Form*

In this section we show how to exploit the structure obtained by Algorithm 3.2.1 in the pencils $\lambda E_{f\eta} - A_{f\eta}$ and $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ in order to obtain more refined algorithms of lower computational complexity (i.e. less flops). Section 3.3.1 is a refinement of Algorithm 3.2.1-D applied to $\lambda E_{f\eta} - A_{f\eta}$, and the next two subsections yield improvements with respect to algorithm 3.2.1-D applied to $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$. Along the way a useful triangular form (see the concluding remarks) is also obtained.

*3.3.1. Separation of $\lambda E_f - A_f$ and $\lambda E_\eta - A_\eta$.* By applying Algorithm 3.2.1-D to $\lambda E_{f\eta} - A_{f\eta}$ no advantage is taken of the fact that $E_{f\eta}$ is in column echelon form. We now present a better alternative for transforming $\lambda E_{f\eta} - A_{f\eta}$ to its Schur form.

We start with the pertranspose $(\lambda E_{f\eta} - A_{f\eta})^P$ of $\lambda E_{f\eta} - A_{f\eta}$. Then $E_{f\eta}^P$ has full row rank and is in row echelon form. $E_{f\eta}^P$ is then reduced to upper triangular form $E_1$ by applying Givens rotations or Householder transformations to its columns. If the same transformation $Q_0$ is applied to $A_1$, we have

$$E_1 \doteq E_{f\eta}^P Q_0 = \left[\begin{array}{c|cccccc} & \times & x & x & \cdots & x \\ & & \times & x & \cdots & x \\ 0 & & & \ddots & & \vdots \\ & & & & \times & x \\ & & & & & \times \end{array}\right], \tag{3.3.1}$$

$$A_1 \doteq A_{f\eta}^P Q_0.$$

By applying Algorithm 3.2.1 (without step 0) to $\lambda E_1 - A_1$ we then find

$$P_1(\lambda E_1 - A_1)Q_1 = \left[\begin{array}{c|c} \lambda\hat{E} - \hat{A} & X \\ \hline 0 & \lambda E_S - A_S \end{array}\right], \tag{3.3.2}$$

where $E_S$ is square and invertible. Finally, transforming back to the original pencil gives

$$P(\lambda E_{f\eta} - A_{f\eta})Q = \left[\begin{array}{c|c} (\lambda E_S - A_S)^P & X^P \\ \hline 0 & (\lambda\hat{E} - \hat{A})^P \end{array}\right]. \tag{3.3.3}$$

REMARK 3.3.1. Note that $E_1$ is in echelon form. (The reduction of $E_{f\eta}$ to $E_1$ can be done a little faster than applying step 0 of Algorithm 3.2.1 to $E_{f\eta}$ by exploiting its echelon form.) Hence we may apply Algorithm 3.2.1 to $\lambda E_1 - A_1$. Moreover, a refined version of Algorithm 3.2.1 can be used, since $E_1$ is upper triangular. For example, in this case one does not have to keep track of the structure of the transformed $E_j$, since they are all upper triangular as a consequence of Algorithm 3.2.1 (see also [18]).

*3.3.2. Triangularization of the Pencil $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$.* Here we consider the pencil $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ in (3.2.8), which only contains the infinite elementary divisors and Kronecker column indices. Instead of applying Algorithm 3.2.1-D, we develop in the next subsection a new algorithm for separating the two structural elements of $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$. Hereby the special properties of this pencil are exploited throughout the algorithm. But first the "staircases" of $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ have to be triangularized, which is explained now.

The starting point here is the $m_{\epsilon\infty} \times n_{\epsilon\infty}$ pencil $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ having the form indicated in (3.2.8). For notational convenience, we will write $\lambda E - A$

instead of $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$. We thus have

$$\begin{array}{c}
A \doteq \begin{array}{cl}
\left[\begin{array}{c|c|c|c|c|c}
A_{1,1} & A_{1,2} & \cdot & \cdot & \cdot & A_{1,l} \\ \hline
 & A_{2,2} & A_{2,3} & \cdot & \cdot & \cdot \\ \hline
 & & \ddots & \ddots & & \cdot \\ \hline
 & & & \ddots & \ddots & \cdot \\ \hline
 & & & & A_{l-1,l-1} & A_{l-1,l} \\ \hline
 & & & & & A_{l,l}
\end{array}\right]
&
\begin{array}{l}
\}\,\nu_1 \\ \}\,\nu_2 \\ \\ \\ \}\,\nu_{l-1} \\ \}\,\nu_l
\end{array}
\\
\begin{array}{cccccc}
\underbrace{\qquad}_{\mu_1} & \underbrace{\qquad}_{\mu_2} & & & \underbrace{\qquad}_{\mu_{l-1}} & \underbrace{\qquad}_{\mu_l}
\end{array} &
\end{array}
\\[2ex]
E \doteq \begin{array}{cl}
\left[\begin{array}{c|c|c|c|c|c}
0 & E_{1,2} & \cdot & \cdot & \cdot & E_{1,l} \\ \hline
 & 0 & E_{2,3} & \cdot & \cdot & \cdot \\ \hline
 & & 0 & \ddots & & \cdot \\ \hline
 & & & \ddots & \ddots & \cdot \\ \hline
 & & & & 0 & E_{l-1,l} \\ \hline
 & & & & & 0
\end{array}\right]
&
\begin{array}{l}
\}\,\nu_1 \\ \}\,\nu_2 \\ \\ \\ \}\,\nu_{l-1} \\ \}\,\nu_l
\end{array}
\\
\begin{array}{cccccc}
\underbrace{\qquad}_{\mu_1} & \underbrace{\qquad}_{\mu_2} & & & \underbrace{\qquad}_{\mu_{l-1}} & \underbrace{\qquad}_{\mu_l}
\end{array} &
\end{array}
\end{array} \tag{3.3.4}$$

The algorithm consists of triangularizing the blocks $E_{i-1,i}$ $(i=2,\dots,l)$ and $A_{i,i}$ $(i=1,\dots,l)$ by row and column rotations, respectively. These unitary transformations can be carried out in such a way that the structure of the blocks already treated is not destroyed when transforming the next ones.

The algorithm consists of $l$ steps. It starts with the triangularization of the blocks in the order $(A_{l,l}, E_{l-1,l})$ up to $(A_{2,2}, E_{1,2})$ followed by $A_{1,1}$. The matrices $P_i$ and $Q_i$ corresponding with $A_{i,i}$ and $E_{i-1,i}$ are defined recursively by the following algorithm.

ALGORITHM 3.3.1 [Reduction of $\lambda E - A$ $(\doteq \lambda E_{\epsilon\infty} - A_{\epsilon\infty})$ to triangular form].

**comment** Perform $l$ reduction steps;
**for** $i = l$ **step** $-1$ **until** 2 **do**
**begin**
  **comment** Reduce $A_{i,i}$ to upper triangular form by a column transformation $Q_i$ using a $QR$ decomposition. Also update blocks with column index $i$ of $A$;

**for** $k = 1$ **step** 1 **until** $i$ **do** $A_{k,i} := A_{k,i}Q_i$;
**comment** Apply this transformation to the same columns in $E$;
**for** $k = 1$ **step** 1 **until** $i - 1$ **do** $E_{k,i} := E_{k,i}Q_i$;
**comment** Reduce $E_{i-1,i}$ to upper triangular form by a row transformation $P_{i-1}$ using a $QR$ decomposition. Also update blocks with row index $i - 1$ of $E$;
**for** $j = i$ **step** 1 **until** $l$ **do** $E_{i-1,j} := P_{i-1}E_{i-1,j}$;
**comment** Apply this transformation to the same rows in $A$;
**for** $j = i - 1$ **step** 1 **until** $l$ **do** $A_{i-1,j} := P_{i-1}A_{i-1,j}$;
**end**
**comment** Perform the final transformation of $A_{1,1}$ using a $QR$ decomposition;
$A_{1,1} := A_{1,1}Q_1$;
**comment** End of Algorithm 3.3.1;

At the end of this algorithm we have constructed unitary matrices $P = \operatorname{diag}\{P_1, \ldots, P_l\}$ with $P_l = I$ and $Q = \operatorname{diag}\{Q_1, \ldots, Q_l\}$ such that $P(\lambda E - A)Q$ has the form shown in Figure 3. Notice that all diagonal elements of the upper triangular matrices are nonzero because all $A_{i,i}$ and $E_{i-1,i}$ have full row and column rank, respectively.

$$[A_1 \parallel E_1] \doteq P[A \parallel E]Q \tag{3.3.5}$$

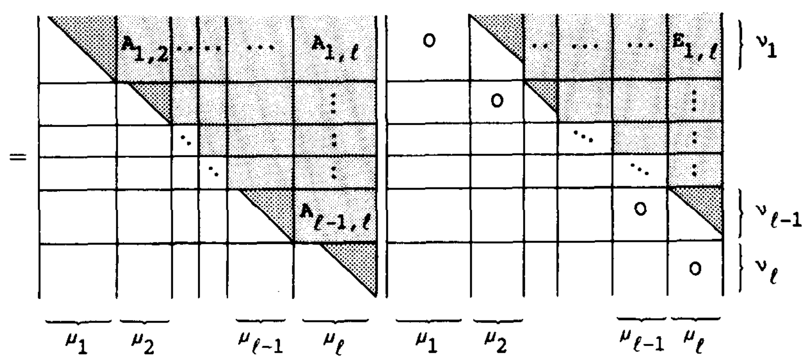



FIG. 3.

3.3.3. *Separation of* $\lambda E_\epsilon - A_\epsilon$ *and* $\lambda E_\infty - A_\infty$. Consider the $m_{\epsilon\infty} \times n_{\epsilon\infty}$ pencil $\lambda E_1 - A_1 = P(\lambda E_{\epsilon\infty} - A_{\epsilon\infty})Q$ having the form (3.3.5). We now describe how the decomposition (3.2.12) can be obtained in a numerically stable and efficient way without applying Algorithm 3.2.1-D. The pencil

$\lambda E_1 - A_1$ is transformed by unitary matrices $U$ and $V$ such that

$$U(\lambda E_1 - A_1)V = UP(\lambda E_{\epsilon\infty} - A_{\epsilon\infty})QV = \left[\begin{array}{c|c} \lambda E_\epsilon - A_\epsilon & X \\ \hline 0 & \lambda E_\infty - A_\infty \end{array}\right], \tag{3.3.6}$$

where $A_\epsilon$ is an upper triangular matrix having full row rank and $E_\epsilon$ is a strictly upper triangular matrix having full row rank. The matrix $A_\infty$ is invertible and upper triangular, and $E_\infty$ is strictly upper block triangular with zero diagonal elements. Thus $E_\infty$ is nilpotent. In Figure 4 an example of the resulting pencil (3.3.6) is given that illustrates the properties just mentioned.

We note that all triangular matrices in Figure 4 have nonzero diagonal elements from which their rank properties directly follow. The transformations used in the reduction process are all row or column Givens rotations applied in a judiciously chosen order. Before describing these transformations in full detail, we shall first sketch the reduction process. This process consists of $l-1$ steps, where $l$ is the number of nontrivial blocks $A_{i,i}$ in (3.3.5). In each step $l-k+1$ $(k=l,\dots,2)$ a $\nu_{k-1} \times \mu_k$ block $E_{k-1,k}$ in (3.3.5) is reduced to a square upper triangular matrix. Thereby $A_{k-1,k-1}$ is also reduced, while in the meantime blocks of $\lambda E_\infty - A_\infty$ are generated.

Initially, the pencil $\lambda E_\infty - A_\infty$ has zero dimensions. The reduction of an $E_{k-1,k}$ is done row by row, starting with the bottom row. Thereby all elements in the bottom row of the blocks $A_{k-1,j}$ $(j \geqslant k)$ and $E_{k-1,j}$ $(j > k)$ are annihilated. Let $i_k$ be the row index of this row in $A$ (and $E$). The transformations for annihilation can be chosen such that there is no fill-in in row $i_k$ in $A$ and $E$. We note that the matrix $A_{k,k}$ is affected by these transformations, but it remains upper triangular with nonzero "diagonal." Consequently, row $i_k$ in $A$ has then only one nonzero element, say $p$, being the bottom diagonal element of $A_{k,k}$. Row $i_k$ in $E$ is now completely zero.

Hereafter cyclic row and column permutations are carried out that move this nonzero element $p$ to the right bottom corner of $A$. Of course, the same transformations are applied to $E$.

$$\left[\begin{array}{c|c||c|c} A_\epsilon & X & E_\epsilon & X \\ \hline 0 & A_\infty & 0 & E_\infty \end{array}\right] =$$

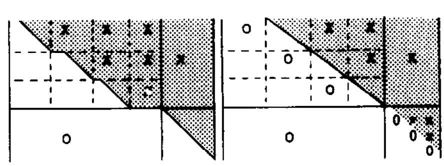


FIG. 4. Example of the final structure of the decomposition (3.3.6).

Let us now outline the remainder of this section. First we discuss the starting situation of the pencil to be transformed. Next the first step of the algorithm is described by means of an example that is typical for the general situation. Moreover we prove that the "diagonal" elements of the $E$-block transformed in step 1 remain nonzero (see Lemma 3.3.1). This property will be of crucial importance in the next steps. Hereafter we consider the general step $l-k+1>1$, which is much more complicated than the first one. Therefore we start with indicating the general situation and summarizing the properties of the transformed matrices (see Theorem 3.3.1). Next the transformations are described in detail. Then we are ready to prove Theorem 3.3.1. The proof is given in parts by the Lemmas 3.3.2 and 3.3.3. Hereafter we note that the procedure can be modified so that pure permutations can be avoided without any extra computational effort. We present the algorithm describing the whole procedure including these modifications.

*Starting situation for the pencil to be transformed.* As stated before, we consider the $m_{\epsilon\infty}\times n_{\epsilon\infty}$ pencil $\lambda E_1 - A_1$ having the form (3.3.5). Clearly, we assume $l\geqslant 1$ and $\mu_l>0$. We distinguish two cases, namely $\nu_l\neq 0$ and $\nu_l=0$.

*Case I*: $\nu_l\neq 0$. If $l=1$, then it can be readily verified that $\lambda E_1 - A_1$ already has the form (3.3.6) with $\lambda E_\epsilon - A_\epsilon$ having dimension $0\times(\mu_1-\nu_1)$. It contains $\mu_1-\nu_1$ Kronecker column indices equal to 0. Here the pencil $\lambda E_\infty - A_\infty$ is $\nu_1\times\nu_1$ with $A_\infty$ upper triangular and invertible and $E_\infty$ completely zero. So we may now assume $l\geqslant 2$. Then we have the situation shown in Figure 5. Let us partition $A_{l,l}$ as $A_{l,l}=[0\,|\,\hat{A}_{l,l}]$ where $\hat{A}_{l,l}$ is square upper triangular. Then we partition all blocks $A_{i,l}$ and $E_{i,l}$ $(1\leqslant i\leqslant l-1)$ conformably with $A_{l,l}$ (see dashed lines in Figure 5). Thus these blocks are split into two subblocks. Now all blocks $A_{i,l}$ and $E_{i,l}$ $(1\leqslant i\leqslant l-1)$ are redefined by only taking their left subblock. Then we have the form shown in Figure 6, where $\lambda\hat{E}-\hat{A}$ has the form (3.3.5) but with $\nu_l=0$ and $\mu_l$ replaced by $\mu_l-\nu_l$. The square $\nu_l\times\nu_l$ pencil $\lambda\hat{0}_{l,l}-\hat{A}_{l,l}$ just separated in $\lambda E_1 - A_1$ becomes the right bottom block of $\lambda E_\infty - A_\infty$ (recall the properties of several

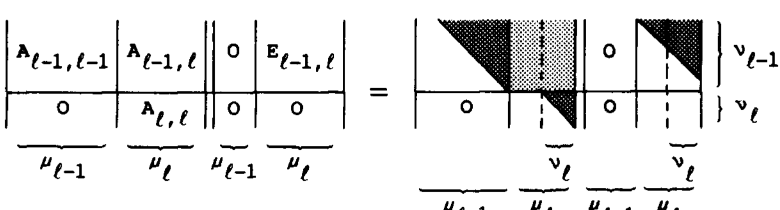



FIG. 5. Starting situation with $\nu_l\neq 0$.

$$[A_1 \| E_1] = \left[\begin{array}{c:c||c:c} \hat{A} & X & \hat{E} & X \\ \hline 0 & \hat{A}_{l,l} & 0 & \hat{0}_{l,l} \end{array}\right]$$

$=$

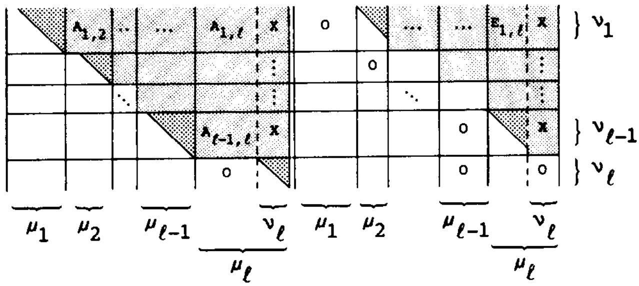



FIG. 6. Preliminary separation when $\nu_l \neq 0$.

pencils mentioned in Section 2). In this situation we shall proceed with $\lambda \hat{E} - \hat{A}$.

*Case II*: $\nu_l = 0$. If in this case $l = 1$, then no transformations are needed, since $\lambda E_1 - A_1 = \lambda E_\epsilon - A_\epsilon$. We now proceed with $\lambda \hat{E} - \hat{A}$.

Thus we see that in both cases I and II we may consider an $m \times n$ pencil $\lambda E - A$ having the form (3.3.5) with $\nu_l = 0$ and $l \leqslant 2$. We are left with $l - 1$ block rows and $l$ block columns. For notational convenience we will replace $l - 1$ by $l$. The starting situation for the remainder of this section is then given in Figure 7.

Let us now turn to the algorithm description. The algorithm for reduction to square upper triangular blocks $E_{k-1,k}$ consists of $l$ steps (one step per block). Of course, when $E_{k-1,k}$ is already a square matrix we can skip the reduction step. The algorithm starts with the transformation of the bottom

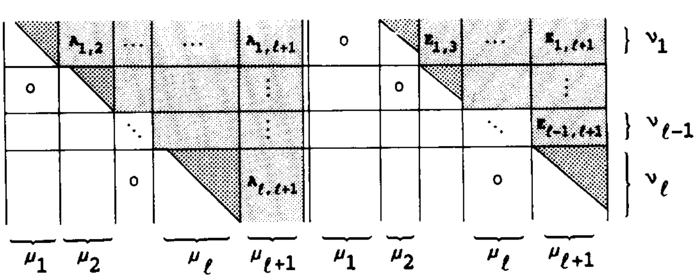



FIG. 7. Initial situation in this section.

$$
\begin{array}{c}
\\ \\ \\ \nu_l
\end{array}
\left|\begin{array}{cc||cc}
A_{1,l} & A_{1,l+1} & E_{1,l} & E_{1,l+1} \\
\vdots & \vdots & \vdots & \vdots \\
A_{l-1,l} & A_{l-1,l+1} & E_{l-1,l} & E_{l-1,l+1} \\
\hline
A_{l,l} & A_{l,l+1} & E_{l,l} & E_{l,l+1}
\end{array}\right|
=
\left|\begin{array}{c|c||c|c}
\mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\
\hline
0\mathrm{xxxx} & \mathrm{x\ x} & & \mathrm{x\ x} \\
\mathrm{xxx} & \mathrm{x\ x} & 0 & \mathrm{x} \\
\mathrm{xx} & \mathrm{x\ x} & & 0 \\
p_1 & \mathrm{x\ x} & & 0
\end{array}\right|
\begin{array}{l}
\Big\}\sum_{i=1}^{l-1}\nu_i \\
\Big\}\nu_l
\end{array}
$$

$$
\begin{array}{cccc}
\underbrace{\qquad}_{\mu_l} & \underbrace{\qquad}_{\mu_{l+1}} & \underbrace{\qquad}_{\mu_l} & \underbrace{\qquad}_{\mu_{l+1}}
\end{array}
\qquad\qquad
\begin{array}{cccc}
\underbrace{\qquad}_{\mu_l} & \underbrace{\quad}_{\mu_{l+1}} & \underbrace{\quad}_{\mu_l} & \underbrace{\quad}_{\mu_{l+1}}
\end{array}
$$

FIG. 8. Situation before reduction of $E_{l,l+1}$.

block row in $A$ and $E$. We shall now explain the first step. To this end, consider Figure 8, which shows a typical situation. Here $A_{l,l}$ is a $\nu_l \times \mu_l$ upper triangular matrix, $E_{l,l}$ is completely zero, and $E_{l,l+1}$ is a $\nu_l \times \mu_{l+1}$ upper triangular matrix. We have taken $\nu_l = 4$, $\mu_l = 5$, and $\mu_{l+1} = 2$.

*Step 1.* First we want to annihilate the $\mu_{l+1}$ elements in row $\nu_l$ of $A_{l,l+1}$. This is done by successively applying column rotations to the $\mu_l$th column of $A_{l,l}$ and the $i$th column of $A_{l,l+1}$ ($i = 1,\ldots,\mu_{l+1}$). Hereby the nonzero bottom diagonal element $p_1$ of $A_{l,l}$ is used as pivot. By construction this element remains nonzero after each transformation. These transformations are applied to all the blocks above $A_{l,l}$ and $A_{l,l+1}$. They are also performed on the corresponding columns in matrix $E$. Consequently, the elements in the $\mu_l$th column of the blocks $E_{i,l}$ ($i = 1,\ldots,l$) are changed. Thus these transformations may introduce new nonzero elements in $E_{l,l}$. But they do not disturb the triangular shape of $E_{l-1,l}$, because only its bottom diagonal elements have been changed. Thus all diagonal elements (except the last one perhaps) in $E_{l-1,l}$ are then zero. The matrix $E_{l,l+1}$ also remains upper triangular after the transformations. However, all its diagonal elements have been changed. In Lemma 3.3.1 below we shall prove that after the transformations all diagonal elements of $E_{l,l+1}$ are again nonzero. We conclude from the above that the triangular structure of $A_{l,l}$ and $E_{l,l+1}$ as well as the fully nonzero diagonals of $A_{l,l}$ and $E_{l,l+1}$ are invariant under these transformations. The procedure just described is illustrated in Figure 9, starting from Figure 8. For clarity, the columns to be transformed are shaded. Furthermore, in the figures the elements are marked with a prime after transformation. However, elements already marked with a prime are indicated without prime after transformation. Thus, in symbolic notation, $c \to c'$ and $c' \to c$.

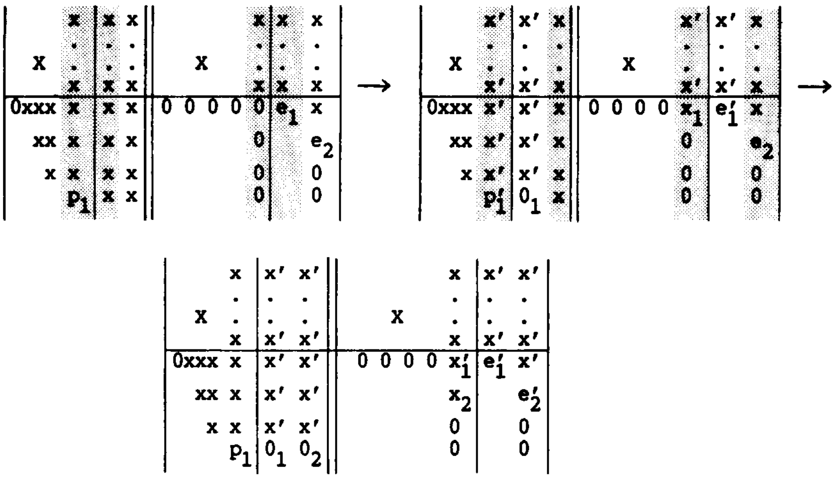


FIG. 9. Step 1 in the reduction of $E_{l,l+1}$.

Notice that each time a zero has been created in the last row of $A_{l,l+1}$, a (possibly) nonzero element has been introduced in column $\mu_l$ of $E_{l,l}$ (i.e., $x_1$ and $x_2$). Furthermore, the elements in the $\mu_l$th column of $E_{i,l}$ $(i = 1,\dots, l-1)$ are also changed. Notice also that after all $\mu_{l+1}$ transformations at least the last $\nu_l - \mu_{l+1}$ elements in column $\mu_l$ of $E_{l,l}$ are still zero. After annihilation of the complete row $\nu_l$ of $A_{l,l+1}$, cyclic column permutations are carried out such that the bottom diagonal element of $A_{l,l}$ becomes the top left element of $M_l$ (see Figure 10). The same names for the reduced blocks and their dimensions are used as before the transformations. For clarity, we have marked the new blocks and dimensions by a prime in Figure 10. Here $\mu'_l = \mu_l - 1 = 4$, $\mu'_{l+1} = \mu_{l+1} = 2$, $\nu'_l = \nu_l - 1 = 3$.

Observe that for the newly defined $A_{l,l}$ the numbers of rows and columns are reduced by one, whereas for the new $E_{l,l+1}$ and $A_{l,l+1}$ only their row dimension is decreased by one. Clearly, the new submatrices $A_{l,l}$ and $E_{l,l+1}$ are both upper triangular with fully nonzero diagonal. Furthermore, the newly defined submatrix $E_{l-1,l}$ is equal to the old one without the last column. We can thus conclude that the new matrices $A_{l,l}$, $E_{l,l+1}$, and $E_{l-1,l}$ have the same form and fully nonzero diagonal property as the previous ones. Now the whole procedure can be repeated until we obtain a square $E_{l,l+1}$. Clearly the procedure consists of $d_l \doteq \nu_l - \mu_{l+1}$ stages, since $E_{l,l+1}$ was originally a $\nu_l \times \mu_{l+1}$ matrix. Let us indicate the dimensions $\mu_i$ and $\nu_i$ at the end of stage $j$ by $\mu_i(j)$ and $\nu_i(j)$, respectively. We define $\mu_i(0) = \mu_i$ and

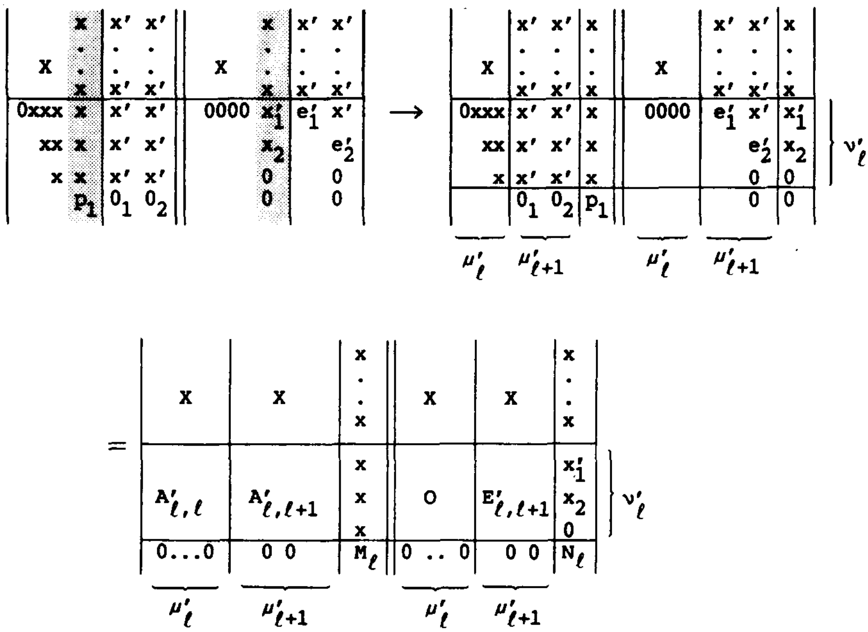


FIG. 10. Elimination of one row in $E_{l,l+1}$.

$\nu_i(0)=\nu_i$ for all $i$. Then we have for $1 \leqslant j \leqslant d_l$

$$
\begin{aligned}
\mu_l(j) &= \mu_l(j-1)-1, & \mu_i(j) &= \mu_i(j-1) && \text{for} \quad i \neq l, \\
\nu_l(j) &= \nu_l(j-1)-1, & \nu_i(j) &= \nu_i(j-1) && \text{for} \quad i \neq l.
\end{aligned} \tag{3.3.7}
$$

Hence for $1 \leqslant j \leqslant d_l$

$$
\begin{aligned}
\mu_l(j) &= \mu_l - j, & \mu_i(j) &= \mu_i && \text{for} \quad i \neq l, \\
\nu_l(j) &= \nu_l - j, & \nu_i(j) &= \nu_i && \text{for} \quad i \neq l.
\end{aligned} \tag{3.3.8}
$$

Notice that in Figure 10 we have indicated $\mu_i(1)$ as $\mu_i'$ and $\nu_i(1)$ as $\nu_i'$. The procedure is illustrated in Figure 11 (starting from the last situation given in Figure 10). Note that in this case $d_l=2$. Moreover, for typographical reasons we have used the notation $\mu_i''$ and $\nu_i''$ instead of $\mu_i(d_l)$ and $\nu_i(d_l)$. Here $\mu_l''=\mu_l-d_l=3$, $\mu_{l+1}''=\mu_{l+1}=2$, $\nu_l''=\nu_l-d_l=2$.

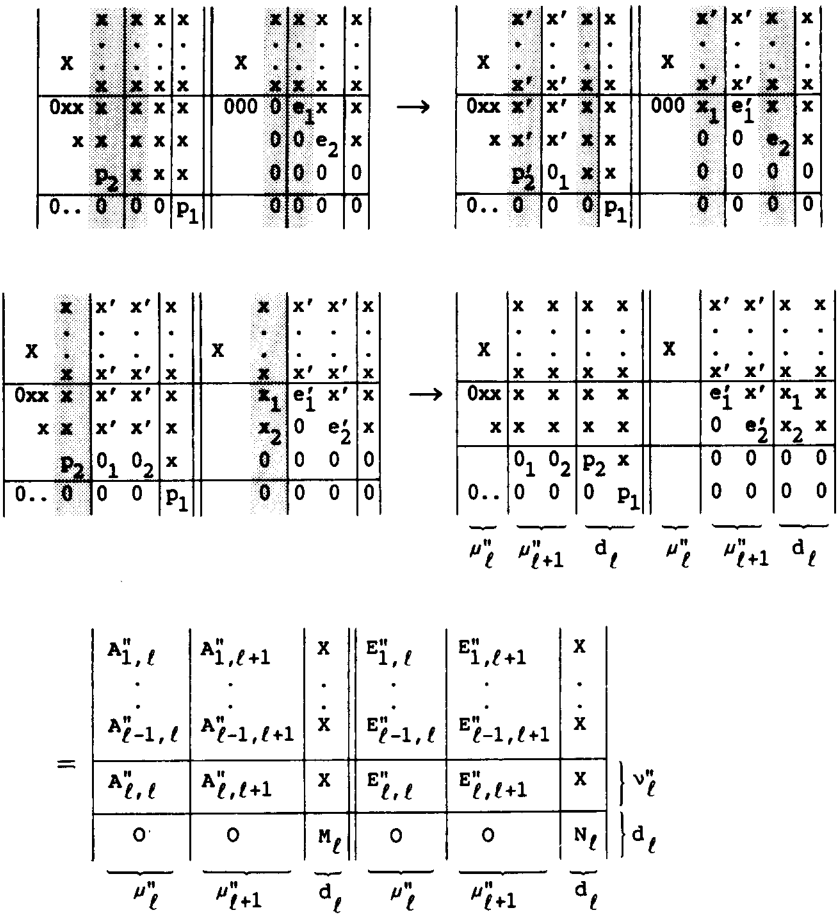


FIG. 11. Situation after reduction of $E_{l,l+1}$.

REMARK 3.3.2. It is easily seen that the order of the triangular matrix $M_l$ just built up in the right bottom corner of $A$ is equal to $d_l = \nu_l - \mu_{l+1}$. All its diagonal elements are nonzero because originally they were diagonal elements in $A_{l,l}$. Furthermore, it can be seen that the corresponding square submatrix $N_l$ in $E$ is completely zero. Indeed, at the end of each stage $j$ $(1 \leqslant j \leqslant d_l)$, when reducing the $\nu_l(j-1) \times \mu_{l+1}(j-1)$ block $E_{l,l+1}$, the permuted column in $E$ has at least its last $\delta_j \doteq \nu_l(j-1) - \mu_{l+1}(j-1)$ elements equal to zero. Using (3.3.8), we see that $\delta_j = \nu_l - \mu_{l+1} - (j-1) = d_l - (j-1)$. Furthermore, at the start of stage $j$ the matrix $E$ already has its

$$\left[A_{l,l}^{(i-1)} \mid A_{l,l+1}^{(i-1)} \| E_{l,l}^{(i-1)} \mid E_{l,l+1}^{(i-1)}\right]$$

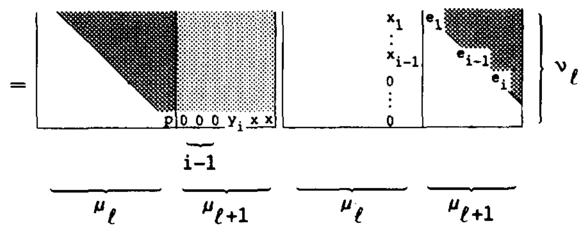



FIG. 12. The situation before annihilating $y_i$.

last $j-1$ rows completely zero by construction. Thus, we have $N_l = 0$. Hence, the pencil $\lambda N_l - M_l$ contains $\nu_l - \mu_{l+1}$ infinite elementary divisors of degree 1 of the original pencil $\lambda E - A$ in accordance with Lemma 3.2.1 in Section 3.2. Consequently, $\lambda N_l - M_l$ is equal to the current $\lambda E_\infty - A_\infty$.

As claimed above, in each stage $j$ $(1 \leqslant j \leqslant d_l)$ the diagonal elements of $E_{l,l+1}$ and $A_{l,l}$ are nonzero after annihilating the last row of $A_{l,l+1}$. Since the transformations in any stage $j$ do not change the structure of the matrices, it is sufficient to prove the properties of the blocks involved in an arbitrary stage $j$. This is done in Lemma 3.3.1 below. However, to avoid notational complexity we do not index the blocks and their dimensions with $j$ any more.

LEMMA 3.3.1. *We have for* $i = 1, \dots, \mu_{l+1}$ *(defining* $A_{l,j}^{(0)} \doteq A_{l,j}$, *and* $E_{l,j}^{(0)} \doteq E_{l,j}$ *for* $j = l, l+1$*)*

*Proposition* P($i$): *The situation just before annihilation of* $y_i \doteq A_{l,l+1}(\nu_l, i)$ *is as indicated in Figure* 12. *The bottom diagonal element* $p$ *of* $A_{l,l}^{(i-1)}$ *and the diagonal elements* $e_i$ $(i = 1, \dots, \mu_{l+1})$ *of* $E_{l,l+1}^{(i-1)}$ *are all nonzero.*

*Proof.* By induction. Clearly, Proposition P(1) is true.

Next, suppose proposition P($i$) is true for some $i \geqslant 1$. Hence $p \neq 0$. In the next annihilation step a column Givens rotation $G$ is constructed such that $(p, y_i)G = (p', 0)$ with $p' \neq 0$. Here $p \neq 0$. In the matrix $E$ we then have $(0, e_i)G = (x_i, e_i')$. Suppose $e_i' = 0$; then $G = \pm\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. This would imply that $(p, y_i)G = \pm(-y_i, p)$, which contradicts the construction of $G$. Thus, $e_i' \neq 0$. So we can conclude that after transformation all diagonal elements of $E_{l,l+1}$ are nonzero again, i.e., Proposition $P(i+1)$ is true. This completes the proof. ■

*Step* $l-k+1$ $(k<l)$. Let us describe the general step $l-k+1>1$ with $1 \leqslant k \leqslant l-1$ in which $E_{k,k+1}$ is reduced. Clearly we assume $l \geqslant 2$, since otherwise we are ready (see step 1). It will turn out that this step is *more complicated* than step 1 described above. Before giving more details, we first note that before reducing $E_{k,k+1}$ the dimensions of some blocks in $A$ and $E$ have been changed in the preceding steps. Furthermore, during the reduction of $E_{k,k+1}$ some block dimensions will be changed again. Since the block dimensions are of crucial importance, we introduce a special notation for them in order to keep track of their changes. To be precise, by adding the superscript $\geqslant i$ to the usual dimension notations $\nu_j$ and $\mu_j$ (viz. $\nu_j^{\geqslant i}$ and $\mu_j^{\geqslant i}$) we indicate that these are the dimensions of the $j$th block row and column in $A$ and $E$ after reducing the blocks in $E$ with row index $\geqslant i$ (i.e., $E_{i,i+1}$ up to $E_{l,l+1}$). For example, the matrix $E_{k,k+1}$ has the dimensions $\nu_k^{\geqslant k+1} \times \mu_{k+1}^{\geqslant k+1}$ and $\nu_k^{\geqslant k} \times \mu_{k+1}^{\geqslant k}$ before and after its reduction in step $l-k+1$, respectively. We emphasize that in the sequel the symbols $\nu_j$ and $\mu_j$ without superscript are strictly reserved for the situation just before step 1, unless otherwise stated. Now we can formulate

THEOREM 3.3.1. *We have for* $k=0,\dots,l-1$ (*interpreting* "*the situation before reduction of* $E_{0,1}$" *as the situation after reduction of* $E_{1,2}$)

*Proposition* G($k$). *The general situation before reduction of* $E_{k,k+1}$ *to a square block is as indicated in Figure* 13. *To be precise, the blocks* $E_{i,i+1}$ $(i=k+1,\dots,l)$, $A_{i,i}$ $(i=k,\dots,l)$, *and* $M_{k+1}$ *are upper triangular with nonzero diagonal elements. Moreover, the blocks* $E_{i,i+1}$ $(i=k+1,\dots,l)$, $M_{k+1}$, *and* $N_{k+1}$ *are square. The matrix* $N_{k+1}$ *is an upper triangular block matrix with all its diagonal elements zero.*

*The dimensions in Figure* 13 *are specified by*

$$\nu_i^{\geqslant k+1}=\nu_i\ (i\leqslant k),\quad \nu_i^{\geqslant k+1}=\mu_{i+1}^{\geqslant k+1}\qquad (k+1\leqslant i\leqslant l),$$

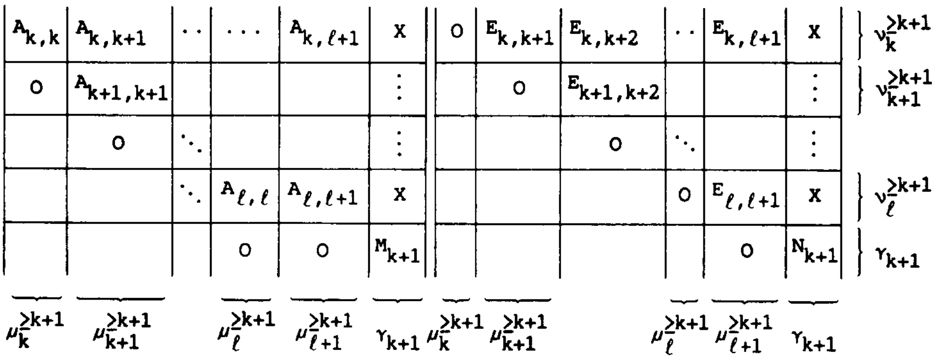
$$\mu_i^{\geqslant k+1}=\mu_i\ (i\leqslant k),\quad \mu_i^{\geqslant k+1}=\mu_i-d_i^{\geqslant i+1}\qquad (k+1\leqslant i\leqslant l+1),$$

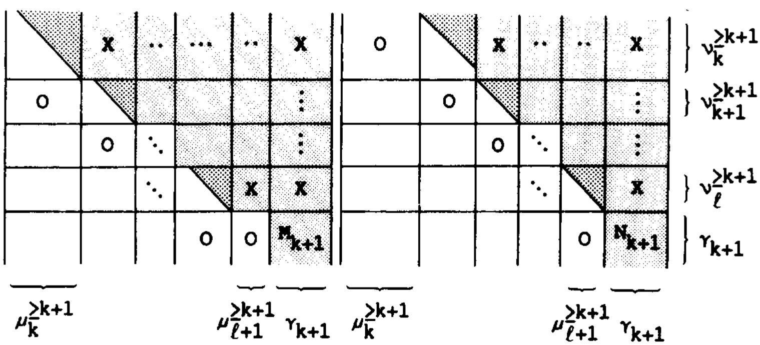
*where*

$$d_i^{\geqslant i+1}\doteq \nu_i^{\geqslant i+1}-\mu_{i+1}^{\geqslant i+1}=\sum_{j=i}^{l}(\nu_j-\mu_{j+1})\qquad (k\leqslant i\leqslant l),$$

$$d_{l+1}^{\geqslant l+2}\doteq 0.$$

$$[\hat{A} \| \hat{E}] \sim \left[\begin{array}{c|c||c|c} \hat{A}^{(k)} & X & \hat{E}^{(k)} & X \\ \hline 0 & M_{k+1} & 0 & N_{k+1} \end{array}\right] =$$





where

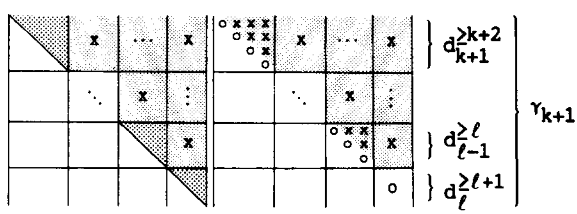
$[M_{k+1} \| N_{k+1}] =$ 

FIG. 13. The situation just before the reduction of $E_{k,k+1}$ $(k < l)$.

*The order of the matrices $M_{k+1}$ and $N_{k+1}$ is given by*

$$\gamma_{k+1} \doteq \sum_{i=k+1}^{l} \left(\nu_i^{\geq i+1} - \mu_{i+1}^{\geq i+1}\right) = \sum_{i=k+1}^{l} (i-k)(\nu_i - \mu_{i+1}).$$

*Proof.* We prove this by induction. Clearly, Proposition G($k$) is true for $k = l-1$ (see Remark 3.3.2 and Lemma 3.3.1 in step 1). Notice that in this case the matrix $N_{k+1}$ is a completely zero square matrix of order $d_l \doteq \nu_l - \mu_{l+1} = d_l^{\geq l+1}$. Furthermore, after reduction the matrix $E_{l,l+1}$ has indeed the dimensions $\nu_l^{\geq l} \times \mu_{l+1}^{\geq l} = \mu_{l+1}^{\geq l} \times \mu_{l+1}^{\geq l} = \mu_{l+1} \times \mu_{l+1}$ (see step 1). Suppose now G($k$) is true for some $k < l$. We shall show that then G($k-1$) is also true. To this end, we first describe step $l-k+1$ for the reduction of $E_{k,k+1}$. Hereafter we indicate the situation after this step. Moreover, in Lemmas 3.3.2 and 3.3.3 below we formulate and prove the properties of the transformed matrices after reducing $E_{k,k+1}$, i.e., before reducing $E_{k-1,k}$. This will prove the validity of G($k-1$). ■

Notice that in Figure 13 we have $\mu_k^{\geq k+1} \geq \nu_k^{\geq k+1} \geq \mu_{k+1}^{\geq k+1} \geq \nu_{k+1}^{\geq k+1} = \mu_{k+2}^{\geq k+1} \geq \cdots \geq \nu_l^{\geq k+1} = \mu_{l+1}^{\geq k+1}$. Thus the blocks $E_{i,i+1}$ ($i = k+1,\ldots,l$) are square before reducing $E_{k,k+1}$. Furthermore, the sequence $\{d_i^{\geq i+1}\}$ of the orders of diagonal blocks in $M_{k+1}$ and $N_{k+1}$ is decreasing.

We now start with the description of the transformations. We have $\nu_k^{\geq k+1} \geq \mu_{k+1}^{\geq k+1}$. If $\nu_k^{\geq k+1} = \mu_{k+1}^{\geq k+1}$, then we can skip step $l-k+1$. Therefore, we now assume $\nu_k^{\geq k+1} > \mu_{k+1}^{\geq k+1}$. The procedure consists of $d_k^{\geq k+1} = \nu_k^{\geq k+1} - \mu_{k+1}^{\geq k+1}$ stages. Below we describe the transformations in stage $q$ with $1 \leq q \leq d_k^{\geq k+1}$. To avoid notational complexity, we shall not use a special notation for the blocks and their dimensions to indicate that we are in stage $q$, unless confusion might arise. Moreover, for the time being we also drop the superscript $\geq k+1$ in the dimension notation. So, from now on we assume we are in the $q$th stage of the reduction process of $E_{k,k+1}$.

Our aim is to annihilate all nonzero elements in the bottom row of the blocks $E_{k,j}$ ($j = k+2,\ldots,l+1$) and $A_{k,j}$ ($j = k+1,\ldots,l+1$) in such a way that the properties of the submatrices already treated remain valid. For simplicity we will only describe the transformations performed on the blocks indicated in

$$\left[\begin{array}{c|c||c|c} A^{(k)} & X & E^{(k)} & X \\ \hline 0 & M_{k+1} & 0 & N_{k+1} \end{array}\right] \tag{3.3.9}$$

instead of those in Figure 13. This will sufficiently illustrate the general idea. Here $A^{(k)}$ and $E^{(k)}$ are submatrices of $\hat{A}^{(k)}$ and $\hat{E}^{(k)}$ specified in Figure 14.

$[A^{(k)} \| E^{(k)}] =$

$$\begin{array}{cccc||cccc|l}
A_{k,k} & A_{k,k+1} & A_{k,k+2} & A_{k,k+3} & 0 & E_{k,k+1} & E_{k,k+2} & E_{k,k+3} & \}\,\nu_k \\
\hline
0 & A_{k+1,k+1} & A_{k+1,k+2} & A_{k+1,k+3} & 0 & 0 & E_{k+1,k+2} & E_{k+1,k+3} & \}\,\nu_{k+1}=\mu_{k+2} \\
\hline
0 & 0 & A_{k+2,k+2} & A_{k+2,k+3} & 0 & 0 & 0 & E_{k+2,k+3} & \}\,\nu_{k+2}=\mu_{k+3} \\
\mu_k & \mu_{k+1} & \mu_{k+2} & \mu_{k+3} & \mu_k & \mu_{k+1} & \mu_{k+2} & \mu_{k+3} &
\end{array}$$

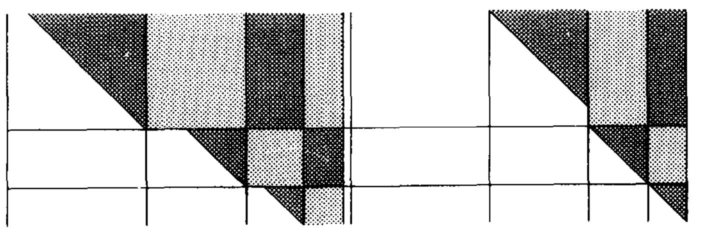



FIG. 14. Some blocks to be transformed when reducing $E_{k,k+1}$.

For the time being, we restrict ourselves to transformations in $[A^{(k)} \| E^{(k)}]$. That is, we take $l = k + 2$. Clearly, we have $k < l - 1$, as is required in step $l - k + 1$. If $k = l - 1$, then the rightmost block column and the bottom block row in $A^{(k)}$ and $E^{(k)}$ are not present. However, the same transformations as described are then carried out on the remaining blocks.

It should be noted that in fact all transformations are performed on the *complete* rows or columns in $A$ and $E$. In Figure 14 we have assumed that both blocks $A_{k+1,k+1}$ and $A_{k+2,k+2}$ are nonsquare ($\mu_{k+1} > \mu_{k+2}$ and $\mu_{k+2} > \mu_{k+3}$). It will turn out that there are $\mu_{k+1} - \mu_{k+2}$ and $\mu_{k+2} - \mu_{k+3}$ extra transformations needed in this case compared to the situation in which these blocks are square.

We start with constructing $\mu_{k+1} - \mu_{k+2}$ column Givens rotations to annihilate the elements $A_{k,k+1}(\nu_k, i)$ $(i = 1, \ldots, \mu_{k+1} - \mu_{k+2})$. The bottom diagonal element of $A_{k,k}$ is used as pivot (see Figure 14). These transformations are also applied to the matrix $E$. This may introduce $\mu_{k+1} - \mu_{k+2}$ nonzero elements in the rightmost column of $E_{k,k}$. Analogously to Lemma 3.3.1 in step 1, we then have that the bottom diagonal element of $A_{k,k}$ and all diagonal elements of $E_{k,k+1}$ are nonzero again (see Figure 15). Notice that when $A_{k+1,k+1}$ is square then $\mu_{k+1} = \mu_{k+2}$. Clearly, the number of transformations is then zero.

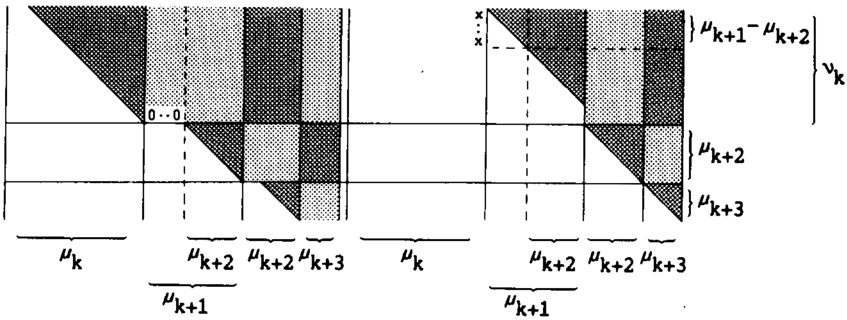



FIG. 15. After the first transformations in case $A_{k+1,k+1}$ is nonsquare.

We now proceed with the transformations to annihilate the remaining $\mu_{k+2}$ elements of $A_{k,k+1}$ and the $\mu_{k+2}$ elements of $E_{k,k+1}$ in the $\nu_k$th row of these blocks. This is done by applying row and column rotations *alternately* to both matrices $E$ and $A$. The nonzero diagonal elements of the upper triangular matrix $E_{k+1,k+2}$ and the nonzero bottom diagonal element of $A_{k,k}$ are successively used as pivot. Each pair of row and column rotations may introduce a possibly nonzero element in the last column of the blocks $A_{k+1,k}$ and $E_{k,k}$, respectively. Clearly, $\mu_{k+2}$ such pairs are needed. This means that after these transformations the last column of $A_{k+1,k}$ may be completely nonzero. Since the last column of $E_{k,k}$ may already have $\mu_{k+1}-\mu_{k+2}$ nonzero elements, it then may contain $\mu_{k+1}$ nonzero elements. Hence, there are still at least $\nu_k-\mu_{k+1}$ zeros in this column. This fact will play an important role for achieving our goal. With respect to the transformations just mentioned we note that they do not disturb the structure of any triangular matrix in $A$ or $E$. Furthermore, they maintain the fully nonzero diagonal property of the triangular matrices involved (see Lemma 3.3.2).

After all transformations described above we have the situation shown in Figure 16. An analogous procedure is used to annihilate all elements in the $\nu_k$th row of the blocks $A_{k,k+2}$ and $E_{k,k+3}$. To be more precise, we start by applying $\mu_{k+2}-\mu_{k+3}$ column Givens rotations to annihilate the first $\mu_{k+2}-\mu_{k+3}$ elements in row $\nu_k$ of $A_{k,k+2}$ using the bottom diagonal element of $A_{k,k}$ as pivot. Obviously, if $A_{k+2,k+3}$ is square (i.e., $\mu_{k+2}=\mu_{k+3}$), then the number of transformations is zero. Note that application of the rotations may introduce nonzero elements in the last column of $E_{k,k}$ as well as of $E_{k+1,k}$, but the last element in column $\mu_k$ of $E_{k,k}$ remains zero, since the bottom row of $E_{k,k+2}$ is zero. Hereafter we alternately apply row and column rotations to annihilate the remaining nonzero elements in the $\nu_k$th rows of $E_{k,k+3}$ and $A_{k,k+2}$. The pivots to be used here are the bottom diagonal of $A_{k,k}$ and the diagonal element of $E_{k+2,k+3}$. Again the transformations do not change the

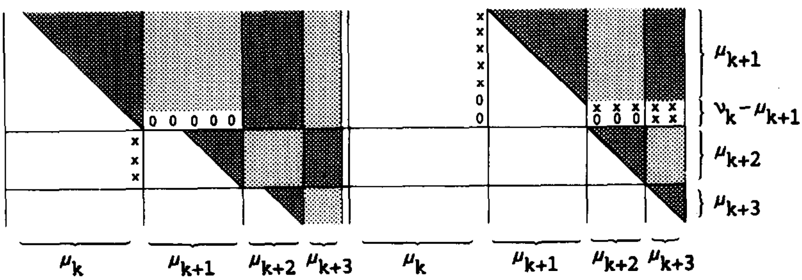


FIG. 16. Situation after transforming $A_{k,k+1}$ and $E_{k,k+2}$.

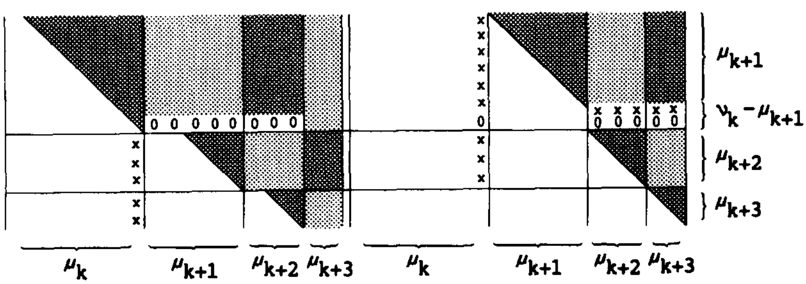


FIG. 17. Situation after transforming $A_{k,k+2}$ and $E_{k,k+3}$.

structure or the fully nonzero diagonal property of the triangular matrices in $A$ and $E$. We find then the situation as indicated in Figure 17. Note that after these transformations column $\mu_k$ of $A_{k+2,k}$ may be nonzero. Since we have taken $l = k+2$, block $E_{k,k+3}$ is in the rightmost block column of $E$. Thus we now have a complete row of zeros in $E$. We still have to create a zero bottom row in $A_{k,k+3}$.

This is done as follows. The nonzero elements in the $\nu_k$th row of $A_{k,k+3}$ are annihilated by column Givens rotations using the bottom diagonal element of $A_{k,k}$ as pivot. Applying these transformations to the matrix $E$ may change all elements in column $\mu_k$ of the blocks $E_{k,k}$, $E_{k+1,k}$ and $E_{k+2,k}$ except the bottom right element of $E_{k,k}$, which remains zero due to the zero row in $E_{k,k+3}$. The structure and the fully nonzero diagonal property of the triangular matrices are again unchanged (see Lemma 3.3.2 below).

We note that up to now we have only transformed blocks in $[A^{(k)} \| E^{(k)}]$. But from now on, the matrices $M_{k+1}$ and $N_{k+1}$ will also be involved when proceeding with the transformations. Therefore, we have also indicated these blocks in Figures 18 and 19.

$$\left[\begin{array}{c|c||c|c} A^{(k)} & X & E^{(k)} & X \\ \hline 0 & M_{k+1} & 0 & N_{k+1} \end{array}\right] =$$

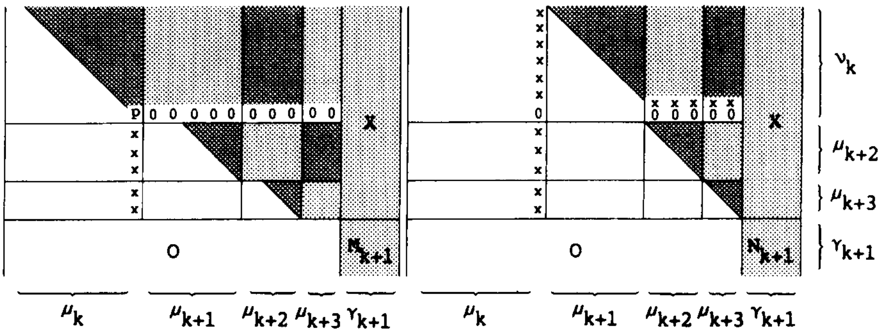



FIG. 18. Situation after the transformation of $A_{k,k+3}$.

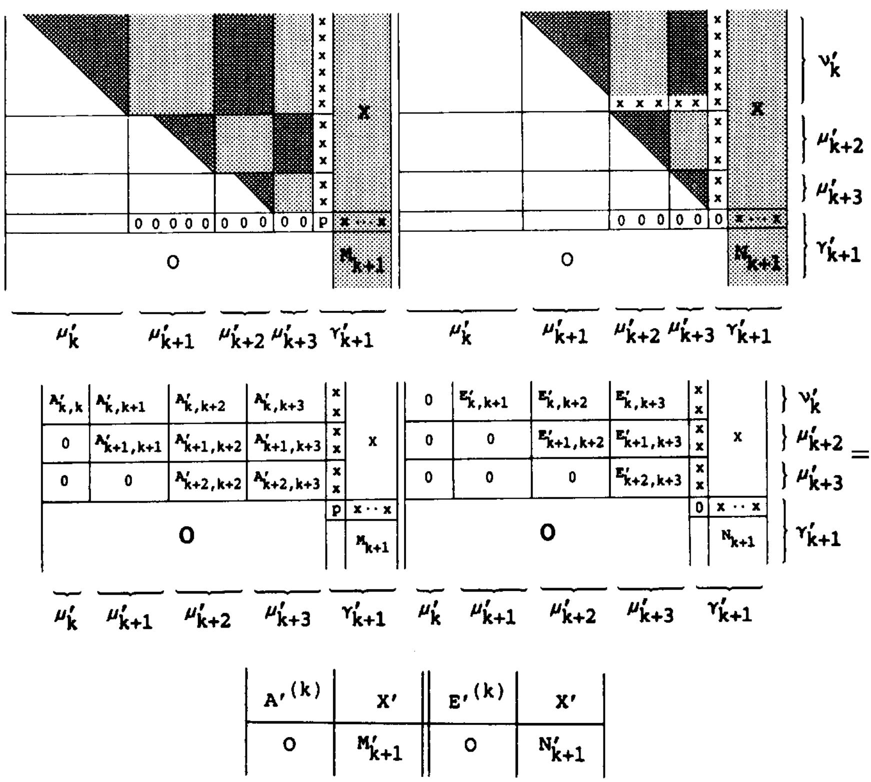



FIG. 19. Final situation after reducing $E_{k,k+1}$ by one row.

At this point we have created a complete row of zeros in $E^{(k)}$ and a row $A^{(k)}$ having all but one element (say $p$) equal to zero (see Figure 18). These two rows in $A$ and $E$ are permuted by cyclic row permutations to the row just above the top row of $M_{k+1}$ and $N_{k+1}$, respectively. Then by cyclic column permutations the column in $A$ containing the element $p$ and its corresponding column in $E$ are permuted to the column just before the leftmost columns of $M_{k+1}$ and $N_{k+1}$, respectively (see Figure 19). Note that thereby the row dimension of all blocks $A_{k,j}$ and $E_{k,j}$ $(j=1,\ldots,l+1)$ as well as the column dimension of the blocks $A_{i,k}$ and $E_{i,k}$ $(i=1,\ldots,l)$ is decreased by one. Furthermore, a new first row and column are added to $M_{k+1}$ and $N_{k+1}$. Here we have

$$M'_{k+1}=\left[\begin{array}{c|c} p & x\cdots x \\ \hline & M_{k+1}\end{array}\right], \qquad N'_{k+1}=\left[\begin{array}{c|c} 0 & x\cdots x \\ \hline & N_{k+1}\end{array}\right], \tag{3.3.10}$$

$$\mu'_k=\mu_k-1, \quad \mu'_j=\mu_j \ (j\neq k), \qquad \nu'_k=\nu_k-1, \quad \nu'_j=\nu_j \ (j\neq k),$$

and

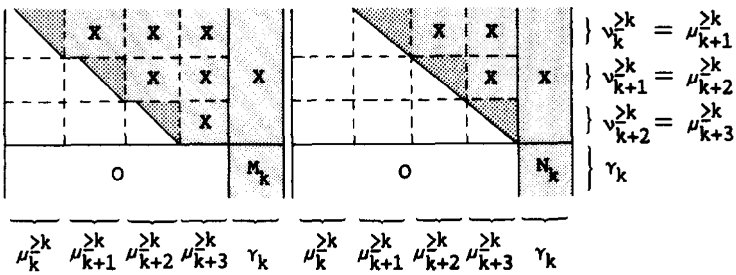
$$\gamma'_{k+1}=\gamma_{k+1}+1. \tag{3.3.11}$$

All blocks $A'_{i,j}$, $E'_{i,j}$, $M'_{k+1}$, and $N'_{k+1}$ are now renamed to their original names. Notice that the redefined matrices have the same structure as before the transformations. Moreover, the upper triangular matrices, including $M_{k+1}$, have the fully nonzero diagonal property again (see Lemma 3.3.2 below). This is the end of stage $q$.

The whole procedure is repeated until the matrix $E_{k,k+1}$ is square. Clearly, the procedure for reducing $E_{k,k+1}$ consists of $\nu_k^{\geq k+1}-\mu_{k+1}^{\geq k+1}$ stages, since $E_{k,k+1}$ has the dimensions $\nu_k^{\geq k+1}\times\mu_{k+1}^{\geq k+1}$ at the start of step $l-k+1$. After we have obtained a square matrix $E_{k,k+1}$, the matrices $M_{k+1}$ and $N_{k+1}$ are renamed to $M_k$ and $N_k$, respectively. Here the description of the transformations in step $l-k+1$ ends.

At the end of the reduction of $E_{k,k+1}$ we then have the structure for $[A^{(k)}\| E^{(k)}]$ given in Figure 20.

Now we recall Theorem 3.3.1 stated at the beginning of the description of step $l-k+1$. In this theorem the general situation before reduction of $E_{k,k+1}$ was given. However, we still have to complete the proof of this theorem, i.e., we have to prove Proposition G$(k-1)$ assuming Proposition G$(k)$ is true. This is done below by the Lemmas 3.3.2 and 3.3.3. For convenience, we recall




*where*

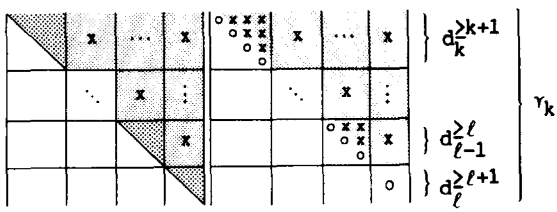
$[M_k \,\|\, N_k] =$ 


FIG. 20. Final situation after reducing $E_{k,k+1}$.

*Proposition* G$(k-1)$ *in Theorem* 3.3.1 $(k>1)$: *Before reducing* $E_{k-1,k}$ (*i.e., after reducing* $E_{k,k+1}$) *we have*:

(1) *the blocks* $E_{i,i+1}$ $(i=k,\dots,l)$ *and* $A_{i,i}$ $(i=k-1,\dots,l)$ *are upper triangular with nonzero diagonal elements*;

(2) *the blocks* $E_{i,i+1}$ $(i=k,\dots,l)$ *are square*;

(3) $M_k$ *is a square upper triangular matrix with a fully nonzero diagonal*;

(4) $N_k$ *is a square upper triangular block matrix with zero diagonal elements*;

(5) *the block dimensions are given by*

$$\nu_i^{\geqslant k} = \nu_i \quad (i \leqslant k-1), \qquad \nu_i^{\geqslant k} = \mu_{i+1}^{\geqslant k} \quad (k \leqslant i \leqslant l),$$

$$\mu_i^{\geqslant k} = \mu_i \quad (i \leqslant k-1), \qquad \mu_i^{\geqslant k} = \mu_i - d_i^{\geqslant i+1} \quad (k \leqslant i \leqslant l+1),$$

*where*

$$d_i^{\geqslant i+1} = \sum_{j=1}^{l} (\nu_j - \mu_{j+1}) \qquad (k-1 \leqslant i \leqslant l)$$

*and*

$$\gamma_k = \sum_{i=k}^{l} (i-k+1)(\nu_i - \mu_{i+1}). \tag{3.3.12}$$

First, the properties of $E_{i,i+1}$ and $A_{i,i}$ are summarized and proved.

LEMMA 3.3.2. *The procedure for the reduction of $E_{k,k+1}$ $(k = 1,\dots, l)$ to square upper triangular form does not disturb the triangular form of the blocks $A_{i,i}$, $E_{i,i+1}$ $(i = 1,\dots, l)$. Moreover, the diagonal elements of these blocks remain nonzero.*

*Proof.* By induction. Obviously, the statement is true for $k = l$ (see step 1).

Now consider the reduction of $E_{k,k+1}$. We leave out the superscript indicating dimensions. The procedure consists of $d_k = \nu_k - \mu_{k+1}$ stages. Clearly, we assume $d_k \geqslant 1$. We consider stage $q$ with $1 \leqslant q \leqslant d_k$. In the sequel we do not indicate this stage explicitly. The induction assumption is that all diagonal elements of $A_{i,i}$ and $E_{i,i+1}$ $(i = 1,\dots, l)$ are nonzero before starting reducing $E_{k,k+1}$. The procedure annihilates all elements in row $\nu_k$ of the blocks $A_{k,j}$ $(j = k+1,\dots, l)$ and $E_{k,j}$ $(j = k+2,\dots, l)$. Without loss of generality we only consider the effects of the procedure on the blocks of $[A^{(k)} \| E^{(k)}]$ in Figure 14 consisting of the blocks $A_{i,j}$ and $E_{i,j}$ $(i = k, k+1;$ $j = k, k+1, k+2)$. All other blocks of $[A^{(k)} \| E^{(k)}]$ involved can be treated analogously. Furthermore we may assume that matrix $A_{k+1,k+1}$ is square, i.e., $\mu_{k+1} = \nu_{k+1}$. (See the discussion after Figure 14.) Thus $\mu_k \geqslant \nu_k > \mu_{k+1} = \nu_{k+1} = \mu_{k+2}$. We have the following propositions $\mathrm{P}(i)$ $(i = 1,\dots, \mu_{k+2})$.

PROPOSITION $\mathrm{P}(i)$. *The situation for the blocks $[A^{(k)} \| E^{(k)}]$ just before annihilating the ith element $x_i$ $(i = 1,\dots, \mu_{k+2})$ in row $\nu_k$ of $E_{k,k+2}$ is given in Figure 21, where the diagonal elements of the triangular blocks are nonzero.*

We prove this proposition by induction. By assumption, $\mathrm{P}(1)$ is true.

Next, suppose $\mathrm{P}(i)$ is true for some $i \geqslant 1$. Hence, the elements $p$, $a_i$, $e_i$, and $t_i$ are all nonzero. We start with constructing a row Givens rotation $G_1$ such that $G_1(x_i, e_i)^T = (0, e_i')$. Clearly, $e_i' \neq 0$. Now $G_1$ is applied to the matrix $A$. We then have

$$M_1 \doteq G_1 \begin{bmatrix} p & y_i \\ 0 & a_i \end{bmatrix} = \begin{bmatrix} p' & y_i' \\ u_i & a_i' \end{bmatrix} \tag{3.3.13}$$

$$\begin{array}{cccccccc} \left[\begin{array}{ccc|ccc} A^{(i-1)}_{k,k} & A^{(i-1)}_{k,k+1} & A^{(i-1)}_{k,k+2} & E^{(i-1)}_{k,k} & E^{(i-1)}_{k,k+1} & E^{(i-1)}_{k,k+2} \\ \hline A^{(i-1)}_{k+1,k} & A^{(i-1)}_{k+1,k+1} & A^{(i-1)}_{k+1,k+2} & 0 & 0 & E^{(i-1)}_{k+1,k+2} \end{array}\right] & \begin{array}{l} \}\,\nu_k \\ \}\,\mu_{k+2} \end{array} & = \\ \begin{array}{cccccc} \underbrace{\qquad}_{\mu_k} & \underbrace{\qquad}_{\mu_{k+1}} & \underbrace{\qquad}_{\mu_{k+2}} & \underbrace{\qquad}_{\nu_k} & \underbrace{\qquad}_{\nu_{k+1}} & \underbrace{\qquad}_{\nu_{k+2}} \end{array} & & \end{array}$$

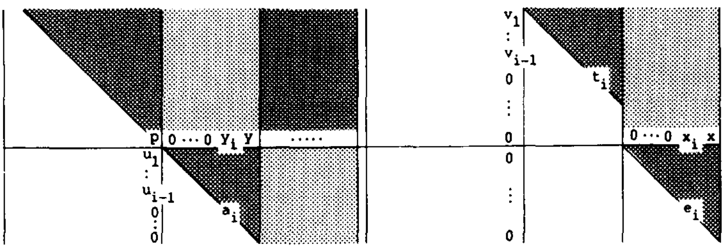



FIG. 21. Intermediate situation when reducing $E_{k,k+1}$.

and rank$(M_1) = 2$, since $p$ and $a_i$ are both nonzero. Moreover, $p'$ and $u_i$ are not both zero. Otherwise, $G_1(p,0)^T = (0,0)^T$; but this is impossible, since $p \neq 0$. Suppose $p' \neq 0$. Then $u_i \neq 0$. Hence

$$G_1 = \pm \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

since $G_1(p,0)^T = (p', u_i)^T = (0, u_i)^T$. Thus $G_1(x_i, e_i)^T = \pm(e_i, -x_i)^T$, which contradicts the construction of $G_1$, since $e_i \neq 0$. So we can conclude $p' \neq 0$.

Now a column Givens rotation $G_2$ is constructed such that $(p', y_i')G_2 = (p'',0)$. Then

$$M_2 \doteq M_1 G_2 = \begin{bmatrix} p' & y_i' \\ u_i' & a_i' \end{bmatrix} G_2 = \begin{bmatrix} p'' & 0 \\ u_i' & a_i'' \end{bmatrix}. \tag{3.3.14}$$

Clearly, $\det(M_1) = \det(M_2)$, i.e. $pa_i = p''a_i''$. Hence $p'' \neq 0$ and $a_i'' \neq 0$.

Applying $G_2$ to the matrix $A$ gives $(0, t_i)G_2 = (v_i, t_i')$. Suppose $t_i' = 0$; then $G_2 = \pm\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ since $G_2$ is a Givens rotation and $t_i \neq 0$. This would imply that $(p', y_i')G_2 = \pm(-y_i', p')$. This contradicts the construction of $G_2$, since $p' \neq 0$. Thus $t_i' \neq 0$. So we have proved that the bottom diagonal element $p$ of $A_{k,k}$ and the diagonal elements $a_i$, $e_i$, and $t_i$ in $A_{k+1,k+1}$, $E_{k+1,k+1}$, and $E_{k,k+1}$ are all nonzero after the transformations $G_1$ and $G_2$, and that possibly nonzero elements $u_i'$ and $v_i$ have been introduced. In other words, P$(i+1)$ is true. We now conclude that all propositions P$(i)$ $(i = 1,\dots,\mu_{k+2})$ are valid. Finally, we note that the blocks $A_{i,i}$ $(i = 1,\dots,k-1)$ and $E_{i,i+1}$ $(i = 1,\dots,k-2)$ are not affected. In $E_{k-1,k}$ only the last column may then be changed, but this column is no longer in $E_{k-1,k}$ after the reduction of $E_{k,k+1}$ by one row. This completes the proof. ■

REMARK 3.3.3. Notice that Lemma 3.3.2 proves statements (i) and (ii) of Proposition G$(k-1)$ [see (3.3.12)]. Furthermore, at the end of the reduction of $E_{k,k+1}$ the matrix $M_{k+1}$ (i.e. the matrix $M_k$ before reducing $E_{k-1,k}$) is square and upper triangular by construction. In each stage of the reduction process of $E_{k,k+1}$, the transformed bottom diagonal element $p''$ of $A_{k,k}$ becomes the top left diagonal element of $M'_{k+1}$ (see Figure 18). By Lemma 3.3.2 we have $p'' \neq 0$. Hence, statement (iii) in (3.3.12) is also true. Furthermore, the correctness of statement (iv) can easily be verified.

Now we shall prove statement (v) in (3.3.12) concerning the dimensions of the transformed matrices. To this end, we note that at the end of stage $q$ when reducing $E_{k,k+1}$ the redefined matrices have a similar structure (but with possibly different dimensions) to that at the beginning of stage $q$. Clearly, this is true for all $q$ with $1 \leqslant q \leqslant d_k^{\geqslant k+1}$. Let us now explicitly indicate the current stage in the dimension notation as follows. The column and row dimension of $E_{k,i}$ (and $A_{k,i}$) at the end of stage $q$ ($q \geqslant 0$) when reducing $E_{k,k+1}$ are denoted by $\mu_i^{\geqslant k+1}(q)$ and $\nu_i^{\geqslant k+1}(q)$, respectively. Furthermore, we define

$$\begin{aligned} &\mu_i^{\geqslant k+1}(0) \doteq \mu_i^{\geqslant k+1}, \quad \mu_i^{\geqslant k} \doteq \mu_i^{\geqslant k+1}\left(d_k^{\geqslant k+1}\right) && k \leqslant i \leqslant l+1, \\ &\nu_i^{\geqslant k+1}(0) \doteq \mu_i^{\geqslant k+1}, \quad \nu_i^{\geqslant k} \doteq \nu_i^{\geqslant k+1}\left(d_k^{\geqslant k+1}\right) && k \leqslant i \leqslant l. \end{aligned} \tag{3.3.15}$$

Using this notation, we can rewrite (3.3.11) as

$$\begin{aligned} &\mu_k^{\geqslant k+1}(q) = \mu_k^{\geqslant k+1}(q-1) - 1, \qquad \mu_i^{\geqslant k+1}(q) = \mu_i^{\geqslant k+1}(q-1), \quad i \neq k, \\ &\nu_k^{\geqslant k+1}(q) = \nu_k^{\geqslant k+1}(q-1) - 1, \qquad \nu_i^{\geqslant k+1}(q) = \nu_i^{\geqslant k+1}(q-1), \quad i \neq k, \\ &\gamma_{k+1}(q) = \gamma_{k+1}(q-1) + 1, \end{aligned} \tag{3.3.16}$$

valid for all $q$ with $1 \leqslant q \leqslant d_k^{\geqslant k+1}$.

We can now formulate

LEMMA 3.3.3. *Assuming Proposition* G$(k)$ *in Theorem* 3.3.1, *the block dimensions in* $A$ *and* $E$ *after reducing* $E_{k,k+1}$ *are given by*

$$\begin{aligned} &\mu_i^{\geqslant k} = \mu_i \quad (i \leqslant k-1), \qquad \mu_i^{\geqslant k} = \mu_i - d^{\geqslant i+1} \qquad (k \leqslant i \leqslant l+1), \\ &\nu_i^{\geqslant k} = \nu_i \quad (i \leqslant k-1), \qquad \nu_i^{\geqslant k} = \mu_{i+1}^{\geqslant k} \qquad (k \leqslant i \leqslant l), \end{aligned}$$

*where*

$$d_i^{\geqslant i+1} = \sum_{j=1}^{l} (\nu_j - \mu_{j+1}) \qquad (k \leqslant i \leqslant l+1).$$

*Furthermore, the order $\gamma_k$ of the matrices $M_k$ and $N_k$ (i.e., the matrices $M_{k+1}$ and $N_{k+1}$ after reducing $E_{k,k+1}$) is given by*

$$\gamma_k = \sum_{i=k}^{l} d_i^{\geqslant i+1} = \sum_{i=k}^{l} (i-k+1)(\nu_i - \mu_{i+1}).$$

*Proof.* See Appendix A. ■

REMARK 3.3.4. It is easily seen that all formulas in statement (v) of Proposition G$(k-1)$ in (3.3.12) except the formula for $d_{k-1}^{\geqslant k}$ are proven by Lemma 3.3.3. However, using some results of Lemma 3.3.3 we find

$$d_{k-1}^{\geqslant k} \doteq \nu_{k-1}^{\geqslant k} - \mu_k^{\geqslant k} = \nu_{k-1} - \left(\mu_k - d_k^{\geqslant k+1}\right) = \sum_{i=k+1}^{l} (\nu_i - \mu_{i+1}).$$

This completes the proof of validity of statement (v) in (3.3.11).

Recalling Lemmas 3.3.2, 3.3.3 and Remarks 3.3.5, 3.3.6, we can now conclude that the proof of Theorem 3.3.1 is complete. ■

*Situation after steps 1 to $l$.* The situation for the pencil $\lambda E_1 - A_1$ after all transformations performed in steps 1 to $l$ is completely described by Proposition G(0) in Theorem 3.3.1. Since the transformations in the above algorithm are all unitary, we have, using Theorem 3.3.1,

$$P_1(\lambda E_{\epsilon\infty} - A_{\epsilon\infty})Q_1 = \left[\begin{array}{c|c} \lambda E_\epsilon - A_\epsilon & X \\ \hline 0 & \lambda E_\infty - A_\infty \end{array}\right],$$

where $\lambda E_\epsilon - A_\epsilon \doteq \lambda \hat{E}^{(0)} - \hat{A}^{(0)}$ and $\lambda E_\infty - A_\infty \doteq \lambda N_1 - M_1$. Clearly the pencil $\lambda E_\infty - A_\infty$ is regular, since $M_1$ is regular. The matrix $E_\infty$ is nilpotent, since $N_1$ is upper triangular with all its diagonal elements zero. Hence we have:

*The pencil $\lambda E_\infty - A_\infty$ only contains infinite elementary divisors.*

As stated in Theorem 3.3.1, all blocks $A_{i,i}$ have full row rank and all $E_{i,i}$ are zero. Hence the pencil $\lambda E_\epsilon - A_\epsilon$ has full row rank for all $\lambda \in \mathbf{C}$. Moreover, $E_\epsilon$ has full row rank, since all blocks $E_{i,i+1}$ are invertible. Hence:

*The pencil $\lambda E_\epsilon - A_\epsilon$ only contains Kronecker column indices.*

Thus the transformations in steps 1 to $l$ separate the Kronecker structure of $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ into two disjunct Kronecker structures of $\lambda E_\epsilon - A_\epsilon$ and $\lambda E_\infty - A_\infty$, respectively. Moreover, the dimensions of the blocks in these two pencils completely determine their Kronecker structure.

*An algorithm for the separation of $\lambda E_\epsilon - A_\epsilon$ and $\lambda E_\infty - A_\infty$ in $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$.* A minor disadvantage of the procedure sketched above is the presence of cyclic row and column permutations. However, the procedure can be modified so that these permutations can be avoided *without any extra computational effort*. To this end, we determine Givens rotations identically as above but when applying them to a pair of rows (or columns), all elements involved are interchanged simultaneously.

Thus, when annihilating the component $x$ of the vector $(x, y)^T$ by Givens rotations, we now determine $c$ and $s$ such that $c^2 + s^2 = 1$ and

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ 0 \end{bmatrix} \quad \text{instead of} \quad \begin{bmatrix} c & -s \\ s & c \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ x' \end{bmatrix}. \tag{3.3.17}$$

Clearly, both transformations require the same number of operations. Notice that now in each step Givens rotations are performed on a pair of *successive* rows or columns. This feature may speed up the implementation of the procedure in a computer program. The algorithm describing the transformations just mentioned is given below.

ALGORITHM 3.3.2 (Separation of $\lambda E_\epsilon - A_\epsilon$ and $\lambda E_\infty - A_\infty$ in $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$).

**comment** Initialization;
$P := I$; $Q := I$;
**comment** Start of the reduction process;
**for** $k = l$ **step** $-1$ **until** 1 **do**
**begin comment** Reduce $E_{k,k+1}$ to square matrix;
  **while** $E_{k,k+1}$ is nonsquare **do**
  **begin**
    **for** $p = k + 1$ **step** 1 **until** $l$ **do**
    **begin**

**comment** Annihilate the elements originally present in the last row of the blocks $E_{k,p+1}$ and $A_{k,p}$. Use original bottom diagonal element of $A_{k,k}$ as pivot. Starting position of pivot in $A$ is $(r_A, c_A)$;
$r_A = \sum_{i=1}^{p-1} \nu_i$; $c_A = \sum_{i=1}^{p-1} \mu_i$;
**for** $j = 1$ **step until** $\mu_p - \mu_{p+1}$ **do**
**begin**
  **comment** $c_j^A$ = current column index of pivot in $A$. Annihilate $A(r_A, c_j^A)$ by applying column rotations $Q^{\pi}_{c_j^A, c_j^A+1}$, also interchanging the elements. Apply the same transformation to $E$ and update $Q$;
  $c_j^A = c_A + j - 1$;
  $A := AQ^{\pi}_{c_j^A, c_j^A+1}$; $E := EQ^{\pi}_{c_j^A, c_j^A+1}$; $Q := QQ^{\pi}_{c_j^A, c_j^A+1}$;
  **result** $A(r_A, c_j^A) = 0$;
**end** $j$-loop;
**comment** Annihilate the remaining elements originally present in the last row of $E_{k,p+1}$ and $A_{k,p}$ by alternately applying row and column rotations also interchanging the elements. Use diagonal elements of $E_{p,p+1}$ and original bottom diagonal elements of $A_{k,k}$ as pivots, respectively. Starting positions of pivots in $A$ and $E$ are $(r_A, c_A)$ and $(r_E, c_E)$;
$r_E = r_A + 1$; $c_E = 1 + \sum_{i=1}^{p} \mu_i$; $c_A = \sum_{i=1}^{p} \mu_i - \mu_{p+1}$;
**for** $j = 1$ **step** 1 **until** $\mu_{p+1}$ **do**
**begin**
  **comment** $(r_j^E, c_j^E)$ = current position of pivot in $E$. Annihilate $E(r_E, c_j^E)$ by applying row rotations $P^{\pi}_{r_j^E, r_j^E-1}$ also interchanging the elements. Apply the same transformation to $A$ and update $P$;
  $r_j^E = r_E + j - 1$; $c_j^E = c_E + j - 1$; $c_j^A = c_A + j - 1$;
  $E := P^{\pi}_{r_j^E, r_j^E-1} E$; $A := P^{\pi}_{r_j^E, r_j^E-1} A$; $P := P^{\pi}_{r_j^E, r_j^E-1} P$;
  **result** $E(r_E, c_j^E) = 0$;
  **comment** Annihilate $A(r_j^E, c_j^A)$ by applying column rotations $Q^{\pi}_{c_j^A, c_j^A+1}$ also interchanging the elements. Apply the same transformation to $E$ and update $Q$;
  $A := AQ^{\pi}_{c_j^A, c_j^A+1}$; $E := EQ^{\pi}_{c_j^A, c_j^A+1}$; $Q := QQ^{\pi}_{c_j^A, c_j^A+1}$;
  **result** $A(r_j^E, c_j^A) = 0$;
**end** $j$-loop
**end** $p$-loop;
**comment** Annihilate the elements originally present in the last row of $A_{k,l+1}$;
$r_A = \sum_{i=1}^{l} \nu_i$; $c_A = \sum_{i=1}^{l} \mu_i$;
**for** $j = 1$ **step** 1 **until** $\mu_{l+1}$ **do**
**begin**

**comment** Annihilate $A(r_A, c_j^A)$ by applying column rotations $Q^{\pi}_{c_j^A, c_j^A+1}$, also interchanging the elements. Apply the same transformation to $E$ and update $Q$;
$c_j^A = c_A + j - 1$;
$A := AQ^{\pi}_{c_j^A, c_j^A+1}$; $E := EQ^{\pi}_{c_j^A, c_j^A+1}$; $Q := QQ^{\pi}_{c_j^A, c_j^A+1}$;
**result** $A(r_A, c_j^A) = 0$;
**end** $j$-loop;
**comment** Reduce $A$ and $E$ by leaving out their last row and rightmost column. Redefine blocks in new $A$ and $E$;
$\nu_k = \nu_k - 1$; $\mu_k = \mu_k - 1$;
**comment** End while clause. Now block $E_{k,k+1}$ is square;
**end** $k$-loop;
**comment** End of Algorithm 3.3.2;

### 3.4. *Numerical Aspects*

In this subsection we look at the numerical aspects of the algorithms developed earlier in this section. We give an operation count of the various algorithms, and we discuss the numerical stability of these algorithms.

3.4.1. *Operation Count for Algorithm 3.2.1.* A Givens transformation acting on a pair of vectors having $q$ elements requires $4q$ flops (neglecting lower order terms; see [22]). Using this, we obtain the following operation count for Algorithm 3.2.1. With respect to step 0 of Algorithm 3.2.1 (reduction to echelon form) we refer to Remark 2.2 of the preliminaries. Let us now consider step $j$ ($j \geqslant 1$) in more detail. Here we have

$$A_j = [\underbrace{A_{j,j}}_{n_j-\rho_j} \mid \underbrace{A_{j,j+1}}_{\rho_j}]\}\, m_j, \qquad E_j = [\underbrace{0}_{n_j-\rho_j} \mid \underbrace{E_{j,j+1}}_{\rho_j}]\}\, m_j \tag{3.4.1}$$

The matrix $A_{j,j}$ is compressed to full row rank $\nu_j$ while keeping $E_j$ in echelon form by Algorithm 3.1.1. In Figure 22 the situation for the pair $(A_j, E_j)$ just before the $k$th stage of this algorithm is indicated.

In the $k$th stage we perform Givens rotations on $m_j - k$ pairs of rows of length $n_j - k + 1$ in $A$ and of length at most $\rho_j$ in $E$. This results in at most $4(m_j - k)(n_j - k + 1 + \rho_j)$ flops. Furthermore, there are at most $\rho_j - 1$ column Givens rotations needed for restoring $E_j^{(k)}$ to echelon form. It should be noted that these column transformations are carried out on the whole matrices $E$ and $A$. For the column transformations in $E$ and $A$, we need at

$[\, A_j^{(k)} \;||\; E_j^{(k)} \,] =$

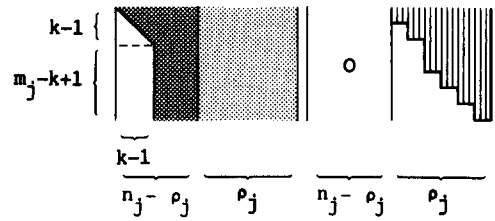



FIG. 22. At the start of stage $k$ in Algorithm 3.1.1.

most

$$4\sum_{l=1}^{\rho_j-1}(m-l+1) \quad \text{and} \quad 4(\rho_j-1)m \tag{3.4.2}$$

flops, respectively. With $\nu_j = \operatorname{rank}(A_{j,j})$, the algorithm takes $\nu_j$ stages for transforming $(A_j, E_j)$. Thus, the overall number $f_j$ of flops needed for transformation of $(A_j, E_j)$ is less than

$$f_j = 4\sum_{k=1}^{\nu_j}\left\{(m_j-k)(n_j-k+1+\rho_j)+\sum_{l=1}^{\rho_j-1}(m-l+1)+m(\rho_j-1)\right\}. \tag{3.4.3}$$

We now have (using $m_j \leqslant m,\ n_j \leqslant n,\ \nu_j \leqslant m,\ \rho_j \leqslant n_j \leqslant n$)

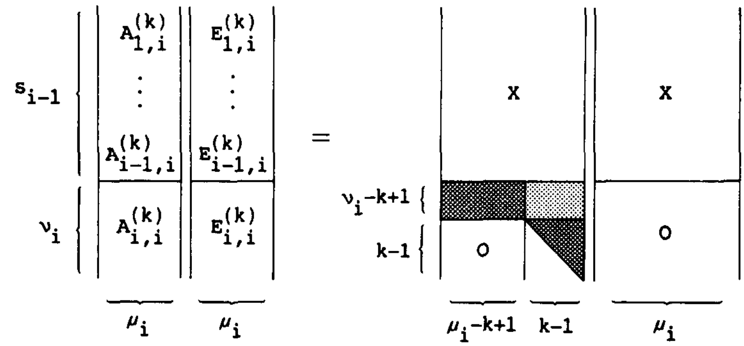
$$f_j \leqslant 4\sum_{k=1}^{\nu_j}\left\{2m_jn_j+2m(\rho_j-1)\right\} \leqslant 16\nu_j mn. \tag{3.4.4}$$

So, the reduction of an $m \times n$ pencil $\lambda E - A$ to the form (3.2.8) using Algorithm 3.2.1 takes at most

$$\sum_j f_j \leqslant 16mn\sum_j \nu_j = 16m_{\epsilon\infty}mn \tag{3.4.5}$$

flops, where $m_{\epsilon\infty}$ is the number of rows of the subpencil $\lambda E_{\epsilon\infty} - A_{\epsilon\infty}$ which was "deflated" by this algorithm.

3.4.2. *Operation Count for Algorithm 3.3.1.* Consider step $i$ of this algorithm in more detail. We first assume that Givens rotations are used. In this step $A_{i,i}$ is reduced to upper triangular form using the $QR$ decomposition. At the beginning of the $k$th stage in the reduction of $A_{i,i}$ we have the following situation for the blocks $A_{j,i}$ and $E_{j,i}$ $(j=1,\dots,i)$:




(3.4.6)

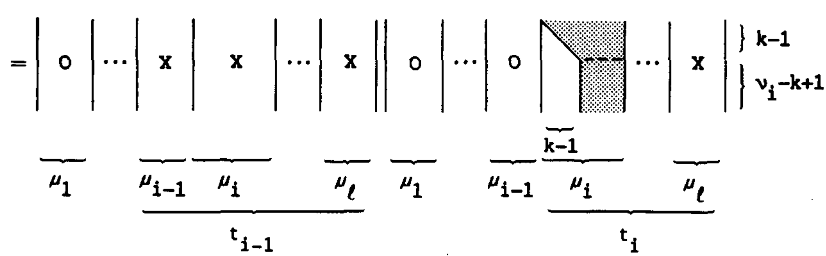
where $s_{i-1}=\sum_{j=1}^{i-1}\nu_j$. In $A_{i,i}^{(k)}$ we now perform rotations on $\mu_i-k$ pairs of columns of length $\nu_i-k+1$. The column transformations needed for reducing $A_{i,i}$ are also applied to the blocks $A_{j,i}$ and $E_{j,i}$ $(j+1,\dots,i-1)$. Therefore, reduction of $A_{i,i}$ plus updating the blocks in $A$ and $E$ requires

$$f_1(i)=4\sum_{k=1}^{\nu_i}(\mu_i-k)(\nu_i-k+1+2s_{i-1})$$

$$\leqslant 4\sum_{k=1}^{\nu_i}\mu_i(\nu_i+2s_{i-1})\leqslant 4\sum_{k=1}^{\nu_i}2n_{\epsilon\infty}m_{\epsilon\infty}\leqslant 8\nu_i n_{\epsilon\infty}m_{\epsilon\infty} \quad (3.4.7)$$

flops, where $m_{\epsilon\infty}$ and $n_{\epsilon\infty}$ are the dimensions of the pencil $\lambda E-A$ $(\doteq\lambda E_{\epsilon\infty}-A_{\epsilon\infty})$. The transformed matrix $E_{i-1,i}$ (which has still full column rank) is then reduced by row transformations using a $QR$ decomposition. We have the following situation for the blocks $E_{i-1,j}$ and $A_{i-1,j}$ $(j=1,\dots,l)$:

$$\left[A_{i-1,1}^{(k)}\cdots A_{i-1,i-1}^{(k)}\cdots A_{i-1,l}^{(k)}\;\|\;E_{i-1,1}^{(k)}\cdots E_{i-1,i-1}^{(k)}\cdots E_{i-1,l}^{(k)}\right]$$




(3.4.8)

where $t_i = \sum_{j=i}^{l} \mu_j$. Here the blocks $A_{i-1,j}$ $(j < i-1)$ and $E_{i-1,j}$ $(j \leqslant i-1)$ are zero. The row transformations needed for the reduction of $E_{i-1,i}$ are also applied to the blocks $E_{i-1,j}$ $(j = i,\dots,l)$ and $A_{i-1,j}$ $(j = i-1,\dots,l)$, which requires

$$f_2(i) = 4 \sum_{k=1}^{\mu_i} (\nu_{i-1} - k)(\mu_i - k + 1 + t_{i+1} + t_{i-1})$$

$$\leqslant 4 \sum_{k=1}^{\mu_i} 2m_{\epsilon\infty}n_{\epsilon\infty} = 8\mu_i m_{\epsilon\infty}n_{\epsilon\infty} \tag{3.4.9}$$

flops (since $\nu_{i-1} \leqslant m_{\epsilon\infty}$, $\mu_i \leqslant n_{\epsilon\infty}$, $t_{i+1} \leqslant n_{\epsilon\infty}$). Algorithm 3.3.1 thus requires less than

$$\sum_{i=1}^{l} \{ f_1(i) + f_2(i) \} \leqslant 8m_{\epsilon\infty}n_{\epsilon\infty} \sum_{i=1}^{l} (\nu_i + \mu_i) = 8(m_{\epsilon\infty} + n_{\epsilon\infty})m_{\epsilon\infty}n_{\epsilon\infty} \tag{3.4.10}$$

flops. It is obvious that this is a very generous upper bound. If instead of Givens transformations one had used Householder transformations, the operation count would be divided by 2.

*3.4.3. Operation Count for Algorithm 3.3.2.* Because this algorithm is very involved, we prefer here to give an operation count per *loop* in the algorithm. We recall that the dimension of the pencil to be transformed is $m_{\epsilon\infty} \times n_{\epsilon\infty}$.

First, consider the two inner *j*-loops in the loop (**for** $p = k+1$ **step** 1 **until** $l$):

(1) *Loop* (**for** $j = 1$ **step** 1 **until** $\mu_p - \mu_{p+1}$). The rotations in $A$ and $E$ are applied to columns of length $r_A$ (row index range $[1 : r_A]$). Hence, the number of flops is less than

$$2 \times 4(\mu_p - \mu_{p+1})r_A \leqslant 8(\mu_p - \mu_{p+1})m_{\epsilon\infty}. \tag{3.4.11}$$

(2) *Loop* (**for** $j = 1$ **step** 1 **until** $\mu_{p+1}$). For each $j$ the row transformations in $E$ and $A$ are applied to the rows $r_j^E$ and $r_j^E - 1$. The corresponding column index ranges in $E$ and $A$ are $[c_j^E : n_{\epsilon\infty}]$ and $[c_j^A : n_{\epsilon\infty}]$, respectively. Thus with $c_j^E = c_E + j - 1$ and $c_j^A = c_A + j - 1$ we find that the number of

flops is less than

$$4[n_{\epsilon\infty} - c_E + 1 - (j-1)] + 4[n_{\epsilon\infty} - c_A + 1 - (j-1)]. \tag{3.4.12}$$

For each value of $j$, column transformations in $A$ and $E$ are applied to the columns $c_j^A$ and $c_j^A + 1$ with row index range $[1: r_j^E]$. Hence, with $r_j^E = r_E + j - 1$ we need less than

$$2 \times 4(r_E + j - 1) \tag{3.4.13}$$

flops. Thus, at the end of this loop we have needed less than (using $\nu_i \leqslant \mu_i$ and $\mu_p \geqslant \mu_{p+1} \geqslant 1$)

$$\begin{aligned}
&4\mu_{p+1}\{(n_{\epsilon\infty} - c_E + 1) + (n_{\epsilon\infty} - c_A) + 1 + 2r_E\} \\
&= 4\mu_{p+1}\left\{\sum_{i=p+1}^{l+1} \mu_i + \sum_{i=p+1}^{l+1} \mu_i + \mu_{p+1} + 3 + 2\sum_{i=1}^{p-1} \nu_i\right\} \\
&\leqslant 8\mu_{p+1}\left\{\sum_{i=p+1}^{l+1} \mu_i + \sum_{i=1}^{p-1} \mu_i\right\} + 4\mu_{p+1}(\mu_{p+1} + 3) \\
&= 8\mu_{p+1}(n_{\epsilon\infty} - \mu_p) + 4\mu_{p+1}(\mu_{p+1} + 3) \\
&\leqslant 8\mu_{p+1}n_{\epsilon\infty} + 12\mu_{p+1} - 4\mu_{p+1}^2 \leqslant 8\mu_{p+1}(n_{\epsilon\infty} + 1).
\end{aligned} \tag{3.4.14}$$

Consequently, after the $p$-loop we find, using (3.4.11), (3.4.14), and $m_{\epsilon\infty} \geqslant 1$,

$$\begin{aligned}
f_p &\leqslant \sum_{p=k+1}^{l} \{8(\mu_p - \mu_{p+1})m_{\epsilon\infty} + 8\mu_{p+1}(n_{\epsilon\infty} + 1)\} \\
&\leqslant 8 \sum_{p=k+1}^{l} \{m_{\epsilon\infty}\mu_p + n_{\epsilon\infty}\mu_{p+1}\}.
\end{aligned} \tag{3.4.15}$$

(3) *Loop* **(for $j = 1$ step 1 until $\mu_{l+1}$).** Here for each $j$, column transformations for $A$ and $E$ are applied to the columns $c_j^A$ and $c_j^A + 1$ with row index range $[1: r_A]$. Hence, for this $j$-loop we need less than

$$f_j \leqslant \mu_{l+1} 8 r_A \leqslant 8 m_{\epsilon\infty} \mu_{l+1} \tag{3.4.16}$$

flops. Combining (3.4.15) and (3.4.16), we find that eliminating *one* row in $E_{k,k+1}$ requires

$$f_p + f_j \leq 8 \sum_{p=k+1}^{l} \{ m_{\epsilon\infty}\mu_p + n_{\epsilon\infty}\mu_{p+1} \} + 8m_{\epsilon\infty}\mu_{l+1}$$

$$= 8m_{\epsilon\infty} \sum_{p=k+1}^{l+1} \mu_p + 8n_{\epsilon\infty} \sum_{p=k+2}^{l+1} \mu_p \leq 8n_{\epsilon\infty}(m_{\epsilon\infty} + n_{\epsilon\infty}). \quad (3.4.17)$$

(4) *Loop* (**for** $k = l$ **step** $-1$ **until** 1). The reduction of $E_{k,k+1}$ to a square matrix (i.e. the **while** loop) consists of eliminating $\nu_k - \mu_{k+1}$ rows. Using (3.4.17), we thus need less than

$$f_{\textbf{while}} \leq 8(\nu_k - \mu_{k+1})n_{\epsilon\infty}(m_{\epsilon\infty} + n_{\epsilon\infty}) \quad (3.4.18)$$

flops.

Finally, we conclude that the operation count for the whole algorithm is

$$\sum_{k=1}^{l} \{ 8(\nu_k - \mu_{k+1})n_{\epsilon\infty}(m_{\epsilon\infty} + n_{\epsilon\infty}) \} \leq 8(m_{\epsilon\infty}^2 n_{\epsilon\infty} + m_{\epsilon\infty} n_{\epsilon\infty}^2). \quad (3.4.19)$$

*3.4.4. Numerical Stability.* An important property of Algorithm 3.2.1 is its backward stability. For the transformations performed in Step $j$ the following result can be proved (see [22]). In the presence of rounding errors, we have for the *computed* matrices in step $j$

$$\left[\begin{array}{c|c||c|c} A_{j,j} & A_{j,j+1} & 0 & E_{j,j+1} \\ \hline 0 & A_{j+1} & 0 & E_{j+1} \end{array}\right] = \tilde{P}_j [\tilde{A}_j \| \tilde{E}_j] \tilde{Q}_j, \quad (3.4.20)$$

where $\tilde{P}_j$ and $\tilde{Q}_j$ are still unitary. Let $[A_j | E_j]$ be the matrix pair at start of step $j$, and let $\tilde{P}_j'$ and $\tilde{Q}_j'$ be the computed transformation matrices in step $j$. If $\epsilon$ is the machine precision of the computer and a threshold $\delta$ of the order of $\epsilon$ is used, then

$$\| [A_j | E_j] - [\tilde{A}_j | \tilde{E}_j] \|_E \leq \Pi_{1,j}\delta \| [A_j | E_j] \|_E,$$

$$\| \tilde{P}_j' - \tilde{P}_j \|_E \leq \Pi_{2,j}\delta, \qquad \| \tilde{Q}_j' - \tilde{Q}_j \|_E \leq \Pi_{3,j}\delta, \quad (3.4.21)$$

where $\Pi_{1,j}$, $\Pi_{2,j}$ and $\Pi_{3,j}$ are polynomial expressions in the dimensions of the corresponding matrices. Let $\lambda E' - A'$ be the computed pencil obtained at the end of Algorithm 3.2.1. The final computed matrices $P'$ and $Q'$ are the products of computed Givens rotation matrices. With respect to the accumulation of rounding errors in $\lambda E' - A'$, $P'$, and $Q'$ we can say that there exist a pencil $\lambda \hat{E} - \hat{A}$ and unitary matrices $\hat{P}$ and $\hat{Q}$ such that

$$\|[E' \mid A'] - [\hat{E} \mid \hat{A}]\|_E \leqslant \Pi_1 \delta \|[E' \mid A']\|_E,$$

$$\|P' - \hat{P}\|_E \leqslant \Pi_2 \delta, \qquad \|Q' - \hat{Q}\|_E \leqslant \Pi_3 \delta, \tag{3.4.22}$$

where $\Pi_1$, $\Pi_2$, and $\Pi_3$ are constants depending on the dimensions of the corresponding matrices. In other words, the computed transformation matrices $P'$ and $Q'$ are nearly unitary, and the computed pencil $\lambda E' - A'$ can be seen as the exact result when applying the algorithm to a slightly perturbed pencil $\lambda \hat{E} - \hat{A}$.

Let us now discuss the numerical stability and the efficiency of the other algorithms of this paper. It is clear that the same type of unitary transformations are used as in Algorithm 3.2.1. Therefore, using the same arguments it is clear that Algorithms 3.3.1 and 3.3.2 are numerically backward stable as well.

## 4. CONCLUDING REMARKS

In this paper we have given a new method to compute the generalized Schur form of a singular pencil which has a complexity that is an order of magnitude lower than earlier methods based on orthogonal transformations [18, 8]. This was achieved by using a technique inspired by similar algorithms where some kind of condensed form was preserved during subsequent steps of the algorithm [15, 4].

The obtained generalized Schur form and the corresponding algorithm to compute it are particularly relevant to the area of systems and control theory where several applications and/or variants can be discerned. We name here a few:

(1) *Staircase form of a generalized state space system* (*GSSM*). For a GSSM $\{\lambda E - A, B, C, D\}$, one is looking for a new coordinate system $\{\lambda E_t - A_t, B_t, C_t, D_t\} = \{Q(\lambda E - A)U, QBV, CU, DV\}$, where $Q$, $U$, and $V$

are unitary and such that:

$$Q\cdot[B|\lambda E-A].\begin{bmatrix} U & 0\\ 0 & V\end{bmatrix}$$

$$=\left[\begin{array}{c|cccccc} B_1 & \lambda E_1-A_1 & X & X & \cdots & X & X\\ 0 & B_2 & \lambda E_2-A_2 & X & \cdots & X & X\\ 0 & 0 & B_3 & \lambda E_3-A_3 & \ddots & \vdots & \vdots\\ \vdots & \vdots & \ddots & \ddots & \ddots & X & X\\ 0 & 0 & \cdots & 0 & B_k & \lambda E_k-A_k & X\\ 0 & 0 & \cdots & 0 & 0 & 0 & \lambda E_{k+1}-A_{k+1}\end{array}\right]. \tag{4.1}$$

Here the $B_i$ ($i=1,\ldots,k$) have full row rank $r_i$ and the $E_i$ ($i=2,\ldots,k$) have full column rank $c_i$. This form displays the controllable subspace of the GSSM and is of crucial importance in several applications. Dual forms can of course also be obtained for the pencil $\left[\begin{array}{c}\lambda E-A\\ \hline C\end{array}\right]$ [18, 19]. In principle these forms can be computed with the algorithms described in this paper provided some minor modification are made (see [2]).

(2) *Computing the zeros of a GSSM.* The pencil

$$S(\lambda)=\left[\begin{array}{c|c} B & \lambda E-A\\ \hline D & -C\end{array}\right] \tag{4.2}$$

is usually called the system matrix of a GSSM $\{\lambda E-A,B,C,D\}$, and its finite generalized eigenvalues are the *transmission zeros* of the system. In [4], an algorithm is given to derive the following decomposition of $S(\lambda)$ (for $E=I$):

$$\left[\begin{array}{c|c} Q_1 & Q_2\\ \hline Q_3 & Q_4\end{array}\right]\left[\begin{array}{c|c} B & \lambda E-A\\ \hline D & -C\end{array}\right]\begin{bmatrix} V & 0\\ 0 & U\end{bmatrix}$$

$$=\begin{bmatrix} R_1 & & & & & & X\\ & \ddots & & & & & \\ & & R_s & & & & \\ & & & \lambda E_f-A_f & & & \\ & & & & L_t & & \\ & 0 & & & & \ddots & \\ & & & & & & L_1\end{bmatrix}, \tag{4.3}$$

where the $R_i$ $(i=1,\dots,s)$ have full row rank and the $L_i$ $(i=1,\dots,t)$ have full column rank. The algorithm could very well be adapted to general $E$ using the principles explained in this paper. This would then be an alternative algorithm to compute the finite eigenvalues of a general pencil (see [2]).

(3) *Deadbeat control of a GSSM.* For deadbeat control of a pair $(\lambda E - A, B)$ one has to construct a state feedback $F$ such that the eigenvalues of the pencil $\lambda E-(A+BF)$ are all at $\lambda=0$, which of course implies that $E$ is regular and $(E^{-1}A, E^{-1}B)$ controllable [16]. In this case the form (4.1) has square $E_i$ matrices and a vanishing $\lambda E_{k+1}-A_{k+1}$ pencil. Deadbeat control could then be performed directly on the pair $(E^{-1}A, E^{-1}B)$, and an algorithm for this is developed in [20] where the minimum norm feedback solution $F$ is constructed in a recursive manner. But the inversion of $E$ should be avoided if possible for numerical reasons. Using the above form (4.1) a recursive algorithm can be derived which constructs unitary transformations $Q$ and $U$ and a feedback matrix $F$ such that

$$Q\cdot[\lambda E-(A+BF)]\cdot U=\begin{bmatrix}\lambda E_1' & \cdots & X\\ & \ddots & \vdots\\ 0 & & \lambda E_k'\end{bmatrix}, \tag{4.4}$$

where the $E_i'$ matrices are square and invertible. The matrices $U$ and $F$ can be shown to be the same as those derived by the method described in [20] when applied to $(E^{-1}A, E^{-1}B)$, and therefore we also find here the minimum norm solution $F$ (see [2]).

(4) *Reduced observer of a GSSM.* Assume we have a system $\{\lambda E - A, B, C, 0\}$ with $C$ having full row rank $p$ and $(\lambda E - A, C)$ observable (i.e. $\left[\begin{array}{c}\lambda E - A\\ \hline C\end{array}\right]$ full column rank for all finite $\lambda$, and $\left[\begin{array}{c}E\\ \hline C\end{array}\right]$ full column rank [21]). We then want to construct a *reduced observer* [14, 20] having the form

$$\lambda Sz = Fz + Pu + Dy \tag{4.5}$$

which uses the input $u$ and output $y$ of the system $\{\lambda E - A, B, C, 0\}$ to reconstruct its state $x$ when the initial state is unknown. In [21] it is shown that the controllability condition implies that then the matrices $E_i$ $(i=2,\dots,k)$ in (4.1) are square invertible and the pencil $\lambda E_{k+1}-A_{k+1}$ vanishes. A recursive solution is then derived to solve this problem by passing via the (generalized) Sylvester equation in $X$:

$$SXA - FXE = DC \tag{4.6}$$

with the constraint that $\left[\begin{array}{c} XE \\ \hline C \end{array}\right]$ is invertible. This is a normal generalization of the equation used by Luenberger for standard state space models [14]. The new method uses the staircase form (4.1) to yield a recursive algorithm for constructing the solution.

(5) *Embedding a polynomial matrix into a unimodular one.* Given a $p \times n$ polynomial matrix $P(\lambda)$ $(p < n)$ which has full row rank $p$ for all finite $\lambda$, one wants to find an *embedding* $\begin{bmatrix} P(\lambda) \\ Q(\lambda) \end{bmatrix}$ which is *unimodular*, i.e. which has nonzero determinant independent of $\lambda$. In [3] it is shown that this can be reduced by a straightforward technique to a similar problem but where now $P(\lambda)$ is replaced by a pencil $[\lambda E - A \,|\, B]$ with $[A \,|\, B]$ full row rank but $E$ possibly singular. Using the staircase form (4.1) of this pencil, one then finds that $\lambda E_{k+1} - A_{k+1}$ is already unimodular, while the remaining part is easily embedded by adding a number of rows of the type

$$\begin{bmatrix} 0 & \cdots & 0 & C_i & X & \cdots & X \end{bmatrix} \tag{4.7}$$

to each corresponding block row

$$\begin{bmatrix} 0 & \cdots & 0 & B_i & \lambda E_i - A_i & X & \cdots & X \end{bmatrix}. \tag{4.8}$$

Using this solution, one then easily works back to the solution for the polynomial $P(\lambda)$. As a by-product one also derives solutions for the null space of $P(\lambda)$ and its (right) generalized inverse (see [2]).

## APPENDIX A. PROOF OF LEMMA 3.3.3

*Proof.* By repeatedly applying (3.3.16) and using the definitions in (3.3.15) we find

$$\mu_k^{\geqslant k} \doteq \mu_k^{\geqslant k+1}\left(d_k^{\geqslant k+1}\right) = \mu_k^{\geqslant k+1} - d_k^{\geqslant k+1}, \tag{A-1}$$

$$\mu_i^{\geqslant k} \doteq \mu_i^{\geqslant k+1}\left(d_k^{\geqslant k+1}\right) = \mu_i^{\geqslant k+1}(0) = \mu_i^{\geqslant k+1} \qquad (i \neq k). \tag{A-2}$$

Analogously,

$$\nu_k^{\geqslant k} \doteq \nu_k^{\geqslant k+1}\left(d_k^{\geqslant k+1}\right) = \nu_k^{\geqslant k+1} - d_k^{\geqslant k+1} = \mu_{k+1}^{\geqslant k+1}, \tag{A-3}$$

$$\nu_i^{\geqslant k} \doteq \nu_i^{\geqslant k+1}\left(d_k^{\geqslant k+1}\right) = \nu_i^{\geqslant k+1}(0) = \nu_i^{\geqslant k+1} \qquad (i \neq k). \tag{A-4}$$

(The last equality in (A-3) follows by definition of $d_k^{\geqslant k+1}$.)

Since the block columns and rows with index $i \leqslant k-1$ are not reduced when transforming $E_{k,k+1}$, we have

$$\mu_i^{\geqslant k} = \mu_i^{\geqslant k+1}, \quad \nu_i^{\geqslant k} = \nu_i^{\geqslant k+1} \qquad (i \leqslant k-1). \tag{A-5}$$

By assumption, the formulas stated in Proposition G($k$) in Theorem 3.3.1 are correct. Combination of (A-5) and Proposition G($k$) gives

$$\mu_i^{\geqslant k} = \mu_i, \quad \nu_i^{\geqslant k} = \nu_i \qquad (i \leqslant k-1). \tag{A-6}$$

It follows from (A-1), (A-2), and Proposition G($k$) that

$$\mu_i^{\geqslant k} = \mu_i - d_i^{\geqslant i+1} \qquad (i \geqslant k). \tag{A-7}$$

Using (A-4), Proposition G($k$) and (A-2), we find

$$\nu_i^{\geqslant k} = \nu_i^{\geqslant k+1} = \mu_{i+1}^{\geqslant k+1} = \mu_{i+1}^{\geqslant k} \qquad (i \geqslant k+1). \tag{A-8}$$

Combination of (A-3) and (A-2) gives

$$\nu_k^{\geqslant k} = \mu_{k+1}^{\geqslant k+1} = \mu_{k+1}^{\geqslant k} \tag{A-9}$$

[Notice that Equation (A-8) is also true for $i = k$.]

Finally, let us prove the formula for $\gamma_k$. By construction we have

$$\gamma_k = \gamma_{k+1} + d_k^{\geqslant k+1} \tag{A-10}$$

where $\gamma_k$ denotes the order of the matrix $M_{k+1}$ (and $N_{k+1}$) before reducing $E_{k,k+1}$. Combination of (A-10) and Proposition G($k$) gives

$$\gamma_k = \sum_{i=k+1}^{l} d_i^{\geqslant i+1} + d_k^{\geqslant k+1} = \sum_{i=k}^{l} d_i^{\geqslant i+1} = \sum_{i=k}^{l} \sum_{j=i}^{l} (\nu_j - \mu_{j+1})$$

$$= \sum_{j=k}^{l} (j-k+1)(\nu_j - \mu_{j+1}). \tag{A-11}$$

This completes the proof. ∎

REFERENCES


1 T. Beelen and P. Van Dooren, A generalized state-space approach for embedding a polynomial matrix into a unimodular matrix, in *Proceedings of the SIAM Conference on Linear Algebra Signals, Systems and Control*, Boston, 1986.

2 T. Beelen, P. Van Dooren, and M. Verhaegen, A class of fast staircase algorithms for generalized state-space systems, in *Proceedings of the American Control Conference*, Seattle, Wash., 1986.

3 R. Eising, Polynomial matrices and feedback, *IEEE Trans. Automat. Control.* AC-30:1022–1025 (1985).

4 A. Emami and P. Van Dooren, Computation of zeros of linear multivariable systems, *Automatica* 18:415–430 (1982).

5 F. R. Gantmacher, *Theory of Matrices*, Vols. I, II, Chelsea, New York, 1959.

6 G. H. Golub and C. F. Van Loan, *Matrix Computations*, Oxford, 1983.

7 D. Jordan and L. F. Godbout, On the computation of the canonical pencil of a linear system, *IEEE Trans. Automat. Control* AC-22:112–114 (1977).

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

9 B. Kågström and A. Ruhe, *Matrix Pencils*, Proceedings, Pite Havsbad, 1982, Lecture notes in Mathematics 973, Springer, Berlin, 1983.

10 T. Kailath, *Linear Systems*, Prentice-Hall, 1980.

11 L. Kronecker, Algebraische Reduction der Schaaren bilinear Formen, *Sitzungsber. Akad. Berlin*, 1890, pp. 763–776.

12 V. N. Kublanosvskaya, *AB*-Algorithm and its modifications for the spectral problems of linear pencils of matrices, *Numer. Math.* 43:329–342 (1984).

13 C. L. Lawson and R. J. Hanson, *Solving least squares problems*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

14 D. Luenberger, Observers for multivariable systems, *IEEE Trans. Automat. Control* AC-11:190–197 (1966).

15 C. Moler and G. Stewart, An algorithm for the generalized matrix eigenvalue problem, *SIAM J. Numer. Anal.* 10:241–256 (1973).

16 C. C. Paige, Properties of numerical algorithms related to computing controllability, *IEEE Trans. Automat. Contr.* AC-26:130–138 (1981).

17 G. W. Stewart, *Introduction to Matrix Computations*, Academic, New York, 1973.

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

19 P. Van Dooren, The generalized eigenstructure problem in linear system theory, *IEEE Trans. Automat. Control* AC-26:111–129 (1981).

20 P. Van Dooren, Deadbeat control: A special inverse eigenvalue problem, *BIT* 24:681–699 (1984).

21 M. Verhaegen and P. Van Dooren, A reduced observer for descriptor systems, *Systems Control Lett.* 8:29–37 (1986).

22 J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, 1965.

23 J. H. Wilkinson, Linear differential equations and Kronecker's canonical form, in *Recent Advances in Numerical Analysis* (C. de Boor and G. Golub, Eds.), Academic, New York, 1970, pp. 231–265.
24 J. H. Wilkinson, Kronecker's canonical form and the $QZ$ algorithm, *Linear Algebra Appl.* 28:285–303 (1979).

*Received June 1987; revised 29 July 1987*