A compound matrix algorithm for the computation of the Smith form of a polynomial matrix

M. Mitrouli and G. Kalogeropoulos

Department of Mathematics, University of Athens, Panepistimiopolis, GR-157 84 Athens, Greece
E-mail: mmitroul@atlas.uoa.ariadne-t.gr

Received 6 July 1992; revised 2 November 1993
Communicated by W.M. Coughran, Jr.

In the present paper is presented a numerical method for the exact reduction of a single-variable polynomial matrix to its Smith form without finding roots and without applying unimodular transformations. Using the notion of compound matrices, the Smith canonical form of a polynomial matrix $M(s) \in \mathbb{R}^{n\times n}[s]$ is calculated directly from its definition, requiring only the construction of all the p-compound matrices $C_p(M(s))$ of $M(s)$, $1 < p \leq n$. This technique produces a stable and accurate numerical algorithm working satisfactorily for any polynomial matrix of any degree.

Keywords: Smith canonical form, compound matrices, matrix algorithms, MATLAB.

1. Introduction

In the last few years, the numerical literature has started to show interest in the computation of the canonical forms of pencils and polynomial matrices in general because of their relevance in several applications and stable algorithms have been developed [1–3,5].

When the matrix pencil approach or polynomial matrices are used for the study of linear time-invariant control systems, the Smith canonical form of the corresponding system matrix is extremely useful for the computation of the sets of finite and infinite elementary divisors.

The eigenstructure problem of a polynomial matrix $M(s) \in \mathbb{R}^{n\times n}[s]$ is reduced to the eigenstructure problem of a simpler (diagonal) polynomial matrix $S(s)$ that forms the Smith canonical form of $M(s)$. Under eigenstructure, we understand here all the invariants of the matrix $M(s)$ under equivalence transformations. The eigenstructure of $M(s)$ is retrieved in its Smith canonical form $S(s)$.

For example, consider the polynomial matrix

$$M(s) = \begin{bmatrix} s+2 & s+1 & s+3 \\ s^3+2s^2+s & s^3+s^2+s & 2s^3+3s^2+s \\ s^2+3s+2 & s^2+2s+1 & 3s^2+6s+3 \end{bmatrix} \in \mathbb{R}^{3\times 3}[s].$$

Its eigenstructure can be retrieved much more easily by computing the Smith canonical form $S(s)$ of $M(s)$, which is equal to

$$S(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s^2+s \end{bmatrix} \in \mathbb{R}^{3\times 3}[s].$$

In section 2 of the present paper, the Smith canonical form $S(s)$ of a polynomial matrix $M(s)$ is analytically described. In the sequel, it is defined that the Smith form $S(s)$ of a matrix $M(s)$ is actually the unique representative (canonical form) of the corresponding $\mathbb{R}[s]$-equivalence class of $M(s)$, and it is characterized by the complete set of invariants (invariant polynomials).

Several numerical methods have been proposed for the computation of the Smith form of a given polynomial matrix $M(s) \in \mathbb{R}^{n\times n}[s]$. Most methods are based on performing elementary transformations on $M(s)$ until the required diagonal form is reached [1–3]. Specifically, these operations in [1] are performed on an integer matrix and in [2] are performed in the p-adic number system which reduces the problem of coefficient growth appearing frequently in this kind of transformations. In [5], a parallel probabilistic algorithm is developed.

In the present paper an algorithm based on the notion and properties of compound matrices [7] is proposed. The classical definition of the Smith form [3] is transferred in computing all the p-compound matrices $C_p(M(s))$ of $M(s)$, $1 < p \leq n$, and evaluating the greatest common divisor (gcd) of their elements by applying an appropriate numerical method. More explicitly, the given polynomial matrix $M(s) \in \mathbb{R}^{n\times n}[s]$ is expanded into a product

$$M(s) = [\boldsymbol{m}_1(s), \ldots, \boldsymbol{m}_n(s)] = T_M \cdot S(s), \quad T_M \in \mathbb{R}^{n\times k}, \quad S(s) \in \mathbb{R}^{k\times n}[s],$$

$$k = \sum_{i=1}^{n} (\delta_i + 1),$$

$\delta_i = \deg\{\boldsymbol{m}_i(s)\}$, and the computation of $C_p(M(s))$ is achieved using the formula $C_p(M(s)) = C_p(T_M)\cdot C_p(S(s))$ and introducing the notions of z-nonsingular sequences, $z \in Q_{p,n}$, where $Q_{p,n}$ is the set of all lexicographic sequences of p integers taken from $1, \ldots, n$ [7].

In section 2 all the required theoretical background for the formulation of the compound matrix algorithm is developed. The numerical algorithm and comments about its implementation are presented in section 3. Finally, in section 4 numerical results achieved after the application of the algorithm are demonstrated.

Throughout the paper $\mathbb{R}^{m\times n}[s]$ denotes the set of all real polynomial matrices and $\mathbb{R}^{m\times n}$ the set of all real matrices. If $A \in \mathbb{R}^{m\times n}$, then $\rho(A)$ denotes its rank, $C_p(A)$ is the pth other compound matrix of A [7] and $Q_{p,n}$ denotes the set of lexicographic sequences of p integers chosen from $1, 2, \ldots, n$. $\deg\{\boldsymbol{p}(s)\}$ denotes the degree of a polynomial vector. The symbol $\equiv$ means equal by definition.

2. Theoretical background

2.1. The Smith canonical form of a polynomial matrix

One of the most important mathematical tools for handling transfer functions as polynomial-matrix fractions and for studying systems described by polynomial matrices is the Smith canonical form $S_M(s)$ of a given polynomial matrix $M(s) \in \mathbb{R}^{m\times n}[s]$.

An equivalence relation on $\mathbb{R}^{m\times n}[s]$ may be defined as follows:

Definition 2.1 [3]
Let $M_1(s), M_2(s) \in \mathbb{R}^{m\times n}[s]$. $M_1(s), M_2(s)$ are said to be $\mathbb{R}[s]$-equivalent if and only if $M_2(s) = R(s)\cdot M_1(s)\cdot Q(s)$, where $R(s) \in \mathbb{R}^{m\times m}[s]$, $Q(s) \in \mathbb{R}^{n\times n}[s]$, $\det\{R(s)\}$, $\det\{Q(s)\} \in \mathbb{R} - \{0\}$. □

The matrices $R(s)$, $Q(s)$ with determinant in $\mathbb{R} - \{0\}$ are known as $\mathbb{R}[s]$-unimodular matrices.

It is well known that the relation defined above is an equivalence relation on the set $\mathbb{R}^{m\times n}[s]$. The $\mathbb{R}[s]$-equivalence class $\mathscr{E}_{\mathbb{R}[s]}(M(s))$ of $M(s) \in \mathbb{R}^{m\times n}[s]$ is characterised by a uniquely defined complete set of invariants as well as canonical form. The invariants and the canonical form are stated by the following theorem.

Theorem 2.1 [3]
Let $M(s) \in \mathbb{R}^{m\times n}[s]$.
(i) There exist unimodular matrices $L(s) \in \mathbb{R}^{m\times m}[s]$ and $R(s) \in \mathbb{R}^{n\times n}[s]$ such that:

$$L(s)M(s)R(s) = S_M(s) = \begin{array}{cl} \left[\begin{array}{ccc|c} f_1(s) & & & \\ & \ddots & & 0 \\ & & f_\rho(s) & \\ \hline & 0 & & 0 \end{array}\right] & \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\1\\1\end{array}}\right\}\rho \\ m-\rho \end{array} \\ \begin{array}{cc} \underbrace{\hphantom{f_1(s)\ddots f_\rho(s)}}_{\rho} & \; n-\rho \end{array} & \end{array} \in \mathbb{R}^{m\times n}[s], \tag{2.1}$$

where $S_M(s)$ is called the *Smith canonical form* of $M(s)$ and

(a) $\rho = \max_{s\in\mathbb{R}}\{\text{rank } M(s)\} \leq \min(m,n)$;
(b) The polynomials $f_i(s) \in \mathbb{R}[s]$, $i = 1,\dots,\rho$, called invariant polynomials of $M(s)$, are monic, uniquely defined by $M(s)$ and satisfy the division property $f_i(s)|f_{i+1}(s)$, $i = 1,2,\dots,\rho-1$.

(ii) The set $\{f_i(s),\ i = 1,2,\dots,\rho\}$ of invariant polynomials defined above forms a complete set of invariants for $\mathscr{E}_{\mathbb{R}[s]}(M(s))$ and $S(s)$ is a canonical form for this class. □

Moreover, consider the polynomials $D_i(s) \in \mathbb{R}[s]$, $i = 1,2,\dots,\rho$, given by $D_0(s) \equiv 1$, $D_i(s) :=$ the monic gcd of all $i \times i$ minors of $M(s)$.

These polynomials are called the determinantal divisors of $M(s)$. They are related to the invariant polynomials of $M(s)$ by the Smith algorithm, i.e. $f_i(s) = D_i(s)/D_{i-1}(s)$, $i = 1,2,\dots,\rho$.

If $f_i(s) = (s - s_{i,1})^{a_{i,1}} \dots (s - s_{i,k})^{a_{i,k_i}}$, $i = 1,2,\dots,\rho$, is the unique factorisation over $\mathbb{C}$ of $f_i(s)$, where $s_{i,1},\dots,s_{i,k_i} \in \mathbb{C}$ and $a_{i,1},\dots,a_{i,k_i} \in \mathbb{Z}^+$, then the elements of the set $\{(s - s_{i,j})^{a_{i,j}} : j = 1,2,\dots,k_i,\ i = 1,2,\dots,\rho\}$ are called the elementary divisors of $M(s)$.

Note that the knowledge of the elementary divisors of $M(s)$ and the number ρ of its invariant polynomials $f_i(s)$, $i = 1,2,\dots,\rho$, is sufficient to construct $f_1(s),\dots,f_\rho(s)$.

Under this construction, the invariant polynomials $f_1(s),\dots,f_\rho(s)$ are given by the formulas:

$$f_j(s) = \prod_{i=1}^{q} (s - s_i)^{a_{i,\rho+1-j}}, \quad j = 1,2,\dots,\rho,$$

where we put $(s - s_i)^{a_{i,j}} = 1$, for $j > k_i$ and q is the number of all different complex numbers that appear as roots of the elementary divisors.

Next, we introduce some useful notation concerning sequences of integers and compound matrices.

Notation 2.1 [7]

(i) $Q_{p,n}$ denotes the set of strictly increasing sequences of p integers $(1 \leq p \leq n)$ chosen from $1,2,\dots,n$. If $\alpha,\beta \in Q_{p,n}$ we say that α precedes β $(\alpha < \beta)$, if there exists an integer t $(1 \leq t \leq p)$ for which $\alpha_1 = \beta_1,\dots,\alpha_{t-1} = \beta_{t-1}$, $\alpha_t < \beta_t$, where α_i, β_i denote the elements of α,β respectively. This describes the lexicographic ordering of the elements of $Q_{p,n}$. The set of sequences $Q_{p,n}$ will be assumed to be lexicographically ordered.

(ii) Suppose $A = [a_{i,j}] \in \mathbb{R}^{m\times n}$, let k,p be positive integers satisfying $1 \leq k \leq m$, $1 \leq p \leq n$, and let $\alpha = (i_1, i_2,\dots,i_k) \in Q_{k,m}$ and $\beta = (j_1,j_2,\dots,j_p) \in Q_{p,n}$. Then $A[\alpha/\beta] \in \mathbb{R}^{k\times p}$ denotes the submatrix of A which contains the rows $i_1, i_2,\dots,i_k$ and the columns $j_1,j_2,\dots,j_p$.

(iii) Let $A \in \mathbb{R}^{m\times n}$ and $1 \leq p \leq \min\{m,n\}$, then the pth compound matrix or pth adjugate of A is the $\binom{m}{p} \times \binom{n}{p}$ matrix whose entries are $\det\{A[\alpha/\beta]\}$, $a \in Q_{p,m}$, $\beta \in Q_{p,n}$ arranged lexicographically in α and β. This matrix will be designated by $C_p(A)$.

2.2. *Computation of compounds of polynomial matrices*

2.2.1. *General formulation*

Let $M(s) = [\boldsymbol{m}_1(s),\dots,\boldsymbol{m}_n(s)] \in \mathbb{R}^{n\times n}[s]$, $\rho(M(s)) = r \leq n$, $\deg\{\boldsymbol{m}_i(s)\} = \delta_i$, $\boldsymbol{m}_i(s) = M_i \boldsymbol{e}_{\delta_i}(s)$, $M_i \in \mathbb{R}^{n\times(\delta_i+1)}$, $\boldsymbol{e}_{\delta_i}(s) = [1,s,\dots,s^{\delta_i}]^{\mathrm{T}}$ and assume that $M(s)$ is column reduced and ordered according to ascending degrees (i.e. $0 \leq \delta_1 \leq$

$\ldots \leq \delta_n$), then

$$M(s) = [M_1, \ldots, M_n] \cdot \mathrm{diag}\{e_{\delta_1}(s), \ldots, e_{\delta_n}(s)\} \equiv T_M \cdot S(s), \tag{2.2}$$

where $T_M \in \mathbb{R}^{n \times k}$, $k = \Sigma_{i=1}^{n}(\delta_i + 1)$ is the coefficient matrix of $M(s)$ and $S(s) \in \mathbb{R}^{k \times n}[s]$ is the structure matrix of $M(s)$ defined by the index $I = \{\delta_i : 0 \leq \delta_1 \leq \ldots \leq \delta_n\}$.

For an integer ρ, $1 < \rho \leq n$, we want to evaluate $C_\rho(M(s))$. From (2.2) it is evident that

$$C_\rho(M(s)) = C_\rho(T_M) \cdot C_\rho(S(s)), \tag{2.3}$$

where $C_\rho(T_M) \in \mathbb{R}^{\binom{n}{\rho} \times \binom{k}{\rho}}$, $C_\rho(S(s)) = [\ldots, \boldsymbol{c}_{\omega_i}(s), \ldots] \in \mathbb{R}^{\binom{k}{\rho} \times \binom{n}{\rho}}$, $\boldsymbol{c}_{\omega_i}(s) \in \mathbb{R}^{\binom{k}{\rho}}[s]$, $\omega_i = (i_1, \ldots, i_\rho) \in Q_{\rho,k}$, $\deg\{\boldsymbol{c}_{\omega_i}(s)\} = \delta = \Sigma_{j=1}^{n} \delta_{i_j}$.

Since the structure of $C_\rho(S(s))$ will define which part of $C_\rho(T_M)$ is essential for the structure of $C_\rho(M(s))$, let us first define the structure of $C_\rho(S(s))$.

2.2.2. Properties and evaluation of $C_\rho(S(s)) = [\boldsymbol{c}_1(s), \ldots, \boldsymbol{c}_{\binom{n}{p}}(s)]$

Every entry in $\boldsymbol{c}_i(s)$ can be parametrised by two sequences $\omega = (\omega_1, \omega_2, \ldots, \omega_\rho) \in Q_{\rho,k}$ and $a = (z_1, z_2, \ldots, z_\rho) \in Q_{\rho,n}$, specifying respectively the chosen rows and columns of $S(s)$ required for the construction of $C_\rho(S(s))$. This $\boldsymbol{c}_i(s)$ may be denoted as

$$\boldsymbol{c}_i(s) = [\ldots \boldsymbol{c}_\omega^z(s) \ldots]^{\mathrm{T}}, \quad \omega \in Q_{\rho,k}, \quad z \in Q_{\rho,n}. \tag{2.4}$$

Due to the form of $S(s)$ many entries of $\boldsymbol{c}_i(s)$ are set equal to zero, thus we are interested in specifying directly only the nonzero entries of $\boldsymbol{c}_i(s)$. More specifically, for each $z \in Q_{\rho,n}$ we want to define on the one hand the appropriate $\omega \in Q_{\rho,k} : c_\omega^z(s) \equiv 0$ and on the other hand the form of nonzero $c_\omega^z(s)$ and their corresponding location (in terms of ω).

We introduce first some notation.

Notation 2.2

The interval of integers $[1, \ldots, k]$ is partitioned into subintervals as shown below:

$$\begin{aligned} &\Delta_1 = [1, 2, \ldots, \delta_1 + 1], \quad \Delta_2 = [\delta_1 + 2, \ldots, \delta_1 + \delta_2 + 2], \quad \ldots, \\ &\Delta_n = [\delta_1 + \ldots + \delta_{n-1} + n, \ldots, k]. \end{aligned} \tag{2.5}$$

With each integer $\rho \in \{1, \ldots, k\}$ we associate two parameters, its *index* $\equiv v(\rho)$, indicating the interval where it belongs and its *stathm* $\equiv \sigma(\rho)$ indicating the relative order in its interval.

Proposition 2.1

For each $z = (j_1, \ldots, j_\rho) \in Q_{\rho,n}$ the coordinates $c_\omega^z(s)$, $\omega = (i_1, \ldots, i_\rho) \in Q_{\rho,k}$ have the following properties:

(i) $c_\omega^z(s) \neq 0$ if and only if $i_1 \in \Delta_{j_1}$, $i_2 \in \Delta_{j_2}$, …, $i_\rho \in \Delta_{j_\rho}$.

(ii) $c^z_\omega(s) \equiv 0$ – if at least two indices in ω are taken from the same interval Δ_{j_i}, $i = 1, \dots, p$;

– if at least one index from ω is taken from an interval Δ_{j_i} that does not contain any index from z.

(iii) If $i_1 \in \Delta_{j_1}, i_2 \in \Delta_{j_2}, \dots, i_p \in \Delta_{j_p}$, then

$$c^z_\omega(s) = s^{\sigma(i_1)+\dots+\sigma(i_p)}.$$

□

The proof of the above proposition is presented in the appendix.
The above proposition leads to the following definition.

Definition 2.2
For a sequence $z = (j_1, \dots, j_p) \in Q_{p,n}$ a sequence $\omega = (i_1, \dots, i_p) \in Q_{p,k}$ for which $i_1 \in \Delta_{j_1}$, $i_2 \in \Delta_{j_2}, \dots, i_p \in \Delta_{j_p}$ is called *z-nonsingular*; otherwise, i.e. if more than one index is taken from the same interval Δ_{j_i}, $i = 1, \dots, p$, it is called *z-singular*. The set of z-nonsingular sequences of $Q_{p,k}$ is denoted by $\Omega^z_{p,k}(\delta_1, \dots, \delta_n)$. □

From proposition 2.1 it is evident that for each $z \in Q_{p,n}$ the z-singular sequences of $Q_{p,k}$ define the zero coordinates of $\boldsymbol{c}_i(s)$ whereas the z-nonsingular ones the non-zero elements.

2.2.3. *Evaluation of the compound*

If $\Omega_k \equiv \{\omega = (x_1, \dots, x_p) \in Q_{p,n} : x_1 + x_2 + \dots + x_p = k\}$, i.e. Ω_k is the set of sequences with a given stathm k, the following theorem formulates the construction of the p-compound matrix $C_p(M(s)) = [\dots, \boldsymbol{c}_z(s), \dots]$, $z \in Q_{p,n}$, $1 < p \leq n$, of the polynomial matrix $M(s) \in \mathbb{R}^{n \times n}[s]$ specified by relation (2.2).

Theorem 2.2
Let $z \in Q_{p,n}$ and Ω_d be the subset of $\Omega^z_{p,k}$ of z-nonsingular sequences of $Q_{p,n}$ with a given stathm d. If $\boldsymbol{t}_\omega$ are the columns of $C_p(T_M)$ corresponding to the $\omega \in \Omega_d$ sequences, then the z-column of $C_p(M(s))$ is defined by

$$\boldsymbol{c}_z(s) = \sum_{i=0}^{k} \boldsymbol{t}_i s^i, \quad \boldsymbol{t}_i = \sum_{\omega \in \Omega_i} \boldsymbol{t}_\omega. \qquad (2.6)$$

The proof follows from the above analysis. □

2.3. *Application of compounds to Smith form evaluation*

Applying to the definitions given in section 2.1 the notion of compound matrices, the following theoretical procedure is derived for the evaluation of Smith form via compound matrices. It is supposed that $M(s) \in \mathbb{R}^{n \times n}[s]$ consists of coprime polynomials. (Otherwise, the computation of the gcd of all the given polynomial entries of $M(s)$ will be required.)

Algorithm SMITH (algorithm 2.1)

```
$D_0(s) := 1$
for $p := 1, \dots, n$
    evaluate $C_\rho(M(s)) = (c_{ij}(s)),\quad i,j = 1,2,\dots,\binom{n}{\rho}$
    find $D_\rho(s) := \gcd\{c_{ij}(s)\},\quad i,j = 1,2,\dots,\binom{n}{\rho}$
    $f_\rho(s) := D_\rho(s)/D_{\rho-1}(s)$
$S_M := \mathrm{diag}\{f_\rho(s)\},\quad \rho = 1,\dots,n$
```

The most important step in formulating an appropriate numerical algorithm for the above procedure is the numerical evaluation of the p-compound matrices $C_\rho(M(s))$ of $M(s) \in \mathbb{R}^{n\times n}[s]$ which will be achieved by the application of theorem 2.2.

Next, the numerical version of Algorithm Smith is presented.

3. The numerical algorithm

Let $M(s) \in \mathbb{R}^{n\times n}[s]$ be a given polynomial matrix, T_M the matrix of (2.2). The procedure described next evaluates matrix S_M that contains the coefficients of the invariant polynomials. The algorithm is written in a MATLAB-like language [4]. For a concrete value of p, $1 < p \leq n$, matrix GCDMAT contains the coefficients of the polynomials of each column of $C_\rho(M(s))$, while matrix COLGCD contains their gcd. Matrix GCDTOT contains the coefficients of the gcd of all the polynomial columns of $C_\rho(M(s))$. If $A \in \mathbb{R}^{n\times n}$, $\boldsymbol{r} \in \mathbb{R}^n$ the symbol $A = [A, r]$ adds a column to matrix A while the symbol $A = [A; r]$ adds a row to matrix A. The notation $A = [\,]$ denotes an empty matrix while $a = [\,]$ denotes an empty vector. Vector $\boldsymbol{r} = [1, 2, \dots, n]^{\mathrm{T}}$ can be generated by writing $r = 1 : n$. The function $\det(A)$ computes the determinant of A.

Algorithm SMITH (algorithm 3.1)

```
function $S_M = \mathrm{SMITH}(T_M)$
$n := \mathrm{rows}(T_M), f_1(s) := 1$
for $\rho = 2, 3, \dots, n$
    {Generate the column sequences of $S(s)$}
    $c = 1 : \rho$, GCDTOT = [ ], Prime = 0
    while $c = (c_1, \dots, c_p)$ is not the last sequence of $Q_{\rho,n}$
```

```
        Construct the $c$-nonsingular sequences $\Omega^{c}_{p,\Delta_{c_i}}$, $i = 1,\dots,p$
        specified from $\Delta_{c_1},\dots,\Delta_{c_p}$
        {Generate the row sequences of $T_M$}
        $r = 1 : \rho$, GCDMAT = [ ]
        while $r$ is not the last sequence of $Q_{\rho,n}$
            gcdv = [ ]
            $d$ = maximum stathm of $\omega \in \Omega^{C}_{\rho,\Delta_{c_i}}$, $i = 1,\dots,p$
            for $\omega \in \Omega^{C}_{\rho,\Delta_{c_i}}$, $i = 1,\dots,p$
              $g_\omega := \det(T_M(\omega, r))$
              gcdv : = [gcdv, $g_\omega$]
            gcdv : = $\sum_{\omega \in \Omega_d}$ gcdv$_\omega$
            if gcdv scalar, Prime : = 1, break
            GCDMAT : = [GCDMAT; gcdv]
            $r$ : = the next lexicographic sequence of $Q_{\rho,n}$
        if Prime, $f_\rho(s) := 1$, break
        else
            [COLGCD, Prime] = gcd(GCDMAT)
            if Prime, $f_\rho(s) := 1$, break
            CGDTOT = [GCDTOT, COLGCD]
        $c$ : = the next lexicographic sequence of $Q_{\rho,n}$
  $f_\rho(s)$ : = gcd(GCDTOT)
  $f_\rho(s) := f_\rho(s)/f_{\rho-1}(s)$
$S_M = \mathrm{diag}\{f_1(s),\dots,f_n(s)\}$
```

3.1. *Implementation of the algorithm*

Various modes can be used for an effective implementation of algorithm SMITH. Since the number of lexicographic sequences is quite remarkable we do not compute all the required sequences at once but only one at a time. When another sequence is actually required (in case the polynomial column is not coprime) we compute the next-in-order lexicographic sequence. The number k of sequences belonging in $\Omega^{c}_{\rho,\Delta_{c_i}}$, $i = 1,\dots,p$, is given from $k = \Pi^{\rho}_{i=1}\delta_{c_i}$, $c = (c_1, c_2,\dots,c_\rho)$. This number can be further reduced to k_r according to the structure of the given matrix (i.e., if concrete columns of T_M are zeros, the c-nonsingular sequences that contain these columns can be dismissed).

For the evaluation of the gcd any matrix-based algorithm [6,8,9] can be used since we construct at each stage a matrix containing the coefficients of the required polynomials.

Whenever coprime entries are specified or a scalar gcd is computed we simply continue to the next available index value of ρ, ignoring intermediate steps. Thus, the number of flops [4] cannot be specified from the beginning. The only required numerical process in the construction of compound matrices is that of the determinant of a matrix which requires $O(n^3)$ flops for a given matrix $T_M \in \mathbb{R}^{n\times n}$. For each value of p, $1 < p \leq n$, an estimation of an upper bound for the required flops in computing $C_\rho(M(s))$, $M(s) \in \mathbb{R}^{n\times n}[s]$ is equal to $O(k_r n^4/4)$.

For the computation of the gcd of polynomials the amount of required flops depends on the selected matrix-based method [8].

3.2. *Error analysis*

It is well known that the computation of the determinant of specified matrices and submatrices is a well defined stable numerical process [12,4]. Since the only numerical computation required in compound matrix determinations is the evaluation of a determinant, the numerical stability of such computations is well formulated.

Every step in algorithm SMITH requires determinantal and gcd computations. The evaluation of gcd is achieved by applying matrix-based methods; the numerical stability of such methods is defined in [8].

Therefore, every step of the method consists of numerically stable evaluations.

4. Numerical results – discussion

Algorithm SMITH was programmed on MATLAB environment and tested over several data. Next, we present some results achieved after the application of algorithm SMITH on a DFI 486 IBM-compatible computer. The computational complexity is estimated using the notion of flops [4] and the required time is evaluated using an appropriate function of MATLAB.

For the gcd computations a matrix-based method developed in [6] was used.

Example 4.1

$$M(s) = \begin{bmatrix} s-1 & 0 \\ 0 & s-1+\varepsilon \end{bmatrix} \in \mathbb{R}^{2\times 2}[s], \quad \varepsilon \in \mathbb{R}.$$

$S_M = \text{diag}\{1, s^2 + (\varepsilon - 2)s + 1 - \varepsilon\}$, $\varepsilon \leq 10^{-3}$, relative error $\leq 10^{-16}$, flops: 94, time: 0.06 sec. □

Example 4.2

$$M(s) = \begin{bmatrix} s & s & s-1 \\ s^2+s & s^2+2s & s^2-1 \\ 2s^2-2s & s^2-2s & 2s^2-3s+2 \end{bmatrix} \in \mathbb{R}^{3\times 3}[s].$$

$S_M = \text{diag}\{1, s, s^2\}$, relative error $\leq 10^{-16}$, flops: 9616, time : 1.7 sec. □

Example 4.3

$$M(s) = \begin{bmatrix} s^2+1 & s^2+3s+3 & s^2+4s-2 & s^2+3 \\ s-2 & s-1 & s+2 & s-2 \\ 3s+1 & 4s+3 & 2s+2 & 3s+2 \\ s^2+2s & s^2+6s+4 & s^2+6s-1 & s^2+2s+3 \end{bmatrix} \in \mathbb{R}^{4\times 4}[s].$$

$S_M = \text{diag}\{1, 1, 1, 1\}$, relative error $\leq 10^{-16}$, flops: 7987, time: 2.04 sec. □

Example 4.4 [2]

$$M(s) = \begin{bmatrix} s & -1 & -1 & 0 & 1 \\ 0 & s & 0 & -1 & 0 \\ 0 & 0 & -1+s & 0 & 0 \\ 0 & 0 & 0 & -2+s & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{5\times 5}[s].$$

$S_M = \text{diag}\{1, 1, 1, 1, -s^3+3s^2-2s\}$, relative error $\leq 10^{-16}$, flops: 9023, time: 1.87 sec. □

Example 4.5 [2]

$$M(s) = \begin{bmatrix} s-1 & 0 & 0 & -1 & 1 & 0 \\ 0 & s-1 & 2 & -3 & 3 & 0 \\ 0 & 0 & s+1 & -2 & 2 & 0 \\ -1 & 1 & -1 & s & -1 & 0 \\ -1 & 1 & -1 & 1 & s-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & s \end{bmatrix} \in \mathbb{R}^{6\times 6}[s].$$

$S_M = \text{diag}\{1, 1, 1, 1, s^2-2s+1\}$, relative error $\leq 10^{-16}$, flops: 127323, time: 33.88 sec. □

Example 4.6 [2]

$$M(s) = \begin{bmatrix} s & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & s-1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & s-1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & s-1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{7\times 7}[s].$$

$S_M = \operatorname{diag}\{1,1,1,1,1,0,0\}$, relative error $\leq 10^{-16}$, flops: 88035, time: 23.7 sec. □

4.1. *Remarks – comparison of methods – conclusions*

The compound-matrix method can work satisfactorily for any given polynomial matrix $M(s) \in \mathbb{R}^{n\times n}[s]$. It requires only the formulation of matrix T_M specified in (2.2). In the sequel, by applying the appropriate numerical version of the procedure described in section 2.2, for various values of $1 < p \leq n$, it computes the pth compound matrices of $M(s)$. Finally, the computation of the Smith canonical form $S_M(s)$ of $M(s)$ is achieved by direct application of its definition without computing unimodular transformation matrices P, Q such as $S_M(s) = P{\cdot}M(s){\cdot}Q$.

Since other methods were also developed for the achievement of Smith's canonical form computation (most of them mentioned in section 1 of the present paper), table 1 summarizes the main characteristics and properties of these existing methods.

In the sequel, we can summarize the main advantages of the new numerical approach based on compound matrices.

(1) *Direct calculations.*

The whole method applies direct formulas and achieves the required computations without going through row and column operations as the methods of [1] and [10] do. Thus it overcomes the problem of coefficient growth.

(2) *Works successfully in floating-point arithmetic.*

The method starts working directly with floating point arithmetic and, due to its nature, the cumulative error introduced does not alter seriously the computed Smith form from the required theoretical one (see example 4.1).

(3) *Freedom in the selection of the required gcd method – ability of computing approximate Smith canonical form.*

Again, due to the nature of the present method, any proposed method achieving gcd calculations of polynomials can be applied, in contrast with the methods of [1]

Table 1
Comparison of existing methods.

Bradley's method [1]	*Origination*: Works only for integer matrices and is based on explicit calculation of the gcd and of a set of multipliers for each of the rows and columns. *Numerical problems*: Numerical instability and the growth of magnitude of the matrix elements throughout the procedure (especially of the determinant of the given matrix is large). *Estimated computational complexity*: $O(n^2 \log_2(\|\det(M)\|) + 4n^3/3)$.
Pace and Barnett's method [10]	*Origination*: Performs gcd calculations implicitly, based on Blankiship's procedure. *Numerical problems*: Same as Bradley's method. *Estimated computational complexity*: An overall estimation was not produced, an estimation for the complexity required for the application of Blankiship's method in two polynomials can be found in [11].
Ramachandran's method [2]	*Origination*: Based on finite field transforms overcomes the problem of coefficient growth and errors in floating point operations by maintaining the coefficients of the polynomial entries in rational form. The whole arithmetic is performed in the p-adic system. *Numerical problems*: The problem of selecting appropriate values for a prime p and a positive integer r required for the beginning of the algorithm. *Estimated computational complexity*: No complexity analysis was produced. The method takes more time than floating point computations in rational form.
Kaltofen et al.'s method [5]	*Origination*: A probabilistic technique is applied which produces random linear combinations of appropriate minors whose gcd is with high probability equal to the needed gcd. The whole method performs matrix multiplications, determinant and gcd calculations. *Estimated computational complexity*: The algorithm belongs to the complexity class RNC^2 (see [5] for the definition of this class).
Compound-matrix method	*Origination*: The method starts working on the given matrix based on compound matrices definitions without performing any row or column operations on the matrix and thus it overcomes problems of numerical instability. *Estimated computational complexity*: $O(k_r n^4/4)$ plus the required operations concerning gcd evaluations.

and [10] that are both adusted to Blankiship's technique for gcd evaluation. Taking advantage of this property, in case we are having polynomial matrices containing numerically ε-dependent [9] polynomial entries, then according to the selected matrix-based method for the gcd computation, an approximate Smith canonical form of the matrix can be determined. For example, let us suppose that we are given the matrix:

$$M(s) = \mathrm{diag}\{s^2 - 3s + 2, \quad s^2 - 2.99s + 1.99, \quad 2s^2 - 5.99s + 3.99\} \in \mathbb{R}^{3\times 3}[s].$$

By applying for gcd evaluations in algorithm SMITH the method developed in [6], the following Smith form is computed:

$$S_M = \mathrm{diag}\{s - 1, \quad s - 1, \quad (s - 1)^2 \cdot (s - 2) \cdot (s - 1.99) \cdot (2s - 3.99)\}.$$

On the other hand, in the same algorithm if the method developed in [9] will be applied for the required gcd evaluations, for specific values of the applied accuracy the following approximate Smith canonical form can be defined:

$$\begin{aligned} S_{M,\mathrm{Appr}} = \mathrm{diag}\{&s^2 - 3s + 2, \quad s^2 - 2.99s + 1.99, \\ &(s^2 - 3s + 2) \cdot (2s^2 - 5.99s + 3.99)\}. \end{aligned}$$

This is due to the ability of this specific method to determine approximate gcds of polynomials according to specified accuracies.

(4) *Satisfactory execution times.*

The following matrix summarizes the time (in seconds) required for the execution of algorithm SMITH over various polynomial matrices with different dimension n.

n	2	3	4	4	5	5	6	7
sec	0.06	1.7	2.04	11.47	1.87	8.84	33.88	23.7

It is evident that the structure of the given matrix determines the execution time (i.e., although the matrix in example 4.4 has higher dimension than the one in example 4.3, the execution time is quicker since it possesses a rather simpler structure). A comparison of the above matrix with the one proposed in [10] shows that the achieved execution times are rather satisfactory.

(5) *Parallel nature of computations.*

Since the compound matrix evaluations for different values of p can be executed simultaneously, the whole algorithm can be applied in a parallel machine. The whole issue is under consideration (this could be extremely useful for cases when n is quite large).

Appendix

Proof of proposition 2.1
Matrix $S(s)$ is of the form

$$S(s)=\left[\begin{array}{c|c|c|c} 1 & & & \\ s & & 0 & \\ \vdots & & & \\ s^{\delta_1} & & & \\ \hline & 1 & & \\ & s & & \\ & \vdots & & \\ & s^{\delta_2} & & \\ \hline & & \ddots & \\ \hline & & & 1 \\ & & & s \\ 0 & & & \vdots \\ & & & s^{\delta_n} \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{matrix}1\\ s\\ \vdots\\ s^{\delta_1}\end{matrix}}\right\}\Delta_1 \\ \left.\vphantom{\begin{matrix}1\\ s\\ \vdots\\ s^{\delta_2}\end{matrix}}\right\}\Delta_2 \\ \vphantom{\ddots} \\ \left.\vphantom{\begin{matrix}1\\ s\\ \vdots\\ s^{\delta_n}\end{matrix}}\right\}\Delta_n \end{array} \qquad (A.1)$$

If $z=(j_1,j_2,\ldots,j_p)\in Q_{p,n}$ and $\omega=(i_1,i_2,\ldots,i_p)\in Q_{p,k}$ are the corresponding sequences determining the columns and rows of $S(s)$ required for the construction of $C_p(S(s))$, $1<p\leq n$, and $S^z_\omega(s)$ is any $p\times p$ submatrix of $S(s)$, the following properties are derived by inspection of the structure of $S(s)$.

(1) Every row of $S^z_\omega(s)$ contains only one nonzero element of the type s^β, $\beta=0,1,\ldots,\delta_i$, $i=1,2,\ldots,n$.
(2) If a column of $S^z_\omega(s)$ contains more than one nonzero element, then from (1) it follows that there is at least one zero column in $S^z_\omega(s)$ and thus $\det(S^z_\omega(s))=0$.
(3) The condition that $S^z_\omega(s)$ has a column with more than one nonzero element is equivalent to the one that at least two indices from ω are taken from an interval Δ_{j_k} containing at least two indices from z.
(4) The condition that $S^z_\omega(s)$ has a zero column is equivalent to the one that at least one index from ω is taken from an interval Δ_{j_1} that does not contain any index from z.
(5) If every column in $S^z_\omega(s)$ contains only one nonzero element then the submatrix $S^z_\omega(s)$ is diagonal and thus has the form:

$$S^z_\omega(s)=\operatorname{diag}\{\, s^{\sigma(i_1)},\quad s^{\sigma(i_2)},\quad \ldots,\quad s^{\sigma(i_p)}\,\} \qquad (A.2)$$

for the following reasons:

- Since $i_1<i_2<\ldots<i_p$ and $i_1\in\Delta_{j_1}$ (otherwise we have two indices from the same interval Δ_{j_k} and thus an entirely zero column) the nonzero element on the

first column is on the first row, i.e. in $(1,1)$ position, and, by inspection, has a value $s^{\sigma(i_1)}$.

- For the same reasons as before, $i_k \in \Delta_{j_k}$, $k = 2, \ldots, \rho$, and since every row has only one nonzero element, it cannot be on the $k-1$ row. Since i_k defines the k row and all nonzero elements are on the kth column, it follows that the kth element associated with i_k is in the (k,k) position of $S_{\omega}^{z}(s)$ and obviously has value $s^{\sigma(i_k)}$.

Thus $c_{\omega}^{z}(s) = \det(S_{\omega}^{z}(s)) = \Sigma_{k=1}^{\rho} s^{\sigma(i_k)}$. □

Acknowledgement

The authors would like to thank Prof. W.M. Coughran for his useful suggestions and references.

References

[1] G. Bradley, Algorithms for Hermite and Smith normal matrices and linear diophantine equations, Math. Comp. 25 (1971) 897–907.

[2] V. Ramachandran, Exact reduction of a polynomial matrix to the Smith normal form, IEEE Trans. Auto. Contr. AC-24 (1979) 638–641.

[3] G. Gantmacher, *Theory of Matrices*, Vol. 1, 2 (Chelsea, 1963).

[4] G. Golub and F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, Baltimore and London).

[5] E. Kaltofen, M. Krishnamoorthy and B. Saunders, Fast parallel computation of Hermite and Smith forms of polynomial matrices, SIAM J. Alg. Discr. Meth. 8 (1987) 683–690.

[6] N. Karcanias and M. Mitrouli, A matrix pencil based numerical method for the computation of the gcd of polynomials, IEEE Trans. Auto. Contr. (1992), to appear.

[7] M. Marcus and M. Minc, *A Survey of Matrix Theory and Matrix Inequalities* (Allyn and Bacon, Boston, 1964).

[8] M. Mitrouli and N. Karcanias, A survey of matrix-based numerical methods for the computation of the gcd of several polynomials, *Proc. IMA Int. Conf. on Control*, Manchester, England (1992).

[9] M. Mitrouli and N. Karcanias, Computation of the gcd of polynomials using Gaussian transformations and shifting, Int. J. Control 58 (1993) 211–228.

[10] I.C. Pace and S. Barnett, Efficient algorithms for linear system calculations. Part I – Smith form and common divisor of polynomial matrices, Int. J. Syst. Sci. 5 (1974) 403–411.

[11] I.C. Pace and S. Barnett, Comparison of algorithms for calculation of gcd of polynomials, Int. J. Syst. Sci. 4 (1973) 211–226.

[12] J.H. Wilkinson, *Rounding Errors in Algebraic Processes* (Her Majesty's Stationery Office, London, 1963).