NUMERICAL ISSUES AND COMPUTATIONAL PROBLEMS IN ALGEBRAIC CONTROL THEORY

BY

MARILENA TH. MITROULI

NUMERICAL ISSUES AND COMPUTATIONAL PROBLEMS IN ALGEBRAIC CONTROL THEORY

BY
MARILENA TH. MITROULI
D. Maths., M. Sc.

**THESIS SUBMITTED FOR THE
AWARD OF THE DEGREE OF
PH. D. IN COMPUTATIONAL METHODS
FOR CONTROL THEORY**

THE CITY UNIVERSITY
LONDON EC1V 0HB
CONTROL ENGINEERING CENTRE
DEPARTMENT OF ELECTRICAL ELECTRONIC
AND INFORMATION ENGINEERING

M A R C H 1 9 9 1

Dedicated to my parents

CONTENTS

DECLARATION

ACKNOWLEDGEMENTS

Among the people that have played a role in the preparation of this thesis, I first of all want to mention my supervisor Dr. N. Karcanias, Reader in Control Theory. I would like to thank him for his invaluable support during the past four years, both in theoretical as well as in more practical matters. His enthusiasm towards (almost) everything related to the field of my research, has always given me the inspiring and exciting feeling of "being close to where it all happens".

Also I want to express my thanks to As/nt Professor of the Department of Mathematics of the University of Athens Dr. Gr. Kalogeropoulos for many stimulating discussions and comments related to this research.

I am very grateful to the Mathematics Department of the University of Athens for the financial support and their understanding.

Special thanks go to Professor S. Negrepontis for his continuous support and interest in the progress of my work.

I would also like to express my appreciation to As/nt Professor Dr. G. Karabatzos for his assistance and interest in the problems formulated and treated in this thesis.

Finally, I want to mention Professor Ch. Charalambides for his support and understanding during the preparation of this work.

My postgraduate studies, leading to this thesis, have been made possible by the unbounded support of my parents, Theodore and Anastasia. In recognition of their invaluable contribution, I dedicate this thesis to them.

ABSTRACT

The work of this thesis concerns computational issues arising from various fields of Algebraic Control Theory. Efficient algorithms covering the following classes of problems are developed.

(i) Exterior Algebra Computations : For given matrices $A \in R^{mxn}$, $B \in R^{mxl}[s]$ algorithms achieving the computation of $C_p(A)$, $C_q(B)$, $1 \leq p \leq \min\{m,n\}$ and $1 \leq q < l$ are formulated. An algorithm for the evaluation of Plucker matrices is also proposed. Most of these algorithms are used in the development of a unifying numerical algorithm for the solution of the Determinantal Assignment Problem.

(ii) Numerical Techniques for handling nongeneric computations : Several numerical tools for the diagnosis of certain properties in an "almost sense", and the definition of procedures attaining the termination of algorithms are developed.

(iii) Evaluation of the Greatest Common Divisor of polynomials : A new numerical algorithm for the evaluation of the greatest common divisor of any set of polynomials is formulated.

(iv) Almost Zero Computations : Algorithms achieving the evaluation of the Prime almost zero of a polynomial set and the computation of the zero radius are given. Useful comments about the achievement of improved bounds for the zero trapping region are also presented.

NOTATION AND ABBREVIATIONS

Throughout this thesis, the following notation and abbreviations will be used.

R, C, $R(s)$	the field of real, complex numbers and rational functions respectively.
Z	the set of integers.
Z^+	the set of positive integers.
$R[s]$	the ring of polynomials over R.
R^n, C^n, $R^n(s)$	the n-dimensional vector spaces over R, C, R(s).
F^{mxn}	the set of mxn matrices with elements from the field, or ring F.
$R^{mxn}[s]$	the set of mxn matrices with elements from the ring of polynomials over R.
$R^n[s]$	the nx1 column vectors with elements from R[s].
$N_r(H)$, $N_l(H)$	the right, left null space of a map H.
$\rho_F(A)$	the rank of a matrix $A\epsilon F^{mxn}$ over the field F.
$\Lambda^p V$	the p-th exterior power of the vector space V.
$Q_{m,n}$	the set of lexicographic ordered, strictly increasing sequences of m integers from 1,2,...,n.
$\underline{a}_{i_1}\wedge\ldots\wedge\underline{a}_{i_m}=\underline{a}_\omega\wedge$	the exterior product of the vectors $\underline{a}_1,\ldots,\underline{a}_m$ of a n-dimensional vector space V where $\omega=(i_1,\ldots,i_m)\epsilon Q_{m,n}$.
$C_p(A)$	the p-th compound matrix of $A\epsilon F^{mxn}$, $p\leq\min\{m,n\}$.
$E(P)$	the equivalence class of a set P.
$P_{m,d}$	the set of m polynomials of maximal degree d.
$\varphi(A)$	the spectral radius of a matrix A.
$\bar{A}$	the complex conjugate of a matrix A.
$M=\left[\begin{array}{c\|c}A & B\\ \hline C & D\end{array}\right]\begin{matrix}k\\ l\end{matrix}$ $\begin{matrix}\quad n & m\end{matrix}$	will denote that $A\epsilon F^{nxk}$, $B\epsilon F^{mxk}$, $C\epsilon F^{nxl}$, $D\epsilon F^{mxl}$.
dim X	the dimension of X.
det(A)	the determinant of a matrix A.
diag(d_i)	the diagonal matrix with elements d_i.
deg{r(s)}	the degree of a polynomial r(s).

$a(s)\|b(s)$	will denote that polynomial a(s) divides polynomial b(s).
$sp\{\underline{x}_i,\ i=1,...n\}$	the space spanned from the vectors $\underline{x}_i$.
$\|.\|$	will denote any norm.
$\|\ \|_2$	the 2-norm or spectral norm.
$\|\ \|_F$	the Frobenius or Euclidean norm.
$\|\ \|_\infty$	the maximum row sum norm.
$\perp$	denotes orthogonality of vectors.
Re{a}	the real part of a complex number.
Im{a}	the imaginary part of a complex number.
iff	means if and only if.
<<	means much smaller.
>>	means much larger.
$\equiv$	means equal by definition.
$\approx$	means approximately equal.
$i\epsilon\underline{n}$	$i\epsilon\{1,2,...,n\}$.
A^T	the transpose of a matrix.
$\underline{a}^t$	the transpose of a vector.
g.c.d.	greatest common divisor.
a.z.	almost zero.
u.f.d.	unique factorization domain.
DAP	Determinantal Assignment Problem.
S.V.D.	Singular Value Decomposition.
P.R.S.	Polynomial Remainder Sequence.
b.m.	basis matrix.
v.r.	vector representative.

Small underlined letters will denote vectors e.g. $\underline{a}$.
Capital letters will denote matrices e.g. A.
Italic letters will denote sets e.g. *A*.

LIST OF ALGORITHMS

1) EXTERIOR ALGEBRA ALGORITHMS

2) ALGORITHMS CONCERNING NONGENERIC COMPUTATIONS

3) ALGORITHMS RELATED TO G.C.D. COMPUTATION

4) ALGORITHMS CONNECTED WITH ALMOST ZEROS AND TRAPPING DISCS

CHAPTER 1

INTRODUCTION

Control Theory and Control Systems Design have demonstrated the importance of many areas of mathematics for the study of problems with real engineering significance. The development of the algebraic approach [Ros., 1], [Kail., 1], [Vid., 1] etc; geometric approach [Won., 1] and algebrogeometric approach [Kar. & Gia., 1,2], have demonstrated the need for algebraic computations and motivated the development of areas in computations, which otherwise could have been thought of no relevance. More specifically, areas such as computational issues of polynomial, rational matrices, matrix pencil theory, exterior algebra, Riccati equations etc, could never have been considered as of any practical significance, if it was not for their role in Control Theory and Design.

The importance of computational issues in the Control Theory design area, has been realised in the last decade, [Laub, 1], [VanDoor., 1] etc, but the work has mainly concentrated to problems related to state space computations, which involve standard Numerical Linear Algebra. The topic of algebraic computations, such as those of transformations and computation of canonical forms and invariants of polynomial and rational matrices, solution of polynomial matrix equations, factorisations etc, have not been properly addressed so far from the computational view point. Issues related to the algebrogeometric approach [Kar. & Gia., 1,2] such as exterior algebra computations of real and polynomial matrices, have not been considered before; however, because of the significance of the approach, the latter issues are of particular current interest. Theoretical procedures for evaluation of algebraic entities, which may be performed by pencil and paper, are not always the best, when it comes to their computer implementation. Reduction of algebraic computational procedures to standard Numerical Linear Algebra problems, is in general something desirable, but not always feasible.

It seems that the development of symbolic languages and packages for symbolic calculations may assist greatly in the development of the algebraic computations, but there are many issues which still have to be addressed using traditional Numerical Analysis tools and techniques. One of these issues has to do with the computation of system invariants which generically do not exist on a family of models, such as zeros of nonsquare systems, noncoprimeness of polynomials etc. For such problems theoretical algorithms applied on generic models almost always converge to the generic solutions. Devising tools for "catching up" of approximate solutions is an important issue, of great engineering significance, and rather difficult to tackle with conventional means. An additional task emerging here is the development of an appropriate "analytic" interpretation for concepts which originate from algebra.

This dissertation is concerned with the development of appropriate numerical methods for handling several important computational problems that arise from various fields of Algebraic Control Theory. Since the number of crucial and complicated computational problems existing in Control Theory is quite remarkable, the derivation of stable numerical techniques for handling them is important.

In the present thesis efficient numerical algorithms are proposed for the following three classes of problems :

(i) Computations based on Exterior Algebra

(ii) Issues Related to Nongeneric Computations

(iii) Computations involving almost zeros and trapping discs

In Chapter 2, some basic concepts and results from Algebraic Theory of Linear Systems are presented. Several descriptions and approaches for Linear Dynamic Models are mentioned and the most important computational problems arising from their introduction are analytically formulated. Some definitions and classification concerning system invariants and unstructured generic systems are summarized in Chapter 3. The notions of generic and nongeneric computations are also introduced and the reasons for computing approximately the values of nongeneric invariants are explained.

In Chapter 4, the problem of Exterior Algebra computations is considered. In the beginning, useful definitions concerning sequences of integers are introduced. The notions of P-Prime and N-Prime sequences are introduced and in the sequel these definitions are applied for the evaluation of compound matrices. For given matrices $A \epsilon R^{mxm}$, $B \epsilon R^{mxl}[s]$, algorithms attaining the evaluation of $C_p(A)$, $1 \leq p \leq \min\{m,n\}$, $C_1(B)$, $C_q(B)$, $1 \leq q \leq l$ are proposed. A numerical technique evaluating the Smith-Normal form of a polynomial matrix using as tools the notion of compound matrices is developed, and finally an efficient numerical technique for the evaluation of Plucker matrices is formulated.

The problem of handling nongeneric computations is introduced in Chapter 5. Stable numerical techniques for facing such computations are demonstrated. Useful tools providing the means for encountering nongenericity are introduced. More specifically, the Chapter starts with a brief description of the Singular Value Decomposition theorem. This theorem is applied almost always when nongeneric computations are required. The notion of rank of a matrix is replaced by the numerical ϵ-rank, for a given accuracy ϵ, and the notions of ϵ-independent, numerically ϵ-dependent, strongly ϵ-dependent, fuzzy

ε-dependent sets of vectors are introduced. The properties of strongly ε-dependent sets of vectors are carefully studied and necessary and sufficient conditions relating the numerical ε-rank of such sets and their singular values are formulated. Efficient criteria for choosing the "best" representative of such sets are also defined.

The problem of selecting a "best uncorrupted" base for the row space of a matrix is also mentioned. Applying the theory of Gram matrices and compound matrices, a stable numerical algorithm achieving this selection is proposed.

Finally, a detailed survey concerning the most important properties of the Gramian of given vectors and the Schur complement is presented. Due to their properties, the Gramian and the Schur complement form efficient tools useful in coping with nongeneric computations.

The problem of finding the greatest common divisor (g.c.d.) of a set of polynomials is very frequently encountered in problems of Control Theory. Thus, a detailed survey of methods for computing the greatest common divisor of polynomials is presented in Chapter 6. These methods are classified in two main categories. The first category contains methods based on the well known Euclid's algorithm. Except Euclid's algorithm, the generalized Euclid's algorithm is discussed and other variations such as Collin's and Routh's algorithm are also mentioned. Furthermore, the extension of all the above algorithms to unique factorization domain is developed. The algorithms in this category are appropriate only when the g.c.d. of two polynomials is required.

The second category contains numerous matrix methods. Numerical algorithms due to Blankiship, Sylvester and Barnett are analytically demonstrated. All the matrix-based algorithms can compute the g.c.d. of several polynomials.

In Chapter 7, a new numerical method for the computaion of the g.c.d. of a m-set of polynomials of R[s], $P_{m,d}$ of maximal degree d, is presented. This method is based on a recently developed theoretical algorithm [Kar., 1] that uses elementary transformations and shifting operations; this algorithm takes into account the nongeneric nature of g.c.d. and thus uses steps which minimize the introduction of additional errors and defines the g.c.d. in an approximate, or almost sense. For a given set $P_{m,d}$, with a basis matrix P_m, the method defines first the most orthogonal uncorrupted base P_r from the rows of P_m, where $r=\rho(P_m)\leq m$. By applying successively Gaussian transformations and shifting on the basis matrix $P_r \epsilon R^{rx(d+1)}$ we produce each time a new basis matrix P_z with $z=\rho(P_z)<r$. The method terminates when the rank of P_z is approximately equal to 1; the coefficient vector of the g.c.d. is then defined as a row of the unit rank matrix P_z. The method, defines the exact degree of

the g.c.d., evaluates successfully an approximate solution, and works satisfactorily with large numbers of polynomials of any fixed degree.

Several subjects connected with the almost zero notion are discussed in Chapter 8. The most important properties of almost zero are summarized in the beginning. In the sequel, an algorithm evaluating the prime almost zero of a given polynomial set is proposed. A detailed study concerning the almost zero's sensitivity is developed next. The new notions of B-scaled, normalized, $\| \|_\infty$ row-scaled almost zero, are defined and several useful remarks about the effect that the distribution of the original polynomials' roots causes in the almost zero's position are derived.

The problem of fixed polynomial combinants is also considered. After a brief presentation of their most important properties, an algorithm computing an upper bound for the zero radius is developed. Useful remarks on how we can attain improved bounds of the zero trapping region if the sensitivity of almost zero to scaling is applied, are also concluded.

Finally, the definition of dynamic combinants is established. Conditions providing zero assignability and strong nonassignability of fixed order dynamic combinants are also proved.

In Chapter 9, the problem of formulating a computational framework for the solution of the Determinantal Assignment Problem (DAP) [Kar. & Gia., 1,2] is introduced. Many of the exterior algebra algorithms developed in Chapter 4 are applied in this concrete Chapter. A unifying algorithm which treats DAP as a constrained optimization problem is suggested. The introduction of such an algorithm is really very useful, since this algorithm may be used as a basis of a design technique centered around the frequency assignment problems.

Throughout the thesis, several numerical examples illustrating the applications of the proposed algorithms are presented. Most of the algorithms were programmed on a CYBER 170-730 computer. This machine has double precision accumulator, arithmetic base $\beta=8$, number of digits of machine's word $t=60$ and arithmetic precision for single precision computations about 10^{-14}. All the source lists of algorithms can be found in [Mit. & Kar., 1].

CHAPTER 2

ALGEBRAIC THEORY OF LINEAR SYSTEMS AND THE NEEDS FOR ALGEBRAIC COMPUTATIONS

2.1 INTRODUCTION

The nature of Control Theory always depends on the type of system model. The types of system model that are frequently used are:

(i) Diagraph Models (Linear graphs)

(ii) Steady-State Models (Linear and Nonlinear)

(iii) Linear, Time invariant, Lumped-parameter Dynamic Models of the state space, Transfer Function matrix, or Matrix Fraction description type.

The last family is referred in short as Linear Dynamic Models (LDM). The family of LDMs is the richest of the three families. Diagraph Models (DM) and Linear Steady-State Models (LSSM) may be considered models describing certain aspects of the structure of LDMs and they may be discussed within the general framework of LDMs. The specific objective of this Chapter is to provide a short review of descriptions and Analysis-Design approaches for LDMs on the one hand and on the other hand to formulate the most important computational problems arising from the various descriptions and approaches of LDMs.

2.2 BASIC SYSTEM DESCRIPTIONS [Kar., 3]

A linear dynamic system [Chen, 1] is a set of four linear spaces U,V,Z,Y and linear maps g,f,h,e interrelated in the manner illustrated in Figure 2.1

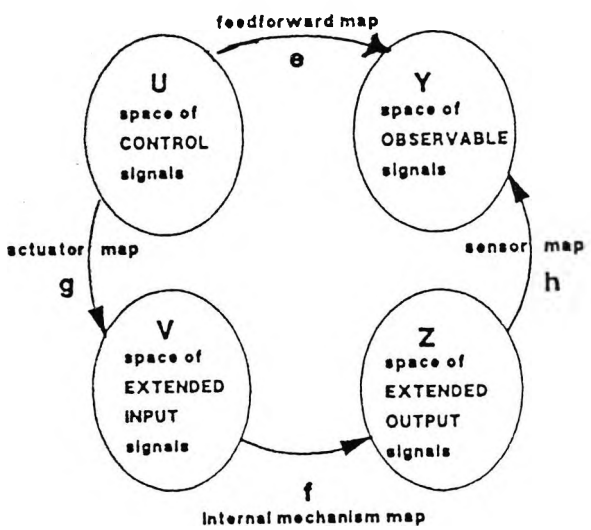


Figure 2.1

Such a model, because it involves internal as well as external variables, will be called internal model. The feedforward map e expresses the direct coupling of inputs-outputs which occasionally may be present. If the system is represented by the linear spaces U,Y and the linear map w:U——>Y, then the representation will be called input-output, or external.

Note that in the external representation, internal variables are ignored; thus, an implicit assumption in external descriptions is that all internal variables have zero initial values. A system with all internal variables having zero initial conditions will be called initially relaxed.

The nonzero initial state of internal variables may be taken into account only in internal descriptions. A diagram illustrating the families of linear models is given next.

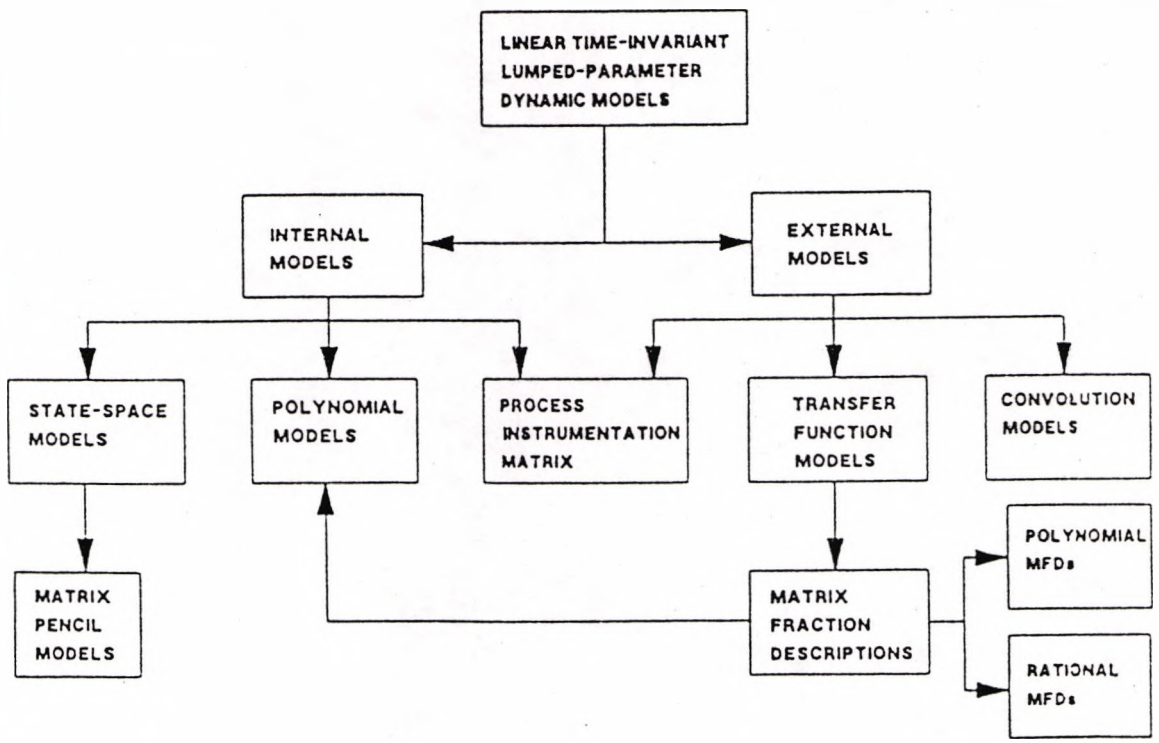


Figure 2.2

Concrete mathematical descriptions for the above types of models are briefly discussed next.

2.2.1 Internal models

The basic two families of internal models for linear systems are the state-space and the polynomial models and they are discussed next.

(i) State Space Models:

Consider the linear system described by

$$S(A,B,C,D): \begin{cases} \dot{\underline{x}} = A\underline{x} + B\underline{u} & (2.1a) \\ \underline{y} = C\underline{x} + D\underline{u} & (2.1b) \end{cases}$$

where $A \epsilon R^{mxn}$, $B \epsilon R^{nxl}$, $C \epsilon R^{mxn}$, $D \epsilon R^{mxl}$, and $\underline{x} \epsilon R^{n}$, $\underline{y} \epsilon R^{l}$ are real-valued vector functions. Such a model is known as a state-space model (see for instance [Chen, 1], [Won., 1]. A is called the internal dynamics matrix and its properties stem from the natural dynamic characteristics of the system. The matrices B, C are called input-, output- matrices respectively and they express the coupling of input, output variables $\underline{u}$, $\underline{y}$ to the internal variables $\underline{x}$, known as states; thus, B, C represent the cummulative effect of selecting actuators (B matrix) and sensors (C matrix) for the system; because of the latter property, we may also refer to B as the actuator matrix and to C as the sensor matrix. The internal variables of this model are the states $\underline{x}$ and its derivatives $\dot{\underline{x}}$.

A standard "block diagram" representation of the system is indicated in Figure 2.3.

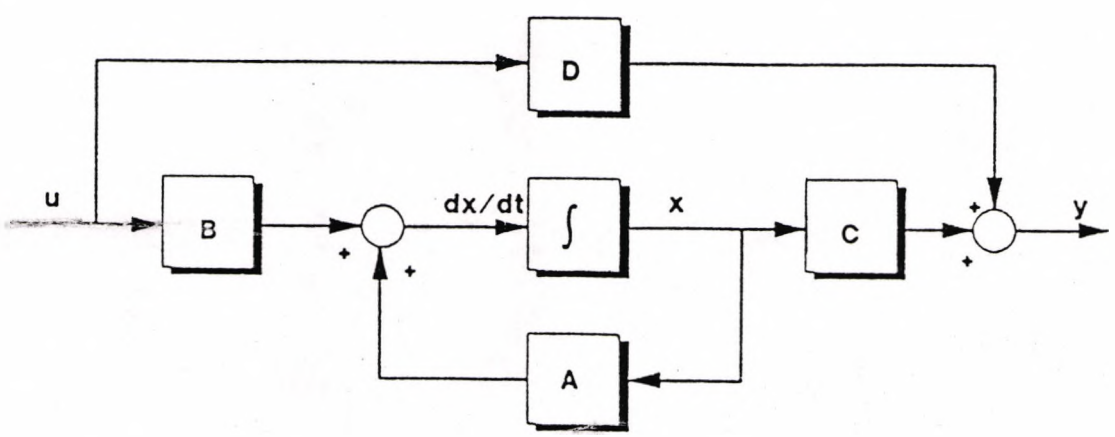


Figure 2.3

A state space model with D constant is called proper (or causal),whereas if D=0, is called strictly proper. If in (2.1b) D is not a constant but a polynomial matrix in p=d(.)/dt, then the system is called nonproper. Nonproper systems may be represented in terms of singular models ($E\dot{x}=Ax+Bu$, $|E|=0$) [Lew,1], which are useful in describing the fast dynamics under failure conditions in circuits etc.; however, such models do not seem to be relevant for process applications.

The number of states n of the state-space model is defined as its order. It is always assumed that the measurements and actuation variables are independent and thus $m<n$, $1<n$ and $\rho(B)=1$, $\rho(C)=m$.

If $p=d(.)/dt$ denotes the derivative operator, the description (2.1) may be expressed as

$$\begin{bmatrix} pI-A & -B \\ -C & -D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ -y \end{bmatrix} \iff P(p)\ \xi(t) = \begin{bmatrix} 0 \\ -y \end{bmatrix} \qquad (2.2)$$

where P(p) is a special type of a polynomial matrix, referred to as the system matrix pencil [Ros., 1]. Matrix pencils [Gant., 1], appear as simple linear operators of the type sF-G, F, $G\epsilon R^{pxk}$, s an indeterminate, and are naturally associated with state-space type problems. The theory of the structure and invariants of state-space models is described by the structural characteristics of appropriate matrix pencils. Matrix pencil theory [Gant., 1], is intimately related to the generalised eigenvalue-eigenvector problem [Wilk., 2] and is thus central to state space computations. The model described by (2.2) will be called a matrix pencil description of S(A,B,C,D). A consequence of the independance of actuation and measurement variables is that $\rho(P(\lambda))=\min\{n+m, n+1\}$ for almost all values of the complex number λ; Systems with such a property will be called nondegenerate. Well designed systems always have this property.

(ii) Polynomial Models:

The state space description of a linear system assumes that the system is described in terms of first order differential equations; however, this is not the most general internal description for linear systems. For a number of processes, the most natural description is that defined by the general differential system [Ros., 1]

$$\Sigma p: \begin{cases} A(p)\underline{y}(t)=B(p)\underline{u}(t) \\ \\ \underline{y}(t) = C(p)\underline{y}(t)+D(p)\underline{u}(t) \end{cases} \qquad p = \frac{d.}{dt} \tag{2.3}$$

where A(p), B(p), C(p), D(p) are polynomial matrices in p [Gant., 1] of dimension nxn, nx1, mxn respectively, and $\underline{y}(t)$ is a vector valued function with values in R^n known as pseudo-state vector [Cal. & Des., 1]. The above description is known as Polynomial Model Description (PMD) and may also represented as

$$\begin{bmatrix} A(p) & -B(p) \\ -C(p) & -D(p) \end{bmatrix} \begin{bmatrix} \underline{y}(t) \\ \underline{u}(t) \end{bmatrix} = \begin{bmatrix} \underline{0} \\ -\underline{y}(t) \end{bmatrix} \Leftrightarrow T(p)\underline{x}(t) = \begin{bmatrix} \underline{0} \\ -\underline{y}(t) \end{bmatrix} \tag{2.4}$$

where T(p) is known as the Rosenbrock's System matrix [Ros., 1]. The relationship between PMDs and state space models is extensively treated in [Ros., 1], [Kail., 1]. It seems that such models, although important for electromechanical systems, are not directly relevant to process applications.

2.2.2 External models

The two basic families of external (input-output) models are the time-domain (convolution) and frequency domain (transfer function) models and are briefly presented.

(i) Convolution Models:

The time domain, input-output description of linear, causal, time invariant systems, which are assumed, to be initially relaxed, gives a mathematical description between the input and output vectors, which is expressed by [Chen, 1]

$$\underline{y}(t) = \int_0^t G(t-\tau)\underline{u}(\tau)d\tau = \int_0^t G(\tau)\underline{u}(t-\tau)d\tau \tag{2.5}$$

where t=0 is the initial time and G(t) is an mx1 matrix-valued function with G(t)=0 for t<0. The integral in (2.5) is known as a convolution integral and

the matrix G(t) as an impulse response matrix. For a proper state space model S(A,B,C,D) the impulse response matrix is expressed by

$$G(t) = Ce^{At}B+D\delta(t) \tag{2.6}$$

where $\delta(t)$ is the Dirac impulse. The above description is called a convolution description, G(t) may be obtained by carrying out experiments on the plant and thus it is of importance in the identification of a linear model (see for instance [Des. & Var., 1], [Kal., 1]. For analysis and design purposes an alternative input-output description is more suitable and it is considered next.

(ii) Transfer Function Matrix Models:

For systems which are describable by convolution integrals, it is of great advantage to use Laplace transform, because it will change a convolution integral in the time domain into an algebraic equation in the frequency domain. Thus, let $\underline{y}(s)$, $\underline{u}(s)$ be the Laplace transforms of $\underline{y}(t)$, $\underline{u}(t)$ vector functions. Then, the convolution description (2.5) becomes

$$\underline{y}(s) = G(s)\,\underline{u}(s) \tag{2.7a}$$

$$G(s) = \int_0^\infty G(t)e^{-st}dt \tag{2.7b}$$

The matrix G(s), defined as the Laplace transform of the impulse response matrix is called the system transfer function, and (2.7a) a transfer function matrix model. Whenever transfer function is used, the system is always implicitly assumed to be relaxed at t=0. For single input single output systems, (2.7a) may be written as

$$g(s) = \left.\frac{y(s)}{u(s)}\right|_{\text{relaxed at } t=0} \tag{2.7c}$$

and this demonstrates that G(s) may be experimentally determined by frequency response tests (amplitudes and phase characteristics, when the system is excited by sinusoidal signals). A transfer function is not necessarily a rational function of s. Delay terms, such us e^{-sT} and transcendental functions such as cosh(as) etc may also be included in G(s) and this indicates that

transfer functions may also be used to represent distributed-parameter characteristics, elements of process models.

A rational matrix $G(s)\epsilon R^{mxl}[s]$ is said to be <u>proper</u> if $G(\infty)$ is finite (zero, or nonzero) constant matrix and <u>strictly proper</u> if $G(\infty)=0$; otherwise, if some of the elements in $G(\infty)$ tend to infinity it will be called <u>nonproper</u>. The set of proper rational functions (the ring of proper rational functions [Vard., & Lim. & Kar., 1]) is denoted by $R_{pr}(s)$. The properness condition of a transfer function is essential for the physical realisability of a transfer function and it is a condition that should be taken into account in control design (the only possible exception is the case of derivative feedback, whenever it can be applied).

For a state space model S(A,B,C,D)the transfer function matrix is defined by

$$G(s) = C(sI-A)^{-1}B+D \tag{2.8}$$

which is a proper transfer function matrix. For every $G(s)\epsilon R_{pr}(s)^{mxl}$ there always exists a state space model S(A,B,C,D) for which (2.8) holds true [Chen, 1]; such state space models are called <u>realisations</u> of G(s) and they are not uniquely defined. A realisation of G(s) with the least possible order (dimension of A) is called <u>minimal realisation</u> and this minimal dimension is denoted by $\delta_M(G(s))$ and referred to as the <u>McMillan degree</u> of G(s) [Kal., 2].

Different approaches for working out minimal, or <u>irreducible</u> realisations of G(s) are discussed in [Chen, 1], [Kail., 1], and efficient numerical procedures exist in standard control CAD packages. Computing the McMillan degree may be achieved algebraically, or in terms of the rank properties of Hankel matrices [Chen, 1].

Certain factorisation of transfer functions which provide alternative representations of the system are the polynomial and rational fractional representations; such representations are crucial in many of the modern control synthesis-design approaches. The polynomial fractional representation is briefly presented next.

(ii.a) <u>Polynomial Matrix Fraction Descriptions</u>:

If R[s] is the set of polynomials (ring) in s variable and with real coefficients then a rational function $g(s)\epsilon R(s)$ may be expressed as $g(s)=n(s)/d(s)$, where $n(s)\epsilon R[s]$ is the numerator and $d(s)\epsilon R[s]$ is the denominator, i.e.

$$g(s) = n(s)\, d(s)^{-1} = d(s)^{-1}n(s) \tag{2.9}$$

Such a representation of g(s) is called a polynomial fractional description (R[s]-FD). Such a description is called coprime R[s]-FD if n(s), d(s) have no common zeros.
If $G(s) \epsilon R(s)^{mxl}$, then it may be also represented as

$$G(s) = N_r(s)D_r(s)^{-1} = D_l(s)^{-1}N_l(s) \qquad (2.10)$$

where $N_r(s)$, $N_l(s) \epsilon R[s]^{mxl}$, $D_r(s) \epsilon R[s]^{lxl}$, $D_l(s) \epsilon R[s]^{mxm}$ with $\begin{cases} |D_l(s)| \neq 0 \\ |D_r(s)| \neq 0. \end{cases}$
$N_r(s)$, $D_r(s)$, $N_l(s)$, $D_l(s)$ are known [Kail., 1] as R[s]-Right Matrix Fraction Descriptions (R[s]-R-MFD), R[s]-Left-Matrix Fraction Descriptions (R[s]-L-MFD) respectively. Every transfer function has R[s]-R-MFDs and R[s]-L-MFDs and such descriptions are not uniquely defined.

If $G(s)=N_r(s)D_r(s)^{-1}=D_l(s)^{-1}N_l(s)$, then $\deg\{|D_r(s)|\}$, $\deg\{|D_l(s)|\}$ is defined as the order of the R-MFD, L-MFD respectively. A R-MFD, or L-MFD is called irreducible, if $\deg\{|D_r(s)|\}$, $\deg\{|D_l(s)|\}$ is minimal amongst all other MFDs. For all irreducible MFDs (left or right), of proper transfer functions we have [Kail., 1]:

$$\min\{\deg\{|D_r(s)|\}\} = \min\{\deg\{|D_l(s)|\}\} = \delta_m(G(s)) \qquad (2.11)$$

Irreducible MFD's are not uniquely defined, but all of them provide equivalent minimal representations of G(s). The theory of MFD's is quite rich and plays a key role on the development of the modern algebraic approaches for the analysis and synthesis of multivariable control systems.

2.3 CONTROL SYSTEM ANALYSIS AND DESIGN-SYNTHESIS APPROACHES [Kar., 3]

In the study of properties of linear systems, as well as the analysis and design of control systems, a variety of approaches have been developed so far. The classification of the different approaches is based on the model that the approach uses, as well as the tools which are deployed. The aim of this section is to discuss briefly the basic characteristics and tools (mathematical and computational) of the different approaches. A diagram that summarizes the different schools of philosophy for the analysis and design of control systems is given in Figure 2.4.

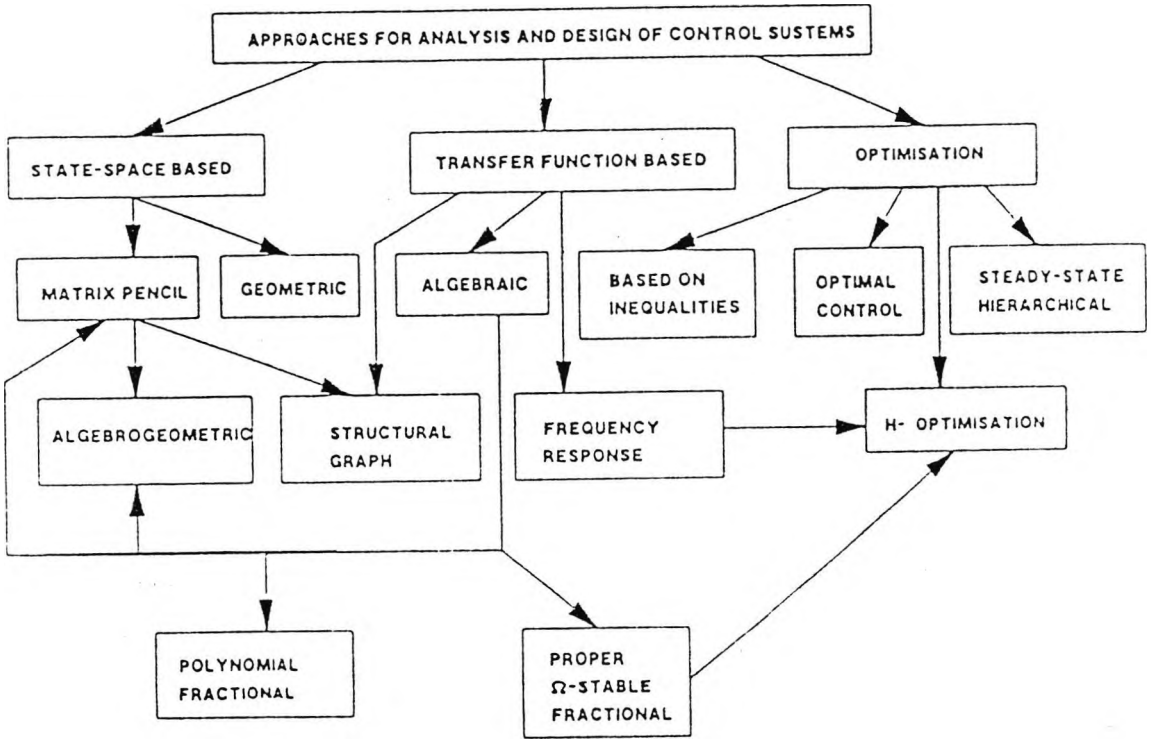


Figure 2.4

A brief description of the most important approaches is given below.

2.3.1 State space Approach

The state space approach is well developed and suited for the study of system properties such as redundancy, minimality, controllability, observability etc. Within the state space framework a variety of techniques has been developed for the solution of design problems and feedback compensation, such as pole-shifting controllers, quadratic regulator synthesis, state observers and estimators, non interactive control etc. The structural characteristics (invariants and indicators) are well developed within this framework and the computational tools are those of standard Numerical Linear Algebra. Software for computations is readily available (such as NAG Libraries, MATLAB etc.) An advantage of the approach is that the instrumentation maps (B,C) are explicitly stated in the model and thus the mechanism of exploring the formation of indicators and invariants are present, although no systematic effort has been made so far for such a study. A disadvantage of the state space approach is that it is sensitive to uncertainty about the system dynamics and cannot cope well with delay and other factors which are present in process control models. Some systematic approaches developed within the state-space framework are:

(i) **Geometric Approach**

A systematic approach for the study of state-space problems has been developed by Wonham and coworkers [Won., 1].

The geometric approach provides an elegant solution to a large family of synthesis problems; however, the solvability conditions are expressed as pure geometric conditions and thus they are not easily testable. For problems of partially fixed structure controllers (such as decentralized control) the solvability conditions become almost untestable. As a conceptual tool for the study of linear systems, this approach is of immense value; however, the pure geometric solutions are not in a suitable form for design purposes. An additional disadvantage of the approach is that it does not describe well system invariants which are mostly of an algebraic type and has difficulties in expressing the degree to which a property holds in system (aspects related to property indicators).

(ii) **Matrix Pencil Approach**

A number of difficulties of the geometric approach arise due to the fact that the algebraic aspects of the structure of the system are ignored. In fact, a variety of system properties have a natural algebraic character and an attempt to characterise algebraic concepts in pure geometric terms, creates unecessary complications. Different attempts to algebrise the geometric theory have been made. The most natural and general is the matrix pencil approach [Kar., 2] and it is based on linear polynomial matrices, known as matrix pencils [Gant., 1]. Matrix pencils are natural operators associated with first order linear ordinary differential equations and their importance for the study of linear systems has been recognised by the work of Kalman [Kal., 3], Rosenbrock [Ros., 2] etc. As a tool for characterising geometric concepts, matrix pencils were first used in [War. & Eck., 1].

The matrix pencil approach provides a complete characterisation and classification of invariants of linear state space models [Kar. & MacB., 1] and a characterisation of all families of invariant spaces of the geometric theory [Won., 1], [Wil., 1] in terms of the invariants of a space restriction pencil [Kar., 2], [Jaf. & Kar., 1]. The matrix pencil approach is well developed as an analysis tool for linear systems, but underdeveloped as a synthesis, design tool. Matrix pencils have the great advantage that describe

all aspects of state-space models, such as algebraic, geometric and computational.

2.3.2 Transfer Function Approach

The pioneering and extensive studies of Rosenbrock [Ros., 1] (and then Popov [Pop., 1], Forney [Forn., 1], Wolovich [Wol., 1] and others) clarified the power of the transfer function approach and the benefits to be gained by a better understanding of the relationships between it and the state space approach. Two main different roads have emerged from the transfer function description of a linear system, the albegraic approaches and the frequency response approaches. The first family of approaches treates the system as an operator between rational vector spaces and the basic tools are algebraic (polynomial matrix theory, integral matrices, theory of rings); the second, views the system as a map between spaces of periodic signals and thus its tools are those of complex analysis. The algebraic approaches are most suitable for addressing structural questions related to the system and synthesis problems. The complex variables approaches, on the other hand are most suitable for describing quantitative properties, design indicators and design problems.

Hybrid approaches, combining both philosophies have also recently emerged and a typical representative is the H-∞ optimization [Fra., 1]. The general futures of the different approaches are discussed below:

(i) **Algebraic Approaches**

A rational function may be always considered as the fraction of special rational functions (rings) such as the polynomials R[s] and the proper and stable rational functions $R_p(s)$. Thus, transfer function matrices may be represented in terms of matrix fractions from R[s], or $R_p(s)$. Thus, according to the type of MFD used to describe the system, we distinguish the:

(a) Polynomial Fractional Approach (PFA)

(b) Proper and Stable Fractional Approach (PSFA)

The main philosophy for analysis and synthesis is common in both of the above approaches. Different types of control problems are formulated as problems, the solution of which is reduced to a solution of a matrix equation over the rings R[s] [Kuc., 1], or $R_p(s)$ [Vid., 1]. The structure of R[s], $R_p(s)$ is similar (Euclidean rings); however, there are some fine differences, which create some divergence between the results that may be obtained. The

types of problems that may be studied are stabilisation, tracking, disturbance rejection, noninteracting control, simultaneous control of many plants, robust design etc. The central result that dominates the overall approach is that the controllers that stabilize a given plant may be analytically defined in a closed parametric form parametrisation [Kuc.,1] . The various synthesis problems are then solved within the family of stabilising controllers. For the two approaches defined above we have the following special characteristics.

(a) Polynomial Fractional Approach: The basic tools are those of polynomial matrix theory and the computations may be reduced to linear algebra problems. The computational aspects is an area where more work is needed, before the approach develops to an efficient synthesis methodologies. Advantage of the approach is the simplicity of the underlying structure of polynomials and the richness of the available results. A disadvantage of the approach is that questions of properness and generalised performance regions are handled with some difficulty.

(b) Proper and Stable Fractional Approach: The basic tools are those of matrices with elements from $R_p(s)$ ring [Vard. & Kar., 1,2]. Computations may be reduced to a polynomial framework (as before), or to state space computations. Advantage of the approach is that it can handle simultaneously questions of properness and generalised performance regions. A disadvantage of the approach is that the background mathematical tools are more complicated and the literature less rich.

As far as analysis, the first method is more suitable, whereas the second more appropriate for defining the solution of synthesis problems. It should be stressed that no ready made software is readily available for both of the approaches. An important main advantage of both approaches is that they handle easily the design problem of high complexity controllers (large dynamic order), whenever such controllers are needed.

The above two approaches deal with linear time invariant finite dimensional systems. Extensions to systems which are linear-time invariant and distributed parameter, as well as nonlinear systems exist, but not as systematic synthesis-design methodologies [Cur., & Glo., 1], [Ham., 1].

(ii) **Algebrogeometric Approach**

For the study of problems of linear systems synthesis-design which are of the determinantal type (such as pole, zero, assign ment, stabilization) a

specific school of thought has developed that is specially suited to tackle such problems. This framework is referred to as algebrogeometric, because it relies on tools from algebra and algebraic geometry. Two distinct approaches have emerged within this general framework, that is:

(i) The modern algebraic geometry approach.

(ii) The exterior algebra-classical algebraic geometry approach.

The first [Mar. & Her., 1], [Bro. & Byr., 1] consider the plant and controller as elements of algebraic varieties of an affine space and studies the solvability of pole assign ment, (by output feedback), simultaneous stabilisation etc. by using tools from modern algebraic geometry. A disadvantage of the approach is that the nongeneric cases are difficult to handle and that no systematic procedure for computing the controllers, whenever they exist, are suggested (the approach is not constructive).

The second approach [Kar. & Gia. 1,2,3], is referred to as the Determinantal Assignment Problem Approach. It has been formulated as a unifying approach for all problems of frequency assign ment (pole, zero) and its basic idea is that determinantal problems are of multilinear nature and thus they may be naturally split to a linear problem and multilinear problem (decomposability of problem of multivectors). The final solution is thus reduced to the solvability of a set of linear equations (characterising the linear problem) together with quadratics (characterising the multilinear problem of decomposability). Classical algebraic geometry (in a projective, rather than affine space) is used to determine the existence of solutions. The approach heavily relies on exterior algebra and this has implications as far as computability of solutions (reconstruction of solutions, whenever they exist) and introduction of new sets of invariants (of a projective character), which characterise the solvability of the problem.

The distinct advantages of the DAP approach, with respect to the first, are that it provides the means for computing the solutions, it can handle both generic and exact solvability investigations and introduces new criteria for the characterisation of solvability of different problems. The computation of solutions is reduced to an optimization problem of a function with quadratic equality constraints. The development of such a technique is essential for the method to become a design technique for frequency assign ment.

The DAP approach is quite promising, since it is the only one that can handle easily design problems where the compensator has a partially fixed

structure (because of decentralization and engineering constraints). It may also be used as an analysis tool for evaluating the relative merits of closing different feedback loops. As far as selection of instrumentation, the technique can be used for the selection of the gains of sensors and actuators, when their number and location has been previously specified. This approach can handle both problems of analysis and design of constant compensators. Further work and development is needed for the method to handle the case of dynamic compensators. The latter problem together with the development of an efficient numerical optimization algorithm are topics which need further consideration. In Chapter 9 of this thesis an efficient algorithm solving DAP is proposed.

2.4 SYSTEM PROPERTIES AND PROPERTY INDICATORS: DEFINITIONS AND CLASSIFICATION [Kar., 3]

Property indicators express the state, value of a certain system property, which however, may change under compensation.

Let M be the family of system models (internal or external); M will be referred to as the model set. By A we shall denote the set of all possible attributes (characteristics), that may be associated with every model $M \epsilon M$ and shall be referred to as the model attributes set. We denote by B a general set with elements, numbers, graphical statements, criteria etc, and we shall call it the criteria set.

Definition (2.1): A system property is a function $P: M \longrightarrow A$. If $P(M)$ is the image of M under P then a p-property test is a function $g: P(M) \longrightarrow \mathcal{B}$ and the composition $f: M \longrightarrow B$ defined by $f = g \circ P$ will be called a P-property indicator. ■

Example (2.1): Consider the family $M = (\dot{x} = Ax)$ of state space models. We may illustrate the definition by the following diagram

Models	Models Attributes	Property indicator	Property Criteria
$\dot{x} = Ax$	Asymptotic stability of free motion	Eigenvalues of A	Negative real parts
	Insensitivity of response to parameter variations	Ei genframe of A	orthogonality of frame

■

In simple terms, a property indicator is a function defined on the system model and whose values characterize the property. Depending on whether the model is internal, or external, the property will be referred to as internal, or external respectively.

If the attribute associated with the property expresses a qualitative property of dynamic behaviour of the system [Hir & Sma., 1], which may be defined on a general family of models, then it will be called qualitative (examples of such properties are stability, controllability, existence of periodic motions etc.).

The criteria set for such properties are of a binary nature (the model has, or does not have the property). If the attribute associated with the property has a quantitative character, that is numerical values are involved in its definition (for instance, stable step response with overshoot less than 10%) then it will be called quantitative; for such properties the criteria set is not of a binary nature, but it may contain a range of values, which express a "degree" of possessness of the property by the model.

The distinction between qualitative and quantitative properties is not clear cut; in fact a quantitative property may also have quantitative aspects (for instance controllability is a qualitative property, but assessing the energy cost of controlling a system state has a quantitative character). Frequently a property indicator may be used to assess both qualitative and quantitative properties of a given property.

A further classification of properties is in terms of the notions of genericity and robustness. If M is a family of models characterised by a common fixed structure (for instance a given linear graph), but with otherwise arbitrary parameters, then with every model $M \in M$ we may associate a parameter vector a(M) in the parameter space R^N. A property is called generic, if it holds true for almost all $M \in M$; otherwise, the subset M of M for which the property does not hold true have parameter vectors a(M) which belong to a proper variety V of the parameter space R^N [Won, 1], [Hir. & Sma., 1]

The property that is valid on a proper variety of R^N is called nongeneric. For the set of nxn real matrices, the property of having distinct eigenvalues is generic, whereas having repeated eigenvalues is a nongeneric property. A property that holds true not only for an $M \in M$, but for some neighborhood R(M) of models around M (use the parameter vectors and appropriate topology) is called well posed. If the neighborhood of models R(M) is large, the property is called robust, otherwise nonrobust. Robustness, is

thus connected to the size of permitted perturbations on the nominal model parameters before the property, that holds true on the nominal model, is violated. The gain and phase margins are typical examples of robustness measures for external stability. A generic property may also be referred to as structural. A property depending on the internal mechanism model will be called simple and if it depends on the interaction of internal mechanism and environment it will be called composite (internal stability is a simple property, but controllability is a composite property).

A property indicator that is used for assessing a single property will be called simple; if many different properties are assessed through the same indicator, then it will be called multiple. If a property indicator is an explicit, implicit function of the models parameters, then the indicator will be called explicit, implicit respectively (the controllability matrix is an explicit indicator for controllability, the Nyquist diagrams are implicit indicators for closed-loop stability). For a given property we may use two alternative indicators; such indicators used for evaluation of the same property are called equivalent (the controllability matrix and the controllability pencils are equivalent indicators, as far as assessing controllability property).

The above classification is summarized in the following diagram.

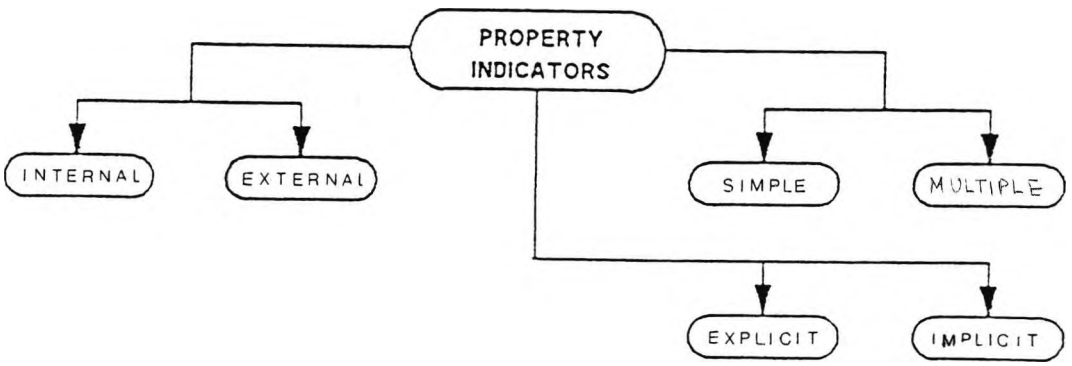


Figure 2.5

2.5 NEEDS FOR SPECIALISED ALGEBRAIC COMPUTATIONS

From the above paragraphs concerning the description of linear systems, the main design approaches and the definition and classification of property

indicators, it is apparent the need for various algebraic computations. Some of the most important computational aspects arising from Control Theory are:

(i) Computation of the Greatest Common Divisor of polynomials

The greatest common divisor of polynomials is essential in the study of problems such as multivariable zeros, controllability, observability, fixed models, solvability of Diophantine equations over R[s] etc.

(ii) Computation of the MFDs

An efficient algorithm for evaluation of irreducible right or left MFDs is very useful in the development of algebraic system theory and related control problems.

(iii) Exterior Algebra Computations

In the development of the DAP approach [Kar. & Gia., 1,2], the existence of numerical algorithms handling various exterior algebra computations are indispensable.

(iv) Computation of property indicators and invariants

When the state space approach is used, computational tools concerning the evaluation of the structural characteristics (invariants and indicators) are required. For that purpose, tools from Numerical Linear Algebra are used. The most important indicators used in the computations are:

(a) The eigenvalues of the internal dynamics matrix with their corresponding structure (as well as the poles). Eigenvalues and poles are indicators of stability.

(b) The controllability matrix, the controllability pencil, the restricted controllability pencil, the observability matrix, the observability pencil, the restricted observability pencil. All of them are equivalent indicators for the controllability, observability properties.

Since controllability, observability tests are based on the notion of rank of matrices, which generically is full, the degree nonsingularity, singularity measured by the smallest singular value, or the condition number is important indicator of "how well" the system is controllable, observable.

(c) The controllability-observability-Plucker matrices, P(A,B), P(A,C), [Kar. & Gia., 2]. These two indicators play an important key role in state space design.

2.6 CONCLUSIONS

The aim of this Chapter was to review the basic system representation, Control Analysis and Design approaches. A systematic description and classification of system properties and property indicators was also presented. Finally the most important computational problems arising in Control Theory were briefly described. As such it serves the following purposes:

(i) It provides a quick review of the various approaches and methodologies developed in the area of linear systems.

(ii) It provides a summary of some crucial computational problems of Control Theory.

CHAPTER 3

NONGENERIC INVARIANTS AND THE NEED FOR THEIR APPROXIMATE COMPUTATION

3.1 INTRODUCTION

On a given system we may apply different types of transformations, some of them corresponding to a change of representation and some others having a compensation or feedback interpretation. The theory of system invariants is important for Control Theory and design since they describe structural characteristics which remain unaffected under the transformation and thus indirectly are related to the limits of compensation.

The specific objective of this Chapter is to provide a short review concerning the definition and classification of system invariants. A systematic description of the unstructured generic systems is also presented. The generic properties of linear systems and the generic values of system invariants are also briefly developed. Finally, is explained why in many applications of Control Theory we need nongeneric computations.

3.2 SYSTEM INVARIANTS: DEFINITIONS AND CLASSIFICATION

System invariants are functions defined on the model, which remain the same under certain types of transformations; thus, they characterise not only a single model but a whole family (equivalence class).

Let M be a family of linear models, E an equivalence relation defined on M, $E(M)$ the equivalence class of $M \in M$ and let M/E be quotient set of orbit (se of all equivalence classes). We may define [MacL. & Bir., 1]:

Definition (3.1): Let M be a family of models, I a set, E an equivalence relation defined on M.

(i) A function $f: M \longrightarrow I$ is called an invariant of E, when $M_1 E M_2$ implies $f(M_1)=f(M_2)$. Also, f is called a complete invariant for E, when $f(M_1)=f(M_2)$ implies $M_1 E M_2$.

(ii) A set of invariants $\{f_i: f_i: M \longrightarrow I_i,\ i=1,2,\ldots,k\}$ is a complete set for E on M, if the map f defined by

$$f: M \longrightarrow \prod_{i=1}^{k} I_i : M \longrightarrow f(M) = \{f_1(M),\ldots,f_k(M)\}$$

is a complete invariant for E on X. The complete set of invariants is called independent, if there is no subset which is also complete.

Note that a complete invariant defines an one-to-one correspondence between $E(M)$ equivalence classes and the image of f in I. The notion of independence is essential in the minimal parametrisation of $E(M)$ by

invariants. An important issue for system identification and control analysis is that of canonical form for $E(M)$.

If $f: M \longrightarrow \times_{i=1}^{k} I_i$ is a complete and independent invariant for E on

M by specializing the invariant f such that its image C is in M, we define a canonical element or a canonical form

■

Definition (3.2): A set of canonical forms, C for E equivalence on M, is a subset of M such that for every $M \epsilon M$ there exists a unique $C \epsilon C$ for which $M E C$.

■

Canonical forms are uniquely defined elements of M, which have the simplest possible structure (least number of parameters) and which describe the invariant in the language of the model (in terms of a simple model).

Canonical forms, are often used as analysis tools and describe the simplest possible type of model that may be defined under the set of transformations defining the equivalence relation.

The set of canonical forms provides a system of canonical distinct representatives for M/E.

The classification of invariants to internal / external, simple / composite, explicit / implicit is the same to that given for properties. An invariant will be called global, if it takes nontrivial values for all $M \epsilon M$; otherwise, that is, it takes nontrivial values only on a proper variety of the model parameter space R^N, it will be called local.

The value of a global invariant will be called generic, if it is constant for almost all $M \epsilon M$. That is the models for which the value may differ from the constant is a proper variety of R^N, such values will be called nongeneric.

An invariant of representation transformations will be referred to as a representation invariant, whereas those of compensation transformations will be called a compensation invariant. An invariant will be called strong, or weak, if it is preserved or not preserved under more general types of transformation.

The following diagram summarizes the classification of system invariants.

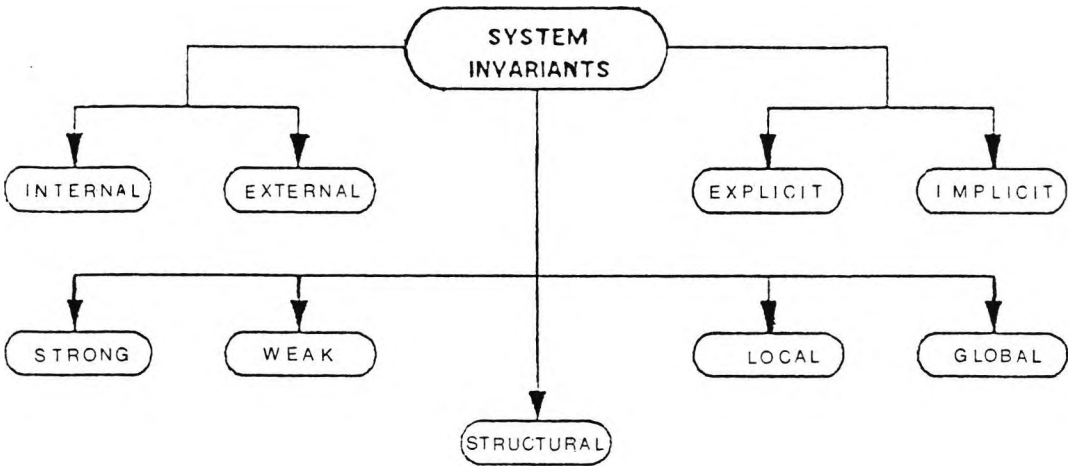


Figure (3.1)

Since a variety of invariants and canonical forms are defined, the need for computing algebraically some of them arises. The most fundamental types of invariants that require algorithms for their computation are:

1) The Jordan canonical form that may be computed algebraically by use of the Smith form (computation of set of ed's)
2) The Kronecker canonical form
3) The Smith McMillan forms over a ring
4) The Hermite, Hermite McMillan forms
5) In the study of determinantal assignment problems (DAP) (pole, zero, assignment) alternative forms of invariants are essential such as the Plucker matrices [Kar. & Gia., 2].

3.3 UNSTRUCTURED GENERIC SYSTEMS [Kar., 3]

The problem of model uncertainty and its effect on control design is one of the important issues in contemporary Control Theory and Design Model uncertainty may be structured, or unstructured. By structured uncertainty we mean that certain model parameters vary within certain range, whereas by unstructured uncertainty we mean that certain property indicators vary within a certain range of values. Studying the effect of uncertainty on the preservation of certain properties of the nominal model is referred to as "robustness analysis". A related topic, that is the study of properties of a whole family of models having fixed certain fundamental parameters (such as

numbers of inputs, states, outputs, McMillan degree), but with the rest of the parameters taking generic values is referred to as structural analysis of unstructured models and the main issue here is the study of properties that hold for the generic element of the family.

We consider unstructured linear systems described by state space or transfer function type models. The family of systems which are considered are the linear, proper time invariant. The unstructured assumption about the model implies that there is no special assumption about the structure of interconnections between the input, output and state variables of the model and that the corresponding parameters take generic values. The unstructured model assumption has different implications for state space and transfer function models. Thus,

(i) For state space models S(A,B,C,D), it is assumed that the number of inputs (l), outputs (m), states (n) are fixed, but the parameters in A,B,C,D matrices are generic. This family is denoted by $\Gamma(l,m,n)$.

(ii) For transfer function models G(s), where G(s) is proper, it is assumed that the number of inputs (l), outputs (m) are fixed, but the $g_{ij}(s)$ elements of G(s) are generic proper rational functions. This family is denoted by $\Sigma_{pr}(l,m)$.

The $\Gamma(l,m,n)$, $\Sigma_{pr}(l,m)$ families of generic systems are not equivalent since the McMillan degree of the elements of $\Sigma_{pr}(l,m)$ may vary. In the following, it is natural to examine separately the results concerning the two families.

3.3.1 Generic Properties of linear systems

Under the genericity assumptions stated above the system properties are considered here and classified to generic and nongeneric. We summarize the basic properties by the following result [Won., 1].

Result (3.1): For generic S(A,B,C,D) system of the $\Gamma(l,m,n)$ family, the following properties hold true:

(i) The set of eigenvalues of A are generically distinct and complex, where the complex appear in complex conjugate pairs. Thus, A is generically cyclic and diagonalisable (under similarity transformations).

(ii) The system is genericaly controllable and observable and thus also stabilisable and detectable.

■

The existence of Jordan forms, as well as uncontrollability, unobservability are nongeneric properties for systems of $\Gamma(l,m,n)$. Some further properties related to poles and zeros are listed below.

Result (3.2): For generic systems of the $\Gamma(l,m,n)$, $\Sigma_{pr}(l,m)$ families the following properties hold true:

(i) The transfer function matrix has full rank (equal to $\min\{m,l\}$ over $R(s)$). If $m \leq l$, then the generic system is output function controllable.

(ii) If $m \neq l$ the system has no finite zeros. If $m=l$, the generic proper system has no finite zeros and the generic strictly proper system has n-m finite zeros.

(iii) The number of elementary divisors at infinity of the generic S(A,B,C,D) system is equal to $\min\{m,l\}$; furthermore, if $D \neq 0$ (proper) then all such divisors are linear and if $D=0$ (strictly proper) then all such divisors are of the $\hat{s}^2$ type.

(iv) The generic element of $\Sigma_{pr}(l,m)$ has no infinite zeros. If the system is strictly proper, then the generic system of $\Sigma_{pr}(l,m)$ has $\min\{m,n,l\}$ number of first order infinite zeros.

■

Properties such as stability, instability they cannot be inferred from genericity arguments since they depend on the parameters of the A matrix. The concept of zeros is generic for square systems and nongeneric for nonsquare systems. For proper square systems the generic number of finite zeros is equal to the number of states, whereas for strictly proper square systems we always have m first order infinite zeros and n-m finite zeros. Genericity arguments cannot infer minimum, or nonminimum phase properties. The presence of higher order (than one) infinite zeros is a nongeneric property thus, the generic asymptotic pattern of root locus is rather simple and can be compensated. For strictly proper square systems, the generic numbers of finite zeros, n-m, is a measure of difference between states and inputs (outputs). By increasing the number of inputs, outputs for such systems, we decrease the number of finite zeros and increase the number of first order infinite zeros and vice versa.

3.3.2 Generic values of system invariants

The values of the invariants for generic systems of the unstructured $\Gamma(l,m,n)$, $\Sigma_{pr}(l,m)$ families are important elements in the solvability conditions of exact control synthesis problems and they are examined here.

Remark (3.1): The generic number of finite and infinite zeros, as well as the values of orders of infinite zeros have been defined in the section (3.3.1). For a generic system the finite zeros are distinct and they appear in pairs of complex conjugate zeros.

■

For state space models, the remaining invariants are those defined by column, row minimal indices of corresponding matrix pencils. The following result [Kar. & Had., 1] provides the means for the characterisation of generic values of invariants of state space models.

Result (3.3): Let $F,G\epsilon R^{mxn}$, $m<n$, and assume that (F,G) is generic (that is the matrices, F,G are generic). For the generic pencil sF-G the following holds true:

(i) sF-G has only column minimal indices (cmi); that is it has no finite, infinite e.d. and no row minimal indices (rmi).

(ii) If $p=\min\{k\ \epsilon Z^{+}:p>m/(n-m)\}$ then the set $I_c(F,G)$ of cmi of the generic sF-G is defined by:

(a) If $p=n/(n-m)$ then $I_c(F,G)=((\varepsilon_1,\eta_1):\varepsilon_1=p-1,\ \eta_1=n-m)$; that is it has one cmi ε_1 with multiplicity η_1.

(b) If $p\neq n/(n-m)$, then $I_c(F,G)=((\varepsilon_1,\eta_1),\ (\varepsilon_2,\eta_2):\varepsilon_1=p-1,\ \eta_1=p(n-m)-m,\ \varepsilon_2=p,\ \eta_2=n-p(n-m))$; that is we have two values ε_1, ε_2 with corresponding multiplicities η_1, η_2.

■

A similar result may be stated for pencils with $m>n$ and this defines the generic values of row minimal indices . From the above result and its dual we have:

Result (3.4): For the generic system of $\Gamma(l,m,n)$, the generic values of controllability indices I_c and observability indices I_o are defined by:

(i) If p is the smallest integer such that $p>n/l$, then

(a) If $p=n/l+1$, then $I_c=\{(\mu_1,\eta_1):\mu_1=n/l,\ \eta_1=l\}$

(b) If $p \neq n/l+1$, then $I_c=\{(\mu_1,n_1), (\mu_2,n_2), \mu_1=p-1,$
$n_1=pl-n, \mu_2=p, n_2=n+l-pl\}$.

where μ_i are distinct values of controllability indices and n_i are the corresponding multiplicities.

(ii) If p is the smallest integer such that $p>n/m$, then

(a) If $p=n/m+1$, then $I_o=\{(\tau_1,\sigma_1):\tau_1=n/m, \sigma_1=m\}$

(b) If $p \neq n/m+1$, then $I_o=\{(\tau_1,\sigma_1),(\tau_2,\sigma_2):\tau_1=p-1,$
$\sigma_1=pm-n, \tau_2=p, \sigma_2=n+m-pm\}$

where τ_i are distinct values of observability indices and σ_i are the corresponding multiplicities.

■

An equivalent statement of the above result may be found in [Won., 1]. This result established the generic values of the controllability and observability indices.

Remark (3.2): The generic value of the controllability, observability indices μ, τ are defined by:

(a) μ is the smallest integer for which $\mu \geq n/l$,
(b) τ is the smallest integer for which $\tau \geq n/m$.

■

For the types of invariants defined in terms of Grassmann vectors, or Plucker matrices, the generic results are summarized below [Kar. & Gia., 2]:

Result (3.5): The controllability, observability Plucker matrices P(A,B), P(A,C) of a generic system of $\Gamma(l,m,n)$ have full rank. The Plucker matrices $P_c(G)$ (if $m>l$), $P_r(G)$ (if $m<l$), as well as $P(T_l)$, $P(T_r)$ have also full rank for a generic system of $\Sigma_{pr}(l,m)$.

■

3.4 THE NEED FOR GENERIC AND NONGENERIC COMPUTATIONS

From the above it is apparent that there is a need for the computation of system invariants which may be generic, or nongeneric on a given family of unstructured linear systems models. We shall refer to the computations of nongeneric invariants, as "nongeneric computations", whereas those of generic invariants, as "generic computations". The issues involved in the two cases are different and deserve some attention.

In the case of nongeneric computations, such as for instance the computation of zeros of nonsquare systems, the greatest common divisor (g.c.d.) of polynomials etc, the attention has to be focussed on the appropriate termination of the computational algorithm that will allow the "catching up" of the approximate solutions. The accuracy of the original data determines the threshold, where an algorithm has to terminate and give an approximate solution and where it has to continue. In the case of generic computations, such as eigenvalues, attention is focussed on the computational accuracy of the functions, rather than the existence, or non existence of values.

It is implicit from the above, that an integral part of the derivation of procedures for nongeneric computations is the relaxation of certain algebraic definitions, and their embedding in an analytical set up. Thus, a g.c.d. of polynomials has to be relaxed to that of "almost zeros" of polynomials, the concept of dependence or independence of vectors to that of "almost dependence, or independence" etc. Appropriate tools have to be devise to indicate degree of presence, or distance from strong possession of a certain property.

The issues related to generic computations are well developed in the appropriate literature. The aim of this thesis is to address issues related to nongeneric computations, such as development of tools for diagnosis of certain properties in an "almost sense", definition of procedures for termination of algorithms, which if they are let to run for sufficient number of iterations will converge to generic values, and address a number of specific problems such as the computation of g.c.d. etc.

Since in Control Theory very frequently nongeneric computations are required, in Chapter 5 useful tools for handling such computations are analytically described.

3.5 CONCLUSIONS

The aim of this Chapter was to review the basic definitions and classification concerning system invariants and thus provide some motivation for the computational issues addressed in this thesis. A brief description of unstructured generic systems was also presented. As such it serves the following purposes:

(i) It provides a quick review of the generic properties of linear systems and of the generic values of system invariants

(ii) It illustrates the need for introducing nongeneric computations.

CHAPTER 4

EXTERIOR ALGEBRA COMPUTATIONS

4.1 INTRODUCTION

The present Chapter is concerned with a major computational problem arising mostly from the DAP approach of LDMs, the Exterior Algebra computations. In the beginning, a brief summary of definitions and background results from Exterior Algebra are presented. The important notions of exterior product of vectors and compound matrix are formulated. In the sequel, useful definitions and algorithms concerning sequences of p integers out of m are developed. The notions of P-Prime and N-Prime sequences are defined and algorithms appropriate for their evaluation are suggested.

Next, algorithms suitable for the evaluation of compound matrices are described. The case of real matrices is considered first and then the complicated case of polynomial matrices. For a given $M(s)\epsilon R^{mxl}[s]$, $m\geq l$ two algorithms for the evaluation of $C_l(M(s))$ are proposed. The one is based on the notion of P-Prime sequences and the second on the notion of N-Prime sequences. An algorithm for the evaluation of $C_p(M(s))$, $1\leq p<l$ is also described. Moreover, the computation of Smith-Normal form of a polynomial matrix based on compound matrices is developed.

Finally, the problem of computing Plucker matrices is also encountered and an efficient algorithm attaining this evaluation is suggested.

Most of the algorithms developed in this Chapter will be used in Chapter 9, in order to derive an efficient algorithm for the solution of DAP.

Throughout the Chapter, several numerical examples illustrating the application of the corrresponding algorithms are presented.

4.2 EXTERIOR ALGEBRA: BACKGROUND RESULTS

4.2.1 Exterior powers of a vector space

Let V be an arbitrary vector space and $p\geq 2$ be an integer. Then a vector space $\Lambda^p V$ together with a skew symmetric p linear map

$$\Lambda^p: \prod_{i=1}^{p} V \longrightarrow \Lambda^p V: (\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_p) \xrightarrow{\Lambda^p} \underline{x}_1\wedge\underline{x}_2\wedge\ldots\wedge\underline{x}_p \tag{4.1}$$

is called a p-th exterior power of V if the following conditions are satisfied:

(i) The vectors $\Lambda^p(\underline{x}_1,\ldots,\underline{x}_p)$ generate $\Lambda^p V$

(ii) If ψ is any skew symmetric p-linear map of $\underset{i=1}{\overset{p}{x}} V$ into an arbitrary vector space U, then there exists a linear map:

$$f:\Lambda^p V \longrightarrow U \text{ such that } \psi = f \circ \Lambda^p .$$

Suppose that V is a vector space of dimension n over the field F (R or C). Then the p-th exterior power of V, $\Lambda^p V$ is a vector space. If $\underline{e}_i$, i=1,2,...,n is a basis of V then the products

$$\underline{e}_{i_1} \wedge \underline{e}_{i_2} \wedge \ldots \wedge \underline{e}_{i_p}, \quad 1 \leq i_1 \leq i_2 < \ldots \leq i_p \leq n$$

span the vector space $\Lambda^p V$. It can be proved that $\underline{e}_{i_1} \wedge \underline{e}_{i_2} \wedge \ldots \wedge \underline{e}_{i_p}$ are linearly independent and thus form a basis for $\Lambda^p V$. Clearly then

$$\dim \Lambda^p V = \binom{n}{p}, \quad p = 0,1,2,\ldots,n$$

and $\Lambda^p V = 0$ for p>n. An element of the form $\underline{x}_1 \wedge \underline{x}_2 \wedge \ldots \wedge \underline{x}_p$ where $\underline{x}_1, \ldots, \underline{x}_p \in V$ is called <u>decomposable</u>.

4.2.2 <u>Exterior powers of linear maps</u>

<u>Theorem (4.1)</u> [Marc. & Minc, 1]: Let V,U be finite dimensional vector spaces over a field F, and let $h:V \longrightarrow U$ be a linear map.
Then, there is a unique homomorphism $\hat{h}:\Lambda V \longrightarrow \Lambda U$ of the exterior algebras such that $\hat{h}(\underline{x}) = h(\underline{x})$ for all $\underline{x} \in V$. Notice that $\hat{h}$ maps $\Lambda^p V$ to $\Lambda^p U$ for all p. ■

The above result simply means the following: If h is a linear map of a vector space V into a vector space U over F, then the map

$$\psi: \underset{i=1}{\overset{p}{x}} V \longrightarrow \Lambda^p U : (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p) \xrightarrow{\psi} h(\underline{x}_1) \wedge h(\underline{x}_2) \wedge \ldots \wedge h(\underline{x}_p) \tag{4.2}$$

defines an alternating multilinear map of $\underset{i=1}{\overset{p}{x}} V$ into $\Lambda^p U$.
By the definition of the exterior product there exists a unique linear map h of $\Lambda^p V$ into $\Lambda^p U$ such that:

$$h:\Lambda^p V \longrightarrow \Lambda^p U:\underline{x}_1\wedge\underline{x}_2\wedge\ldots\wedge\underline{x}_p \xrightarrow{h} h(\underline{x}_1)\wedge h(\underline{x}_2)\wedge\ldots\wedge h(\underline{x}_p) \tag{4.3}$$

We write $\Lambda^p h$ for h and we call it the <u>p-th exterior power of the linear map h</u>.

4.2.3 <u>Representation theory of exterior powers of linear maps</u> [Kar., 4]

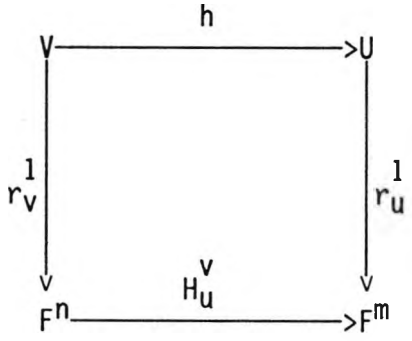
Let V be an n-dimensional vector space over the field F and $\Lambda^p V$, $p\leq n$ be the p-th exterior power of V. If $\{\underline{v}_i,\ i=1,2,\ldots,n\}$ is a basis of V, then $\Lambda^p V$ is spanned by the vectors of the basis $\{\underline{v}_\omega:\omega=(i_1,i_2,\ldots,i_p),\ 1\leq i_1\leq\ldots\leq i_p\leq n,\ \underline{v}_\omega=\underline{v}_{i_1}\wedge\underline{v}_{i_2}\wedge\ldots\wedge\underline{v}_{i_p}\}$.

Every vector $\underline{v}\in\Lambda^p V$ can be written as $\underline{v}=\{\sum_\omega a_\omega\underline{v}_\omega\}$. Let the map:

$$r_V^p:\Lambda^p V\longrightarrow R^{\binom{n}{p}}:\underline{v}\longrightarrow r_V^p(v)=[\ldots,a_\omega,\ldots]^t \tag{4.4}$$

then r_V^p is linear and it is called the <u>representation map</u> of $\Lambda^p V$ associated with the basis $\{\underline{v}_i,\ i=1,2,\ldots,n\}$.

Let V,U be vector spaces over F of dimensions, n,m, respectively and let h be a linear map of V into U. The linear map h can be represented with respect to the bases $B_V=\{\underline{v}_i:i=1,2,\ldots,n\}$, $B_U=\{\underline{u}_i:i=1,2,\ldots,m\}$ of V and U by a matrix $H_u^v\in F^{m\times n}$ which is defined by the following commutative diagram



<u>Fig. 4.1</u>

Applying the representation result for linear maps for the linear map $\Lambda^p h:\Lambda^p V\longrightarrow\Lambda^p U$ we have the following commutative diagram

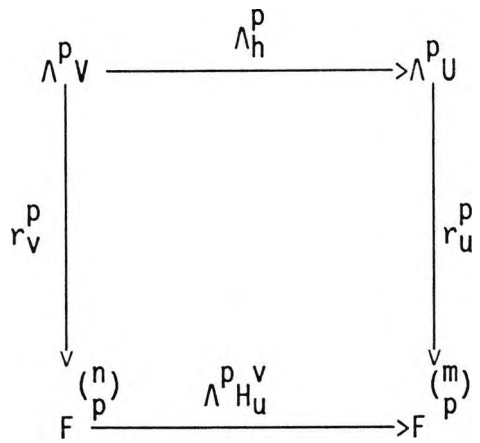


Fig. 4.2

The description of $\Lambda^p H_u^v \epsilon F^{\binom{m}{p}\times\binom{n}{p}}$ in terms of H_u^v will be defined in 4.2.4 by using the notion of the compound matrix.

4.2.4 Compound matrices and Grassmann products [Marc. & Minc, 1].
Some useful notation and definitions on the sequences of integers and on submatrices of a given matrix are stated next.

(i) **Notation (4.1)**:
(a) $Q_{p,n}$ denotes the set of strictly increasing sequences of p integers ($1\leq p\leq n$) chosen from $1,\ldots,n$, e.g. $Q_{2,3}=\{(1,2), (1,3), (2,3)\}$.
Thus, the number of the sequences which belong to $Q_{p,n}$ is $\binom{n}{p}$.
If $\alpha,\beta\epsilon Q_{p,n}$ we say that α precedes β ($\alpha<\beta$), if there exists an integer t ($1\leq t\leq p$) for which $\alpha_1=\beta_1,\ldots,\alpha_{t-1}=\beta_{t-1}$, $\alpha_t<b_t$, where α_i,β_i denote the elements of α,β respectively, e.g. in the set $Q_{3,8}$ $(3,5,8)<(4,5,6)$. This describes the lexicographic ordering of the elements of $Q_{p,n}$. The set of sequences $Q_{p,n}$ from now on will be assumed with its sequences lexicographically ordered and the elements of the ordered set $Q_{p,n}$ will be denoted by $Q_{p,n}(t)$, $t=1,\ldots,\binom{n}{p}$ or simply by ω.

(b) $Q_{p,n}^{\alpha)}$ denotes the subset of $Q_{p,n}$ whose sequences do not contain any of the indices of a given $\alpha\epsilon Q_{p,n}$, e.g.

$Q^{\alpha)}_{2,5} = \{(3,4), (3,5), (4,5)\}$, if $\alpha=(1,2)$

This set has $\binom{n-p}{p}$ elements and will be also assumed to be lexicographically ordered. The elements of $Q^{\alpha)}_{p,n}$ will be denoted by $Q^{\alpha)}_{p,n}(t)$, where $t=1,\ldots,\binom{n-p}{p}$, or simply by $\omega_{\alpha)}$.

(c) If $c_1,\ldots,c_n$ are elements of the field F and $\omega=(i_1,i_2,\ldots,i_p)$ is a sequence in $Q_{p,n}$, $1\leq p\leq n$, then the product $c_{i_1}c_{i_2}\ldots c_{i_p}$ will be designated by c_ω.

(d) Suppose $A=[a_{ij}]\in F^{m\times n}$, let k,p be positive integers satisfying $1\leq k\leq m$, $1\leq p\leq n$ and let $\alpha=(i_1,\ldots,i_k)\in Q_{k,m}$ and $\beta=(j_1,\ldots,j_p)\in Q_{p,n}$. Then $A[\alpha|\beta]\in F^{k\times p}$ denotes the submatrix of A which contains the rows $i_1,\ldots,i_k$ and the columns $j_1,\ldots,j_p$. We use the notation $A(\alpha|\beta]$ to designate the submatrix of A which excludes rows $i_1,\ldots,i_k$ and includes columns $j_1,\ldots,j_p$. The submatrices $A[\alpha|\beta)$ and $A(\alpha|\beta)$ can be defined similarly.

(ii) **Compound matrices**

Let $A\in F^{m\times n}$ and $1\leq p\leq \min\{m,n\}$, then the p-th compound matrix or p-th adjugate of A is the $\binom{m}{p} \times \binom{n}{p}$ matrix whose entries are $\det(A[\alpha|\beta])$, $\alpha\in Q_{p,m}$, $\beta\in Q_{p,n}$ arranged lexicographically in α and β. This matrix will be designated by $C_p(A)$. For example if $A\in F^{3\times 3}$ and p=2, then $Q_{2,3}=\{(1,2), (1,3), (2,3)\}$ and

$$C_2(A)=\begin{bmatrix} \det(A[(1,2)|(1,2)]) & \det(A[(1,2)|(1,3)]) & \det(A[(1,2)|(2,3)]) \\ \det(A[(1,3)|(1,2)]) & \det(A[(1,3)|(1,3)]) & \det(A[(1,3)|(2,3)]) \\ \det(A[(2,3)|(1,2)]) & \det(A[(2,3)|(1,3)]) & \det(A[(2,3)|(2,3)]) \end{bmatrix}$$

or setting for convenience $\det(A[\alpha|\beta])=a^{\alpha}_{\beta}$ we have

$$C_2(A)=\begin{bmatrix} a_{1,2}^{1,2} & a_{1,3}^{1,2} & a_{2,3}^{1,2} \\ a_{1,2}^{1,3} & a_{1,3}^{1,3} & a_{2,3}^{1,3} \\ a_{1,2}^{2,3} & a_{1,3}^{2,3} & a_{2,3}^{2,3} \end{bmatrix} \tag{4.5}$$

Properties of compound matrices

(a) If $A\epsilon F^{nxn}$, $1\leq p\leq n$ and also A is non-singular

(i) $[C_p(A)]^{-1} = C_p(A^{-1})$ (4.6)

(ii) $C_p(A)=[C_p(A)]^*$, where A^* is the conjugate transpose of A(F=C) (4.7)

(iii) $C_p(A^T)=[C_p(A)]^T$ (4.8)

(iv) $C_p(\bar{A}) = \overline{C_p(A)}$, (4.9)

(v) $C_p(kA) = k^pC_p(A)$, for any $k\epsilon F$ (4.10)

(vi) $C_p(I_n) = I_{\binom{n}{p}}$ (4.11)

(vii) $\det(C_p(A)) = (\det A)^{\binom{n-1}{p-1}}$ Sylvester-Franke Theorem (4.12)

(viii) $C_n(A)=\det A$ (4.13)

(b) If $A\epsilon F^{mxn}$ and $B\epsilon F^{nxk}$ and $1\leq p\leq \min\{m,n,k\}$, then

$$C_p(AB) = C_p(A)C_p(B) \quad \text{Binet-Cauchy Theorem} \tag{4.14}$$

(c) If $A\epsilon F^{pxn}$ and the p rows of A are denoted by $\underline{a}_1^{-t},\ldots,\underline{a}_p^{-t}$ in succession ($1\leq p\leq n$), then $C_p(A)$ is an $\binom{n}{p}$-tuple and it is called the Grassmann product or skew symmetric product of the vectors $\underline{a}_1^{-t},\ldots,\underline{a}_p^{-t}$ for reasons which will become apparent later on. The usual notation for this $\binom{n}{p}$-tuple of subdeterminants of A is $\underline{a}_1^{-t}\wedge\ldots\wedge\underline{a}_p^{-t}$ and it denotes a row

vector. The Grassmann product of the columns of a matrix $A\epsilon F^{nxp}$ $(1\leq p\leq n)$ may be defined in a similar manner; the product in this case, however, will be an $\binom{n}{p}$-column vector. If $\underline{a}_1,\ldots,\underline{a}_p$ are the columns of A, then in this case this $\binom{n}{p}$-tuple of subdeterminants of A will be denoted by $\underline{a}_1\wedge\ldots\wedge\underline{a}_p$. By the properties of determinants, if $\sigma\epsilon S_p$ (where S_p denotes the totality of permutations of $1,\ldots,p$), then

$$\underline{a}_{\sigma(1)}\wedge\ldots\wedge\underline{a}_{\sigma(p)} = \operatorname{sign}\sigma\ \underline{a}_1\wedge\ldots\wedge\underline{a}_p \tag{4.15}$$

If $B\epsilon F^{nxn}$, $A\epsilon F^{nxp}$, then by the Binet-Cauchy theorem it follows that

$$C_p(B)\underline{a}_1\wedge\ldots\wedge\underline{a}_p = B\underline{a}_1\wedge\ldots\wedge B\underline{a}_p \tag{4.16}$$

Grassmann products suitably deployed may greatly reduce the complexity of the expressions in compound matrices. Thus, let $A\epsilon F^{mxn}$ and $1\leq p\leq\min\{m,n\}$. The matrix A may be written in terms of its rows or columns respectively as

$$A = \begin{bmatrix} \overset{-t}{\underline{a}_1} \\ \cdot \\ \cdot \\ \cdot \\ \overset{-t}{\underline{a}_m} \end{bmatrix} \qquad \text{or } A = [\underline{a}_1,\ldots,\underline{a}_n] \tag{4.17}$$

Let $\omega=(i_1,\ldots,i_p)\epsilon Q_{p,m}$ and $\Phi=(j_1,\ldots,j_p)\epsilon Q_{p,n}$ and let us denote by $\overset{-t}{\underline{a}_\omega}$ the Grassmann product $\overset{-t}{\underline{a}_{i_1}}\wedge\ldots\wedge\overset{-t}{\underline{a}_{i_p}}$ and by $\underline{a}_\Phi$ the Grassmann product $\underline{a}_{j_1}\wedge\ldots\wedge\underline{a}_{j_p}$. The p-th compound matrix of A may then be expressed in either of the following forms

$$C_p(A)=\begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \overset{-t}{\underline{a}_\omega} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}, \quad \omega\epsilon Q_{p,m} \quad \text{or} \quad C_p(A)=[\ldots,\underline{a}_\Phi,\ldots], \ \varphi\in Q_{p,n} \tag{4.18}$$

which will be referred to as row, columns representations of $C_p(A)$

respectively. If $\{\underline{x}_{i_1},\ldots,\underline{x}_{i_k}\}$ is a set of vectors of V, $\omega=(i_1,i_2,\ldots,i_k)$ $\epsilon Q_{k,n}$,then $\underline{x}_{i_1}\wedge\ldots\wedge\underline{x}_{i_k}=\underline{x}_\omega\wedge$ denotes their exterior product and $\Lambda^p V$ denotes their exterior product and $\Lambda^p V$ denotes the r-th exterior power of V. The matrix $\Lambda^p H_u^v$ of (Fig. 2) is exactly the $C_p(H_u^v)$.

4.3 COMPUTATION OF SEQUENCES

All the algorithms and techniques required for handling problems of exterior algebra are based mostly on sequences of integers. Therefore, it is indispensable to develop separately few things concerning generally sequences of integers.

Very frequently, in our evaluations it is required for given integers p,m to find the set $Q_{p,m}$ defined in 4.2.2. Next an algorithm evaluating all the sequences $\omega\epsilon Q_{p,m}$ is presented.

Algorithm CONSEQ

For given integers m,p, $1\leq p\leq m$, the following algorithm produces the elements $\omega=(\omega_1,\omega_2,\ldots,\omega_p)\epsilon Q_{p,m}$

```
STEP 1: for i = 1,...,p
            ω_i := i
        Print ω
STEP 2: while ω_1<m-p+1
            k : =0
            while ω_{p-k} = m-k
                k : =k+1
            ω_{p-k} := ω_{p-k}+1
            for i = p-k+1,...,p
                ω_i := ω_{i-1}+1
            Print ω
```

Alg: 4.1

Implementation of the algorithm

Each new sequence of $Q_{p,m}$ is printed immediately after its construction. In this way, we need only one array of dimension p.

In case that we would like to keep in the memory all the $\binom{m}{p}$ sequences of $Q_{p,m}$, we would need $\binom{m}{p}$ arrays of dimension p each. For large m, this is utterly unaffective and it is not recommended unless we are obliged to do so from the requirements of our problem.

Example (4.1): For m=5, p=3 algorithm **CONSEQ** prints the following sequences $\omega \epsilon Q_{3,5}$.

$\omega=(1,2,3)$, $\omega=(1,2,4)$, $\omega=(1,2,5)$, $\omega=(1,3,4)$,

$\omega=(1,3,5)$, $\omega=(1,4,5)$, $\omega=(2,3,4)$, $\omega=(2,3,5)$,

$\omega=(2,4,5)$, $\omega=(3,4,5)$.

■

As it will be developed in the sequel, during the evaluation of compounds of polynomial matrices many zero entries are appeared. Therefore, it is required to have the ability of choosing out of a given $Q_{p,m}$, some concrete sequences satisfying certain properties and corresponding to the nonzero entries. In order to accomplish this, we need some more notation and definitions.

Notation (4.2): For given $k,q \epsilon Z^+$ let us denote by $N_{k,q}=\{1,2,3,\ldots,q,q+1,\ldots,2q,\ldots,kq\}$ the set containing all the integers starting from 1 till a specified multiple (kq) of q. $N_{k,q}$ can be partitioned into q subsets with k elements each. Thus, we can obtain the set: $P_{q,kq}=\{P_1,P_2,\ldots,P_q\}$, where each P_i, i=1,2,...,q is of the form:

$P_i=(p_{i_1},p_{i_2},\ldots,p_{i_k})$ with $p_{i_j}=(i-1)k+j$, j=1,2,...,k.

■

Definition (4.1): Let $k,q \epsilon Z^+$ be given integers.

(i) For each $P_i \epsilon P_{q,kq}$, i=1,2,...,k we define as weight of P_i

$$w(P_i) \equiv q-i$$

(ii) For each integer $n \epsilon N_{k,q}$ there exists a $P_r \epsilon P_{q,kq}: n \epsilon P_r$. The weight of n with respect to $P_{q,kq}$ partitions, is defined by

$$w(n) \equiv w(P_r)$$

(iii) The order of n with respect to $P_{q,kq}$ partitions, is defined by $\sigma(n)$ which is the order n appears in

$$P_r=(p_{r_1},p_{r_2},\ldots,p_{r_k})$$

(iv) For $\omega=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_k}) \epsilon Q_{k,kq}$ we define as order of ω

$$O(\omega) \equiv (\sigma(\omega_{i_1}),\sigma(\omega_{i_2}),\ldots,\sigma(\omega_{i_k}))$$

(v) $\omega \epsilon Q_{k,kq}$ is P-Prime if $O(\omega)$ has distinct elements.

(vi) If ω is P-Prime, we define as sign of ω

$$sgn(\omega) \equiv sgn\pi(\sigma(\omega_{i_1})\sigma(\omega_{i_2})\ldots\sigma(\omega_{i_k}))$$

where $sgn\pi(\sigma(\omega_{i_1})\sigma(\omega_{i_2})\ldots\sigma(\omega_{i_k}))$ is the sign of the permutation

$\pi(\sigma(\omega_{i_1})\ \sigma(\omega_{i_2}))\ldots\sigma(\omega_{i_k}))$.

(vii) If ω is P-Prime, then we define as weight of ω

$$W(\omega) \equiv \sum_{j=1}^{k} w(\omega_{i_j})$$

■

Remark (4.1): Parts (iv), (v), (vi) and (vii) of the above definition can hold for any sequence $\omega \epsilon Q_{p,kq}$ with $1 \leq p < k$. Particularly in this case, for a given sequence $r=(r_{i_1},r_{i_2},..,r_{i_p}) \epsilon Q_{p,k}$ the sequence $\omega=(\omega_{j_1},\omega_{j_2},\ldots,\omega_{j_p}) \epsilon Q_{p,kq}$ is said to be r-Prime if

$$\forall\ \omega_{j_1},\ l=1,2,\ldots,p\ \ \exists\ r_{i_m},\ m=1,2,\ldots,p:\ O(\omega_{j_l})=r_{i_m}$$

■

All the above notions will be applied in section 4.5 in order to compute effectively the compound of a given polynomial matrix.

Example (4.2): Let k=5, q=3 be given integers.

$N_{5,3}=\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$

$P_{3,5\cdot 3}=P_{3,15}=\{P_1,P_2,P_3\}$ with

$P_1=(1,2,3,4,5)$, $w(p_1)=2$

$P_2=(6,7,8,9,10)$, $w(p_2)=1$

$P_3=(11,12,13,14,15)$, $w(p_3)=0$

For an integer $n=7\epsilon N_{5,3}$ ∄ $P_2\epsilon P_{3,15}$: $7\epsilon P_2$

$$w(7) = w(P_2) = 1, \quad \sigma(7) = 2$$

For any sequence $\omega\epsilon Q_{5,15}$ we can test whether it is P-Prime or not.

For $\omega=(2,4,7,10,11)\epsilon Q_{5,15}$,

$O(\omega)=(\sigma(2),\sigma(4),\sigma(7),\sigma(10),\sigma(11))=(2,4,2,5,1)$

Thus, ω is not P-Prime.

For $\omega=(1,5,9,12,13)\epsilon Q_{5,15}$,

$O(\omega)=(\sigma(1),\sigma(5),\sigma(9),\sigma(12),\sigma(13)) = (1,5,4,2,3)$

Thus, ω is P-Prime, $sgn(\omega)=sgn\pi(1\ 5\ 4\ 2\ 3)=-1$

■

For a given sequence $r=(r_1,r_2)=(2,5)\epsilon Q_{2,5}$ the sequence $\omega=(\omega_1,\omega_2)=(10,12)\epsilon Q_{2,15}$ is r-Prime because $O(\omega_1)=O(10)=5=r_2$ and $O(\omega_2)=O(12)=2=r_1$. On the contrary the sequence $\omega=(\omega_1,\omega_2)=(7,11)\epsilon Q_{2,15}$ is not r-Prime because $O(\omega_1)=O(7)=2=r_1$, but $O(\omega_2)=1 \neq r_1,r_2$.

The above theory is leading us to the following procedure for testing the P-Prime property of a sequence.

Algorithm SEQUEN1

For given integers $k,q\epsilon Z^+$ the following Algorithm constructs the set $P_{q,kq}$ and for a given sequence $\omega=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_k})\epsilon Q_{k,kq}$ it tests if it is P-Prime. In the sequel, it computes the weight(W) and the sign(SIGN) of the P-Prime ω.

construct $P_{q,kq}=(P_1,P_2,\ldots,P_q)$

Each $P_i := (p_{i_1},p_{i_2},\ldots,p_{i_k})$

$$p_{i_j} := (i-1)j+k, \quad j=1,2,\ldots,k$$

for $l = 1,\ldots,k$

Specify appropriate indices j,x

such as $\omega_{i_l} \epsilon P_j$, $\omega_{i_l} = p_{j_x}$

$w(\omega_{i_l}) := q-j$

$\sigma(\omega_{i_l}) := x$

for $l = 1,\ldots,k-1$

for $j = l+1,\ldots,k$

If $\sigma(\omega_l)=\sigma(\omega_{i_j})$ **then**

ω is not P-**Prime**

quit

ω is P-Prime

$$W := \sum_{l=1}^{k} w(\omega_{i_l})$$

s:=number of pairs $(\omega_{i_m},\omega_{i_n})$ for which

$\omega_{i_m} > \omega_{i_n}$ but ω_{i_m} precedes ω_{i_n} in ω

if s=even **then**

SIGN:=1

else

SIGN:=-1

Alg: 4.2

Due to the special construction of the set $P_{q,kq}$ and to the definition concerning the determination of the P-Prime property of a sequence $\omega \epsilon Q_{k,kq}$, when algorithm **SEQUEN1** is to be implemented to a computer, the following modified form can be used.

Algorithm P-PRIME

For given integers $k,q \epsilon Z^+$ and a sequence $\omega=(\omega_1,\ldots,\omega_k) \epsilon Q_{k,kq}$, the following algorithm tests if it is P-Prime. If it is, it computes its weight(W) and its sign(SIGN).

```
for  i = 1,...,k
    if  ω_i mod k=0  then
            cond := 1
    else
            cond := 0
    s_i := ω_i mod k + cond·k
P-Prime := True
for i = 1,...,k-1
    for  j = i+1,...,k
         if  s_i=s_j  then
             P-Prime := False
if  P-Prime = True
    w := 0
    for i = 1,...,k
        if  ω_i mod  k ≠ 0  then
              cond := 1
        else
              cond := 0
        w := w+q-(ω_i div k)-cond
   SIGN := 1
   for i = 1,...,k-1
       for j=i+1,...,k
           if s_i>s_j  then
              SIGN := SIGN·(-1)
```

Alg: 4.3

In section 4.2 it was mentioned that the number of sequences belonging to $Q_{k,kq}$ is $\binom{kq}{k}$. After the definition of the P-Prime sequences of a set, it

is reasonable to examine what is the number of them. The following Proposition will help us to handle the above issue.

Proposition (4.1): Given a matrix $A=(a_{ij})\epsilon R^{mxn}$ with $a_{ij}<a_{i,j+1}$, $a_{in}<a_{i+1,1}$, $i=1,2,\ldots,m$, $j=1,2,\ldots,n$, the number of the sequences $(a_{i_1j_1},a_{i_2j_2},\ldots,a_{i_nj_n})$ under the restrictions:

$$a_{i_1j_1}<a_{i_2j_2}<\ldots<a_{i_nj_n},\quad j_1\neq j_2\neq\ldots\neq j_n \tag{4.19}$$

is m^n.

Proof: We notice that the required sequences can be constructed in the following way: We take the orderings of the m rows of matrix A by n with repetition (without restriction). Each ordering is of the form $\sigma=(\sigma_1,\sigma_2,\ldots,\sigma_n)$, $\sigma_i\epsilon\{1,2,\ldots,m\}$ $i=1,2,\ldots n$ and let Σ the set all of them. It is known [Char., 1] that the number of these sequences is m^n ($N(\Sigma)=m^n$, where $N(\Sigma)$ is the cardinal number of Σ).

Let $M:R^n\longrightarrow R^n$ a function that orders in ascending order the sequences $(x_1,x_2,\ldots,x_n)$ of R^n e.g. $M(x_1,x_2,\ldots,x_n)=$ $(x_{i_1},x_{i_2},\ldots,x_{i_n})$, $x_{i_1}\leq x_{i_2}\leq\ldots\leq x_{i_n}$. To each sequence $\sigma=(\sigma_1,\sigma_2,\ldots,\sigma_n)\epsilon\Sigma$ we correspond the sequence $M(a_{\sigma_1,1},a_{\sigma_2,2},\ldots,a_{\sigma_n,n}) = (a_{\sigma_{i_1},i_1},a_{\sigma_{i_2},i_2},a_{\sigma_{i_n},i_n})$ and let Σ_M the set all of them. For reasons due to their construction, each sequence of Σ_M satisfies restrictions (4.19). It is evident now, that the sequences whose number we are asked to find, are precisely the elements of Σ_M. $N(\Sigma_M)=m^n$, so the required number is m^n.

■

Example (4.3): Let $A=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}\epsilon R^{2x3}$ be a given matrix. What is the number of the sequences $(a_{i_1j_1},a_{i_2j_3},a_{i_3j_3})$ under the restrictions:

$$a_{i_1j_1}<a_{i_2j_2}<a_{i_3j_3},\quad j_1\neq j_2\neq j_3 \tag{4.20}$$

We construct the set Σ containing the orderings of the 2 rows of A taken by 3 with repetition.

Σ = {(1,1,1),(1,1,2),(1,2,1),(2,1,1),(1,2,2),(2,1,2),(2,2,1),(2,2,2)}

Now to each element $\sigma=(\sigma_1,\sigma_2,\ldots,\sigma_n)\epsilon\Sigma$ we correspond the element $M(a_{\sigma_1,1},a_{\sigma_2,2},\ldots,a_{\sigma_n,n})$. More explicitly to

$(2,1,1)\epsilon\Sigma\longrightarrow M(a_{21},a_{12},a_{13})=M(4,2,3)=(2,3,4)$. So we produce the set:

$\Sigma_M = \{(1,2,3),(1,2,6),(1,3,5),(2,3,4),(1,5,6),(2,4,6),(3,4,5),(4,5,6)\}$

The required sequences $(a_{i_1j_1},a_{i_2j_2},a_{i_3j_3})$ satisfying restrictions (4.20) are exactly the elements of Σ_M. The number of them is $m^n=2^3=8$.

■

If we regard the matrix $A=(a_{ij})\epsilon R^{qxk}$ with $a_{11}=1$, $a_{ij}=a_{i,j-1}+1$, $a_{i+1,1}=a_{i,n}+1$, $i=1,2,\ldots,q$, $j=1,2,\ldots,k$, then according to Proposition (4.3.1), the number of the sequences $(a_{i_1j_1}, a_{i_2j_2},\ldots, a_{i_kj_k})$ satisfying :

$$a_{i_1j_1}<a_{i_2j_2}<\ldots<a_{i_kj_k},\quad j_1\neq j_2\neq\ldots\neq j_k$$

is q^k. But these sequences are exactly the P-Prime sequences of $Q_{k,kq}$. Therefore, for a given set $Q_{k,kq}$ the number of its P-Prime sequences is q^k.

Notation (4.2) and Definition (4.1) can be restated in a different way, which will give us the ability of facing the problems of section 4.5 in another manner different than the one provided from the notion of P-Prime sequences. (Both ways will be fully developed in section 4.5). The following notation and definitions are required.

Notation (4.3): For given integers $k_1,k_2,\ldots,k_m\epsilon Z^+$, $\Sigma=\sum_{i=1}^{m} k_i$ we denote by $N_\Sigma=\{1,2,3,\ldots,\Sigma\}$ the set containing all the integers starting from 1 till Σ. The set N_Σ can be partitioned into m successive subsets containing k_i elements each:

$N_i=(n_{k_{i-1}+1},\ldots,n_{k_i})$, $i=1,2,\ldots,m$, $n_j\epsilon N_\Sigma$, $j=k_{i-1},\ldots,k_i$, $n_{k_0}=0$

e.g. $N_1=\{1,2,3,\ldots,n_{k_1}\}$, $N_2=\{n_{k_1}+1,\ldots,n_{k_2}\},\ldots,$ $N_m=\{n_{k_{m-1}+1},\ldots,n_{k_m}\}$

■

Definition (4.2): For an integer $j\epsilon N_\Sigma$ there exists a $N_k\subset N_\Sigma: j\epsilon N_k$

(i) The degree of j with respect to N_k is defined by

$$d(j)\equiv k$$

(ii) The order of j with respect to N_k is defined by $\sigma(j)$ which is the order j appears in N_k minus one.

(iii) For $\omega=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_m})\epsilon Q_{m,\Sigma}$ we define as degree of ω

$$D(\omega) \equiv (d(\omega_{i_1}),d(\omega_{i_2}),\ldots,d(\omega_{i_m}))$$

(iv) ω is N-Prime if $D(\omega)$ has distinct elements.

(v) If ω is N-Prime, then we define as weight of ω

$$W(\omega) \equiv \sum_{j=1}^{m} \sigma(\omega_{i_j})$$

■

We may illustrate the above definitions with the following example.

Example (4.4): For $k_1=2$, $k_2=3\epsilon Z^+$, $m=2$, $\Sigma=\sum_{i=1}^{2} k_i=5$ the set $N_\Sigma=\{1,2,3,4,5\}$ and the partitions are $N_1=\{1,2\}$, $N_2=\{3,4,5\}$. For an integer $j=3\epsilon N_\Sigma$ ⫶ N_2:$j\epsilon N_2$, $d(3)=2$, $\sigma(3)=0$

For $\omega=(1,2)\epsilon Q_{2,5}$, $D(\omega)=(d(1),d(2))=(1,1)$, ω is not N-Prime

For $\omega=(1,5)\epsilon Q_{2,5}$, $D(\omega)=(d(1),d(5))=(1,2)$, ω is N-Prime and

$W(\omega)=\sigma(1)+\sigma(5)=2$.

■

Now, except the notion of a P-Prime sequence we have also the notion of a N-Prime sequence. For a concrete sequence of a given lexicographically ordered set, we can test if it is P-Prime or N-Prime (respectively to what kind of integers' sets we have used) and if it does, we take it into account in later calculations. This technique will be exclusively used for the evaluation of the compounds of polynomial matrices.

The above theory is leading us to the following procedure for testing the N-Prime property of a sequence.

Algorithm SEQUEN2

For given integers $k_1,k_2,\ldots,k_m\epsilon Z^+$, $\Sigma=\sum_{i=1}^{m} k_i$, the following algorithm constructs the sets N_i, $i=1,2,\ldots,m$ and for a given sequence

$\omega=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_m})\epsilon Q_{m,\Sigma}$ it tests if it is N-Prime. It also finds the weight of the N-Prime sequences.

```
for  i = 1,...,m
     construct N_i={n_{k_{i-1}+1},...,n_{k_i}}
     n_ij := n_{i-1,k_{i-1}+j},  j=1,2,...,k_i, n_{0,0}=0
for  l = 1,...,m
     Specify appropriate indices j,x
     such as ω_{i_l}εN_j, ω_{i_l}=n_{j_x}
     d(ω_{i_l}) := j
     σ(ω_{i_l}) := x
for  l = 1,...,m-1
     for j = l+1,...,m
         if  d(ω_{i_l})=d(ω_{i_j}) then
             ω is not N-Prime
             quit
ω is N-Prime
     m
W := Σ  σ(ω_{i_l})
    l=1
```

Alg: 4.4

Due to the special construction of the subsets N_i i=1,2,...,m of the set N_Σ and to the definition concerning the determination of the N-Prime property of a sequence $\omega\epsilon Q_{m,\Sigma}$, when algorithm **SEQUEN2** is to be implemented to a computer, the following modified form can be used.

Algorithm N-PRIME

For given integers $k_1,k_2,\ldots,k_m\epsilon Z^+$, $\Sigma=\sum_{i=1}^{m} k_i$ and a sequence $\omega=(\omega_1,\omega_2,\ldots,\omega_m)\epsilon Q_{m,\Sigma}$ the following algorithm tests if ω is N-Prime. If it is, it evaluates the weight(W) of the sequence.

```
for i = 1,...,m
    W := k_1
    j := 1
    while  ω_i>W
           j := j+1
           W := W+k_j
    d_i := j
    s_i := ω_i-(W-k_j)-1
N-Prime := True
for i = 1,...,m
    for j = i+1,...,m
        if d_i=d_j then
           N-Prime := False
if N-Prime := True
          m
   W :=   Σ  s_i
         i=1
```

Alg: 4.5

Finally, before closing this section, one more question can be set about the number of the N-Prime sequences of a given set $Q_{m,\Sigma}$. This can be easily answered using simple notions of Combinatorial Analysis.

Let $\omega=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_m})$ be a sequence of $Q_{m,\Sigma}$. Then, ω_{i_1} can be chosen from N_1 in k_1 ways and for each way ω_{i_2} can be chosen from N_2 in k_2 ways,..., and for each way ω_{i_m} can be chosen from N_m in k_m ways. Therefore, applying the multiplicative law [Char., 1] the elements $\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_m}$ can be chosen in $k_1,k_2,\ldots,k_m$ ways. Thus, the number of N-Prime sequences is $\prod_{i=1}^{m} k_i$.

4.4 COMPUTATION OF COMPOUNDS OF REAL MATRICES

4.4.1 The numerical algorithm

Let $A \epsilon R^{mxn}$ be a given matrix. Using specific tools of section 4.3, the following algorithm computes the compound of a real matrix.

Algorithm COMREL

For a given matrix $A \epsilon R^{mxn}$ and for an integer p, $1 \leq p \leq \min\{m,n\}$, the following algorithm computes the p-th compound matrix of A $C_p(A) \epsilon R^{\binom{m}{p} x \binom{n}{p}}$.

```
for each sequence r=(r1,r2,...,rp)εQp,m
    for each sequence c=(c1,c2,...,cp)εQp,n
        comp := a_c^r
        Print comp
```

Alg: 4.6

Implementation of the algorithm

Applying this algorithm to a computer there is no need to store in the memory all the sequences of the sets $Q_{p,m}$ and $Q_{p,n}$, something that requires two arrays of dimension $(\binom{m}{p} x p)$, $(\binom{n}{p} x p)$ each. Also there is no need to keep all the elements of the compound matrix $C_p(A)$, which would need an extra array of dimension $(\binom{m}{p} x \binom{n}{p})$. This ability is very useful especially if m and n are large. (For instance, if m=20, n=18 and we want to compute $C_2(A)$ for $A \epsilon R^{20x18}$, it is completely impractical to store in the memory all the elements of $C_2(A) \epsilon R^{190x153}$!)

Algorithm **COMREL** uses only two arrays of dimension p each, for storing temporalily the elements of $Q_{p,m}$ and $Q_{p,n}$.

These elements are computed according to algorithm **CONSEQ** of section 4.3. In order to avoid keeping in memory all the elements of the sets $Q_{p,m}$ and $Q_{p,n}$, for each sequence of $Q_{p,m}$ all the sequences of $Q_{p,n}$ are found and for each pair the corresponding element of the compound matrix is computed and printed directly without been kept in memory. For each evaluation subroutine F03AAF of NAG Library is used and O(p3) flops are required. Doing so, we save

memory but on the other hand we introduce extra evaluations. Taking into account the facts that computer memory costs a great deal and that the extra evaluations needed are only simple additions, which are done very fast without any more cost, we claim that it is preferable to save memory than simple arithmetic additions.

In order to combine both less memory and less arithmetic operations an alternative way based on the previous discussion is to keep in memory only the sequences of $Q_{p,n}$ and for each sequence of $Q_{p,m}$ do not find again all the sequences of $Q_{p,n}$ but take them directly from the memory.

Algorithm **COMREL** was programmed [Mit. & Kar., 1] and tested for several cases. Some numerical examples illustrating its application are presented in Appendix A.

4.4.2 Applications of the numerical algorithm

The evaluation of the compound of a real matrix can have several applications in many fields. Some of the most important are:

(I) Computation of exterior products

Let $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_n \epsilon R^m$ be given vectors, m<n. These vectors form the matrix $A=[\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_n]\epsilon R^{mxn}$. The n-exterior product $\underline{x}_1\wedge\underline{x}_2\wedge\ldots\wedge\underline{x}_n$ can be expressed using the notion of compound matrix. More specifically,

$$\underline{x}_1\wedge\underline{x}_2\wedge\ldots\wedge\underline{x}_n = C_m(A) = [\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_p],\ p=\binom{n}{m}$$

(II) Computation of Decentralization characteristics

In [Kar., Lai. & Gia., 1] the Decentralized Determinantal Assignment Problem (DDAP) and the Decentralization Characteristic (DC) are defined. Using compound matrices, a procedure for the evaluation of $D(H^v_{r,p})$, $D(H^v_{q,p})$ can be derived.

The formulation of this algorithm is still under consideration.

(III) Computation of an uncorrupted base

In Chapter 5 a convenient technique for selecting a best uncorrupted base for the row space of a matrix based on compounds of real matrices, is developed. This approach will be very useful in several computational problems arising from Control Theory and requiring the selection of linearly independent sets of vectors.

4.5 COMPUTATION OF COMPOUNDS OF POLYNOMIAL MATRICES

Let $M(s)\epsilon R^{mxl}[s]$, $m\geq l$ be a given polynomial matrix, q the maximum degree of polynomials. For a given integer $p\leq l$, we want to compute $C_p(M(s))$. According to the values of p, we discriminate the following cases.

4.5.1 For $M(s)\epsilon R^{mxl}[s]$, evaluation of $C_l(M(s))$

According to the way that matrix M(s) can be expressed, we develop separately two different methods of tackling this evaluation.

a) Matrix M(s) can be written in the form:

$$M(s)=A_q s^q+A_{q-1}s^{q-1}+\ldots+A_0 s^0=\sum_{i=0}^{q} A_i s^i, \text{ where } A_i\epsilon R^{mxl},\ i=0,1,\ldots,q$$

More explicitly,

$$M(s)=[A_q\ A_{q-1}\ldots A_0]\cdot\begin{bmatrix} s^q I_l \\ s^{q-1} I_l \\ \vdots \\ s^0 I_l \end{bmatrix} = A\cdot B_{q,l} \qquad (4.21)$$

where $A\epsilon R^{mxl(q+1)}$, $B_{q,l}\epsilon R^{l(q+1)xl}[s]$, $I_l\epsilon R^{lxl}$ the identity matrix.
We are interested in evaluating $C_l(M(s))$. Using expression (4.21) we have:

$$C_l(M(s))=C_l(A\cdot B_{q,l}) = C_l(A)\cdot C_l(B_{q,l}) \qquad (4.22)$$

$C_l(A)$ can be easily computed using algorithm **COMREL** of section 4.4, since A is a real matrix. Let us discuss the evaluation of $C_l(B_{q,l})$. Matrix $B_{q,l}$ can be expressed analytically in the following form:

$$B_{q,l} = \left[\begin{array}{cccc|l} s^q & 0 & 0 \ldots & 0 & 1 \text{ row} \\ 0 & s^q & 0 \ldots & 0 & 2^{nd} \text{ row} \\ \cdot & \cdot & \cdot \ldots & \cdot & \cdot \\ 0 & 0 & 0 \ldots & s^q & l \text{ row} \\ \hline s^{q-1} & 0 & 0 \ldots & 0 & l+1 \text{ row} \\ 0 & s^{q-1} & 0 \ldots & 0 & \cdot \\ \cdot & \cdot & \cdot \ldots & \cdot & \cdot \\ 0 & 0 & 0 \ldots & s^{q-1} & 2l \text{ row } \epsilon R^{l(q+1)xl}[s] \\ \hline \cdot & \cdot & \cdot \ldots & \cdot & \\ \hline 1 & 0 & 0 \ldots & 0 & l_{q+1} \text{ row} \\ 0 & 1 & 0 \ldots & 0 & \cdot \\ \cdot & \cdot & \cdot \ldots & \cdot & \cdot \\ 0 & 0 & 0 \ldots & 1 & l(q+1) \text{row} \end{array}\right] \qquad (4.23)$$

We remark that $B_{q,l}$ has a special structure with many zero entries. Evidently, many entries of the column matrix $C_l(B_{q,l}) \epsilon R^{\binom{l(q+1)}{l} x l}[s]$ will be zeros. The point is, that we must try to specify the nonzero elements of $C_l(B_{q,l})$ and attempt to evaluate only these entries, ignoring the other zero ones. Thus, the following question arises. Is there any way to assess exactly the nonzero entries of $C_l(B_{q,l})$ and write them directly by using some formula?

Next, a Proposition based on notions defined in section 4.3 and answering the above question is developed.

Proposition (4.2): Let $B_{q,l} \epsilon R^{l(q+1)xl}[s]$ be a polynomial matrix of the form (4.23). Let $C_l(B_{q,l}) = [a_{\omega_1}(s), \ldots, a_{\omega_\sigma}(s)]^t \epsilon R^{\sigma x 1}$,

$\sigma = \binom{l(q+1)}{l}$, $\omega_i \epsilon Q_{l,l(q+1)}$, $i=1,2,\ldots,\sigma$, be the l-th compound matrix of $B_{q,l}$.

(i) For some $i=1,2,\ldots,\sigma$, $a_{\omega_i}(s) \not\equiv 0$ iff $\omega_i \epsilon Q_{l,l(q+1)}$ is P-Prime.

(ii) The nonzero entries $a_{\omega_i}(s)$ are given by:

$$a_{\omega_i}(s) = \text{sgn}(\omega_i) s^{W(\omega_i)} \qquad (4.24)$$

Proof: It is defined, that each element $a_{\omega_i}(s)$ of $C_l(B_{q,l})$ is equal to $\det(B_{q,l}[\omega_i/\ ])$, $\omega_i \epsilon Q_{l,l(q+1)}$. Thus, in order to compute the elements of the compound, we must choose sequences of l rows from the given set of rows $\{1,2,\ldots,l(q+1)\}$. This set can be partitioned in the subsets: $P_1=\{1,2,\ldots,l\}$, $P_2=\{l+1,\ldots,2l\},\ldots,P_{q+1}=\{lq+1,\ldots,l(q+1)\}$. From the special structure that matrix $B_{q,l}$ has, we remark that the only sequences $r_j=(r_{j_1},\ldots,r_{j_l})$, $j\epsilon\{0,1,\ldots,\sigma\}$, of matrix rows that give a nonzero lxl determinant are those satisfying $\sigma(r_{j_1})\neq\sigma(r_{j_2})\neq\ldots\neq\sigma(r_{j_l})$ with respect to $P_1,P_2,\ldots,P_{q+1}$ partitions.

(i) If $\omega_i = (\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_l})\epsilon Q_{l,l(q+1)}$ is P-Prime, then $\sigma(\omega_{i_1})\neq\sigma(\omega_{i_2})\neq\ldots\neq\sigma(\omega_{i_l})$ and thus $\det(B_{q,l}[\omega_i/\ ])=a_{\omega_i}(s)\neq 0$. On the contrary, if $\det(B_{q,l}[\omega_i/\ ])=a_{\omega_i}(s)\neq 0$, for some $\omega_i=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_l})\epsilon Q_{l,l(q+1)}$, then $\sigma(\omega_{i_1})\neq\ldots\neq\sigma(\omega_{i_l})$ with respect to $P_1,P_2,\ldots,P_{q+1}$ partitions, which means that ω_i is P-Prime.

(ii) Each row of the elements $\det(B_{q,l}[\omega_i/\ ])$, $\omega_i\epsilon Q_{l,l(q+1)}$ P-Prime, has only one nonzero element of the form s^j, for some $j=0,1,\ldots,q$ and all the rest elements are zeroes. Also all these nonzero elements appear in a different position in each row.

Thus, for a given $\omega_i=(\omega_{i_1},\ldots,\omega_{i_l})\epsilon Q_{l,l(q+1)}$, P-Prime the element $\det(B_{q,l}[\omega_i/\ ])$ will be of a form:

$$\det\left\{\begin{array}{l}\omega_{i_1}\\ \omega_{i_2}\\ \vdots\\ \omega_{i_l}\end{array}\left[\begin{array}{ccccccc} s^{j_1} & 0 & \ldots & \ldots & . & 0\\ 0 & 0 & .. & s^{j_2} & . & 0\\ . & . & . & . & . & . & .\\ 0 & s^{j_l} & . & . & . & 0\end{array}\right]\right\}, j_i\epsilon\{0,1,\ldots,q\},\ i=1,2,\ldots,l \qquad (4.25)$$

We remark that each ω_{i_j} belongs to some $P_1,P_2,\ldots,P_{q+1}$, let us say to P_m. The

weight of ω_{i_j} is equal to $w(\omega_{i_j})=w(P_m)=q+1-m$. If we expand the determinant (4.25) we remark that is equal to

$$sgn(\omega_i)\cdot s^{w(\omega_{i_1})+\ldots+w(\omega_{i_l})} = sgn(\omega_i)\cdot s^{W(\omega_i)}$$

■

Combining (4.21), (4.22) and the preceding Proposition we conclude the following result.

Proposition (4.3): Let $M(s)\epsilon R^{mxl}[s]$, $m\geq l$, q the maximum degree of the polynomials, be a given polynomial matrix that can be expressed in the form (4.21). Then, the rows $r_i(s)$, $i=1,2,\ldots,\binom{m}{l}$ of $C_l(M(s))\epsilon R^{\binom{m}{l}xl}[s]$ are given by:

$$r_i(s) = \sum_{\omega} sgn(\omega)\cdot\det(A[a_i/\omega])s^{W(\omega)} \qquad (4.26)$$

where $a_i=(a_{i_1},a_{i_2},\ldots,a_{i_l})\epsilon Q_{l,m}$, $\omega\epsilon Q_{l,l(q+1)}$ P-Prime, $A\epsilon R^{mxl(q+1)}$ the matrix of (4.21).

■

Proposition (4.3) gives us a convenient formula for the computation of the rows of $C_l(M(s))$. The main advantage of (4.26) is that it does not use all the $\binom{l(q+1)}{l}$ sequences of $Q_{l,l(q+1)}$ but only the P-Prime ones, that are $(q+1)^l$. Formula (4.26) can be used for the construction of an algorithm for the evaluation of $C_l(M(s))$. Next, we present such an algorithm.

Algorithm COMPOL1

Let $M(s)\epsilon R^{mxl}[s]$, $m\geq l$, $p=l$, q the maximum degree of the polynomials and $A\epsilon R^{mxl(q+1)}$ the matrix of (4.21). The following algorithm evaluates $C_p(M(s))=(comp)_{ij}\epsilon R^{\binom{m}{l}xl}[s]$.

```
for each sequence α=(a_{i_1},a_{i_2},...,a_{i_l})εQ_{l,m}

    for each sequence ω=(ω_{i_1},ω_{i_2},...,ω_{i_l})εQ_{l,l(q+1)}

        if  ω  is P-Prime then

                    l
            W :=   Σ   w(ω_{i_j})
                  i=1

            sgn := sgn(ω)

            comp := sgn·a_α^ω·s^W

            Print comp
```

Alg: 4.7

Implementation of the algorithm

As in algorithm **COMREL**, in order to avoid keeping in memory all the sequences of $Q_{l,m}$ and $Q_{l,l(q+1)}$, we perform a bit more computations and for each sequence of $Q_{l,m}$ all the sequences of $Q_{l,l(q+1)}$ are found and for those that are P-Prime the corresponding element of the compound is computed. This element is directly printed so it does not have to be stored in memory. The P-Prime elements of $Q_{l,l(q+1)}$ will be found using algorithm **SEQUEN1**.

An alternative way for the previous algorighm which combines both economy in memory and in arithmetic operations, is to keep in memory only the P-Prime sequences of $Q_{l,l(q+1)}$. Then, for each sequence of $Q_{l,m}$ and for all those P-Prime elements, we evaluate the elements of the compound matrix. This way will be used later in algorithm **COMPOL2.**

Example (4.9):

$$\text{Let} \quad M(s)=\begin{bmatrix} 3s+1 & 2s \\ 5s+2 & s+4 \\ 3s+1 & s+2 \end{bmatrix} \epsilon R^{3x2}[s], \quad q=1$$

We want to evaluate $C_2(M(s))=(comp)_{ij}\epsilon R^{\binom{3}{2}x1}[s]$

$$M(s) = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 5 & 1 & 2 & 4 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} s & 0 \\ 0 & s \\ \hline 1 & 0 \\ 0 & 1 \end{bmatrix} = A \cdot B_{1,2}$$

Applying algorithms **COMPOL1** and **SEQUEN1** for k=1=2, q := q+1=2 we have:

$P_{2,4}=\{P_1,P_2\}$,$P_1=\{1,2\}$, $P_2=\{3,4\}$

For $\alpha=(1,2)\epsilon Q_{2,3}$

For $\omega=(1,2)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=2$, $sgn(\omega)=1$

comp := $a_{12}^{12} \cdot s^2 = -7s^2$

For $\omega=(1,3)\epsilon Q_{2,4}$, $O(\omega)=(1,1)$ not P-Prime

For $\omega=(1,4)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=1$, $sgn(\omega)=1$

comp := $a_{13}^{14} \cdot s = 12s$

For $\omega=(2,3)\epsilon Q_{2,4}$, $O(\omega)=(2,1)$ P-Prime, $W(\omega)=1$, $sgn(\omega)=-1$

comp := $-a_{13}^{23} \cdot s = -3s$

For $\omega=(2,4)\epsilon Q_{2,4}$, $O(\omega)=(2,2)$ not P-Prime

For $\omega=(3,4)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=0$, $sgn(\omega)=1$

comp := $a_{13}^{23} \cdot s^0 = 4$

$comp_{11} := -7s^2+12s-3s+4$

For $\alpha=(1,3)\epsilon Q_{2,3}$

For $\omega=(1,2)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=2$, $sgn(\omega)=1$

comp := $a_{13}^{12} \cdot s^2 = -3s^2$

For $\omega=(1,3)\epsilon Q_{2,4}$, $O(\omega)=(1,1)$ not P-Prime

For $\omega=(1,4)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=1$, $sgn(\omega)=1$

comp := $a_{13}^{14} \cdot s = 6s$

For $\omega=(2,3)\epsilon Q_{2,4}$, $O(\omega)=(2,1)$ P-Prime, $W(\omega)=1$, $sgn(\omega)=-1$

comp := $-a_{13}^{23} \cdot s = -s$

For $\omega=(2,4)\epsilon Q_{2,4}$, $O(\omega)=(2,2)$ not P-Prime

For $\omega=(3,4)\epsilon Q_{2,4}$ $O(\omega)=(1,2)$ P-Prime, $W(\omega)=0$, $sgn(\omega)=1$

comp := $a_{13}^{34} \cdot s = 2$

$comp_{21} := -3s^2+6s-s+2$

For $\alpha=(2,3)\epsilon Q_{2,3}$

For $\omega=(1,2)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=2$, $sgn(\omega)=1$

comp := $a_{23}^{12}\cdot s^2 = 2s^2$

For $\omega=(1,3)\epsilon Q_{2,4}$, $O(\omega)=(1,1)$ not P-Prime

For $\omega=(1,4)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=1$, $sgn(\omega)=1$

comp := $a_{23}^{14}\cdot s = -2s$

For $\omega=(2,3)\epsilon Q_{2,4}$, $O(\omega)=(2,1)$ P-Prime, $W(\omega)=1$, $sgn(\omega)=-1$

comp := $-a_{23}^{23}\cdot s = s$

For $\omega=(2,4)\epsilon Q_{2,4}$, $O(\omega)=(2,2)$ not P-Prime

For $\omega=(3,4)\epsilon Q_{2,4}$, $O(\omega)=(1,2)$ P-Prime, $W(\omega)=0$, $sgn(\omega)=1$

comp := $a_{23}^{34}\cdot s^0 = 0$

$comp_{31} = 2s^2-2s+s+0$

Finally $$C_2(M(s)) = \begin{bmatrix} -7s^2+12s-3s+4 \\ -3s^2+6s-s+2 \\ 2s^2-2s+s+0 \end{bmatrix} = \begin{bmatrix} -7s^2+9s+4 \\ -3s^2+5s+2 \\ 2s^2-s \end{bmatrix} \epsilon R^{3x1}[s]$$

■

Algorithm **COMPOL1** was programmed [Mit. & Kar., 1] and tested for several cases. Some numerical results illustrating its application are presented in Appendix A.

b) Matrix M(s) can be written in the form:
$M(s)=[\underline{m}_1(s),\underline{m}_2(s),\ldots,\underline{m}_l(s)]\epsilon R^{mxl}[s]$ and let d_i, $i=1,2,\ldots,l$ be the maximum degree of each column. Each column $\underline{m}_i(s)$, $i=1,2,\ldots,l$ is a polynomial vector which can be expressed in the form

$\underline{m}_i(s)=M_i\cdot\underline{e}_{d_i}(s)$, where $M_i\epsilon R^{mx(d_i+1)}$ and $\underline{e}_{d_i}(s)=[1,s,s^2,\ldots,s^{d_i}]^t$.

Therefore, matrix M(s) can be analysed as follows:

$$M(s) = [M_1\ M_2\ \ldots\ M_l]\cdot \begin{bmatrix} \underline{e}_{d_1}(s) & & 0 \\ & \underline{e}_{d_2}(s) & \\ & & \ddots \\ 0 & & \underline{e}_{d_l}(s) \end{bmatrix} = M\cdot B_{d,l} \qquad (4.27)$$

where $M \epsilon R^{mxd}$, $B_{d,1} \epsilon R^{dx1}$, $d = \sum_{i=1}^{1} (d_i+1)$.

We are interested in evaluating $C_1(M(s))$. Using expression (4.27) we have:

$$C_1(M(s)) = C_1(M \cdot B_{d,1}) = C_1(M) \cdot C_1(B_{d,1}) \tag{4.28}$$

$C_1(M)$ can be easily computed using algorithm **COMREL** since M is a real matrix. Let us discuss the evaluation $C_1(B_{d,1})$. Matrix $B_{d,1}$ can be expressed analytically in the form:

$$B_{d,1} = \begin{bmatrix} 1 & & \\ s & & \\ s^2 & & 0 \\ . & & \\ . & & \\ . & & \\ s^{d_1} & & \\ \hline & 1 & \\ & s & \\ & . & \\ 0 & . & 0 \\ & . & \\ & s^{d_2} & \\ \hline & & \\ \hline & & 1 \\ & & s \\ & & . \\ 0 & & . \\ & & . \\ & & s^{d_1} \end{bmatrix} \epsilon R^{dx1}[s] \tag{4.29}$$

Matrix $B_{d,1}$ has a similar structure with matrix $B_{q,1}$ that was developed before. Thus, for the evaluation of $C_1(B_{d,1})$ the same question as in the evaluation of $C_1(B_{q,1})$ arises: Is there any way to assess exactly the nonzero entries of $C_1(B_{d,1})$ and write them directly by using some formula?

Next, a Proposition based on notions defined in section 4.3 and answering the above question is developed.

Proposition (4.4): Let $B_{d,l} \epsilon R^{dxl}[s]$ be a polynomial matrix of the form (4.9). Let $C_l(B_{d,l})=[a_{\omega_1}(s),\ldots,a_{\omega_\sigma}(s)]^t \epsilon R^{\sigma x l}$, $\sigma=\binom{d}{l}$, $\omega_i \epsilon Q_{l,d}$, $i=1,2,\ldots,\sigma$, be the l-th compound matrix of $B_{d,l}$.

(i) For some $i=1,2,\ldots,\sigma$, $a_{\omega_i}(s) \neq 0$ iff $\omega_i \epsilon Q_{l,d}$ is N-Prime

(ii) The nonzero entries $a_{\omega_i}(s)$ are given by:

$$a_{\omega_i}(s) = s^{w(\omega_i)} \tag{4.30}$$

Proof: It is defined, that each element $a_{\omega_i}(s)$ of $C_l(B_{d,l})$ is equal to $\det(B_{d,l}[\omega_i/\ ])$, $\omega_i \epsilon Q_{l,d}$. Thus, in order to compute the elements of the compound, we must choose sequences of l rows from the given set of rows $\{1,2,\ldots,d\}$. This set can be partitioned in the subsets: $N_i=\{\Sigma_i=\sum_{j=1}^{i-1} d_j+i,\ldots,d_i+\Sigma_i\}$ $i=1,2,\ldots,l$. From the special structure that matrix $B_{d,l}$ has, we remark that the only sequences $r_j=(r_{j_1},\ldots,r_{j_l})$, $j\epsilon\{0,1,\ldots,\sigma\}$, of matrix rows that give a nonzero lxl determinant are those satisfying $d(r_{j_1}) \neq \ldots \neq d(r_{j_l})$ with respect to N_i partitions.

(i) If $\omega_i=(\omega_{i_1},\omega_{i_2},\ldots,\omega_{i_l}) \epsilon Q_{l,d}$ is N-Prime, then $d(\omega_{i_1}) \neq d(\omega_{i_2}) \neq \ldots \neq d(\omega_{i_l})$ and consequently $\det(B_{d,l}[\omega_i/\ ])=a_{\omega_i}(s) \neq 0$.

On the contrary, if $\det(B_{d,l}[\omega_i/\ ])=a_{\omega_i}(s) \neq 0$, for some $\omega_i=(\omega_{i_1},\omega_{i_2},..,\omega_{i_l}) \epsilon Q_{l,d}$, then $d(\omega_{i_1}) \neq d(\omega_{i_2}) \neq \ldots \neq d(\omega_{i_l})$ with respect to N_i partitions, which means that ω_i is N-Prime.

(ii) Each row of the elements $\det(B_{d,l}[\omega_i/\ ])$, $\omega_i \epsilon Q_{l,d}$ N-Prime, has only one nonzero element of the form s^j, for some $j=0,1,\ldots,d_i$, $i=1,2,\ldots,l$ and all the rest elements are zeros.
Thus, for a given $\omega_i=(\omega_{i_1},\ldots,\omega_{i_l}) \epsilon Q_{l,d}$, N-Prime the element $\det(B_{d,l}[\omega_i/\ ])$ will be of a form:

$$\det\left\{\begin{bmatrix}\omega_{i_1}\\ \omega_{i_2}\\ .\\ .\\ .\\ \omega_{i_1}\end{bmatrix}\begin{bmatrix}s^{j_1} & & & \\ & s^{j_2} & & 0\\ & & . & \\ & 0 & & . \\ & & & s^{j_1}\end{bmatrix}\right\}, j_k\epsilon\{0,1,\ldots,d_i\},\ i,k=1,2,\ldots,l \tag{4.31}$$

We remark that each ω_{ij} belongs to some N_i, let us say to N_m. If we expand the determinant (4.31) we observe that is equal to: $s^{w(\omega_i)}$.

■

Combining (4.27), (4.28) and the preceding Proposition we conclude the following result.

Proposition (4.5): Let $M(s)=[\underline{m}_1(s),\ldots,\underline{m}_l(s)]\epsilon R^{mxl}[s]$, $m\geq l$, be a polynomial matrix. Let d_i, $i=1,2,\ldots,l$ be the maximum degree of each column and $d=\sum_{i=1}^{l}(d_i+1)$. Matrix $M(s)$ can be expressed in the form (4.27). Then, the rows $r_i(s)$, $i=1,2,\ldots,\binom{m}{l}$ of $C_l(M(s))\epsilon R^{\binom{m}{l}xl}[s]$ are given by:

$$r_i(s) = \sum_{\omega} \det(M[a_i/\omega])s^{W(\omega)} \tag{4.32}$$

where $a_i=(a_{i_1},a_{i_2},\ldots,a_{i_l})\epsilon Q_{l,m}$, $\omega\epsilon Q_{l,d}$, N-Prime, $M\epsilon R^{mxd}$ the matrix of (4.27)

■

Next we present an algorithm for the computation of polynomial compound matrices based on Proposition (4.5).

Algorithm COMPOL2

Let $M(s)=[\underline{m}_1(s),\underline{m}_2(s),\ldots,\underline{m}_l(s)]\epsilon R^{mxl}[s]$, $m\geq l$, $p=\min(m,l)$, $d=\sum_{i=1}^{l}(d_i+1)$, d_i the maximum degree of each column $\underline{m}_i(s)$, and $M\epsilon R^{mxd}$ the matrix of (4.27).

The following algorithm evaluates $C_p(M(s))=(comp)_{ij}\epsilon R^{\binom{m}{l}x1}[s]$.

```
Find the N-Prime elements of Q_{l,d}
for each sequence a=(a_{i_1},a_{i_2},...,a_{i_l})εQ_{l,m}
    for each N-Prime sequence ω=(ω_{i_1},...,ω_{i_l}) of Q_{l,d}
        comp := m_a^ω · s^{w(ω)}
        Print comp
```

Alg: 4.8

Example (4.6):

Let $M(s) = \begin{bmatrix} 3s+1 & 2s^2 \\ 5s+2 & 2s+3 \\ 3s+2 & 5s^2+1 \end{bmatrix} \epsilon R^{3x2}[s]$, $d_1=1$, $d_2=2$, $d=5$

We want to evaluate $C_2(M(s))=(comp)_{ij}\epsilon R^{\binom{3}{2}x1}[s]$

$$M(s) = \begin{bmatrix} 1 & 3 & 0 & 0 & 2 \\ 2 & 5 & 3 & 2 & 0 \\ 2 & 3 & 1 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ s & 0 \\ \hline 0 & 1 \\ 0 & s \\ 0 & s^2 \end{bmatrix} = M \cdot B_{5,2}$$

Applying algorithms **COMPOL2** and **SEQUENC2** for m=l=2, Σ := d=5 we have:

$N_d=\{1,2,3,4,5\}$, $N_1=\{1,2\}$, $N_2=\{3,4,5\}$

For $\omega=(1,2)\epsilon Q_{2,5}$, $O(\omega)=(1,1)$ not N-Prime

For $\omega=(1,3)\epsilon Q_{2,5}$, $O(\omega)=(1,2)$ N-Prime, $W(\omega)=0$

For $\omega=(1,4)\epsilon Q_{2,5}$, $O(\omega)=(1,2)$ N-Prime, $W(\omega)=1$

For $\omega=(1,5)\epsilon Q_{2,5}$, $O(\omega)=(1,2)$ N-Prime, $W(\omega)=2$
For $\omega=(2,3)\epsilon Q_{2,5}$, $O(\omega)=(1,2)$ N-Prime, $W(\omega)=1$
For $\omega=(2,4)\epsilon Q_{2,5}$, $O(\omega)=(1,2)$ N-Prime, $W(\omega)=2$
For $\omega=(2,5)\epsilon Q_{2,5}$, $O(\omega)=(1,2)$ N-Prime, $W(\omega)=3$
For $\omega=(3,4)\epsilon Q_{2,5}$, $O(\omega)=(2,2)$ not N-Prime
For $\omega=(3,5)\epsilon Q_{2,5}$, $O(\omega)=(2,2)$ not N-Prime
For $\omega=(4,5)\epsilon Q_{2,5}$, $O(\omega)=(2,2)$ not N-Prime

For $a=(1,2)\epsilon Q_{2,3}$

For $\omega=(1,3)\epsilon Q_{2,5}$, comp := $m_{12}^{13}\cdot s^0=3$
For $\omega=(1,4)\epsilon Q_{2,5}$, comp := $m_{12}^{14}\cdot s=2s$
For $\omega=(1,5)\epsilon Q_{2,5}$, comp := $m_{12}^{15}\cdot s^2=-4s^2$
For $\omega=(2,3)\epsilon Q_{2,5}$, comp := $m_{12}^{23}\cdot s=9s$
For $\omega=(2,4)\epsilon Q_{2,5}$, comp := $m_{12}^{24}\cdot s^2=6s^2$
For $\omega=(2,5)\epsilon Q_{2,5}$, comp := $m_{12}^{25}\cdot s^3=-10s^3$

$comp_{11} := 3+2s-4s^2+9s+6s^2-10s^3$

For $a=(1,3)\epsilon Q_{2,3}$

For $\omega=(1,3)\epsilon Q_{2,5}$, comp := $m_{13}^{13}\cdot s^0=1$
For $\omega=(1,4)\epsilon Q_{2,5}$, comp := $m_{13}^{14}\cdot s=0$
For $\omega=(1,5)\epsilon Q_{2,5}$, comp := $m_{13}^{15}\cdot s^2=s^2$
For $\omega=(2,3)\epsilon Q_{2,5}$, comp := $m_{13}^{23}\cdot s=3s$
For $\omega=(2,4)\epsilon Q_{2,5}$, comp := $m_{13}^{24}\cdot s^2=0$
For $\omega=(2,5)\epsilon Q_{2,5}$, comp := $m_{13}^{25}\cdot s^3=9s^3$

$comp_{21} := 1+s^2+3s+0s^3$

For $a=(2,3)\epsilon Q_{2,3}$

For $\omega=(1,3)\epsilon Q_{2,5}$, comp := $m_{23}^{13}\cdot s^0=-4$
For $\omega=(1,4)\epsilon Q_{2,5}$, comp := $m_{23}^{14}\cdot s=-4s$
For $\omega=(1,5)\epsilon Q_{2,5}$, comp := $m_{23}^{15}\cdot s^2=10s^2$
For $\omega=(2,3)\epsilon Q_{2,5}$, comp := $m_{23}^{24}\cdot s=-4s$
For $\omega=(2,4)\epsilon Q_{2,5}$, comp := $m_{23}^{24}\cdot s^2=-6s^2$
For $\omega=(2,5)\epsilon Q_{2,5}$, comp := $m_{23}^{25}\cdot s^3=25s^3$

$comp_{31} := -4-4s+10s^2-4s-6s^2+25s^3$

$$\text{Finally } C_2(M(s)) = \begin{bmatrix} 3+11s+2s^2-10s^3 \\ 1+s^2+3s+9s^3 \\ -4-8s+4s^2+25s^3 \end{bmatrix} \in R^{3x1}[s]$$

■

Remark (4.2): Comparing the two different ways for the evaluation of the compound of a polynomial matrix $M(s)\in R^{mxl}[s]$ mentioned previously, we remark that the main difference between them is in the dimension of the matrices arising from the analysis of M(s) into a product of a real and a polynomial matrix. Analysis of (4.21) ends up into matrices $A\in R^{mxl(q+1)}$, $B_{q,l}\in R^{l(q+1)xl}[s]$, q is the maximum degree of polynomials. Analysis of (4.27) ends up into matrices $M\in R^{mxd}$, $B_{d,l}\in R^{dxl}$, $d=\sum_{i=1}^{l}(d_i+1)$, d_i, $i=1,2,\ldots,l$ the maximum degree of each column.

Generally, $d\leq l(q+1)$ thus the matrices taken using the second way are of lower dimension comparing them with those taken using the first way. It is evident that for matrices $M(s)\in R^{mxl}[s]$ with $d_i=d_j$, $i,j=1,2,\ldots,l$, $i\neq j$ the dimensions are exactly the same in both ways. In case that there is a considerable difference between the di's i=1,2,...,l of M(s), the application of the second way for the evaluation of its compound matrix is strongly recommended.

■

4.5.2 For $M(s)\in R^{mxl}[s]$, evaluation of $C_p(M(s))$, $1 \leq p < l$

Based on formulas (4.21), (4.22), (4.23) and using tools from section 4.3 we can derive the following Proposition.

Proposition (4.6): Let $B_{q,l}\in R^{l(q+1)xl}[s]$ be a polynomial matrix of the form (4.23). For a given integer p: $1\leq p<l$ let

$C_p(B_{q,l})=[\underline{m}_{\omega_1}(s),\ldots,\underline{m}_{\omega_\sigma}(s)]\in R^{\binom{l(q+1)}{p}x\sigma}$, $\sigma=\binom{l}{p}$, $\omega_i\in Q_{p,l}$ be its p-th compound matrix. Each column $\underline{m}_{\omega_i}(s)$ of $C_p(B_{q,l})$ is of the form:

$$\underline{m}_{\omega_i}(s)=[m^{r_1}_{\omega_i}(s),\ m^{r_2}_{\omega_i}(s),\ldots,m^{r_k}_{\omega_i}(s)]^t\in R^k,\quad i=1,2,\ldots,\sigma,\quad r_j\in Q_{p,l(q+1)},$$

$j=1,2,\ldots,k$, $k=\binom{l(q+1)}{p}$.

(i) For some $i=1,2,\ldots,\sigma$, $j=1,2,\ldots,k$, $m^{r_j}_{\omega_i}(s)\neq 0$ iff $r_j\in Q_{p,l(q+1)}$ is P-Prime and ω_i-Prime.

(ii) The nonzero entries $m^{r_i}_{\omega_i}(s)$ are given by:

$$m^{r_j}_{\omega_i}(s) = \mathrm{sgn}(r_j)\cdot s^{W(r_j)} \tag{4.33}$$

Proof: It is defined that each element of column $\underline{m}_{\omega_i}(s)$ of $C_p(B_{q,l})$ is equal to $\det(B_{q,l}[r_j/\omega_i])$, $\omega_i\in Q_{p,l}$, $r_j\in Q_{p,l(q+1)}$. Thus, in order to compute the elements of each column, for a given sequence $\omega_i\in Q_{p,l}$ corresponding to p columns of $B_{q,l}$, we must choose sequences of p rows from the given set of rows of $B_{q,l}$ and evaluate the relevant determinant. The set of rows $\{1,2,\ldots,l(q+1)\}$ of $B_{q,l}$ can be partitioned in the subsets: $P_1=\{1,2,\ldots,l\}$, $P_2=\{l+1,\ldots,2l\},\ldots,$ $P_{q+1}=\{lq+1,\ldots,l(q+1)\}$. From the special structure that matrix $B_{q,l}$ has, we remark that for a given sequence of columns $\omega_i=(\omega_{i_1},\ldots,\omega_{i_p})\in Q_{p,l}$ the only sequences of rows $r_j=(r_{j_1},\ldots,r_{j_r})\in Q_{p,l(q+1)}$ that have $\det(B_{q,l}[r_j/\omega_i])\neq 0$ are those with the property: $\sigma(r_{j_1})\neq\ldots\neq\sigma(r_{j_r})$ and for all $r_{j_z}\exists\ \omega_{i_m}$: $\sigma(r_{j_z})=\omega_{i_m}$ with respect to $P_1,P_2,\ldots,P_{q+1}$ partitions.

(i) If $\omega_i=(\omega_{i_1},\ldots,\omega_{i_p})\in Q_{p,l}$ and $r_j=(r_{j_1},\ldots,r_{j_p})\in Q_{p,l(q+1)}$ is P-Prime and ω_i-Prime then $\sigma(r_{j_1})\neq\ldots\neq\sigma(r_{j_p})$ and $\forall\ r_{j_z}\exists\ \omega_{i_m}$: $\sigma(r_{j_z})=\omega_{i_m}$ with respect to $P_1,P_2,\ldots,P_{q+1}$ partitions and consequently $\det(B_{q,l}[r_j/\omega_i])=m^{r_i}_{\omega_i}(s)\neq 0$. On the contrary, if $\det(B_{q,l}[r_j/\omega_i])=m^{r_i}_{\omega_i}(s)\neq 0$, for some $\omega_i=(\omega_{i_1},\ldots,\omega_{i_p})\in Q_{p,l}$, $r_j=(r_{j_1},\ldots,r_{j_p})\in Q_{p,l(q+1)}$, then $\sigma(r_{j_1})\neq\ldots,\neq\sigma(r_{j_p})$ and $\forall r_{j_z}\exists\ \omega_{i_m}:\sigma(r_{j_z})=\omega_{i_m}$, which means that r_j is P-Prime and ω_i-Prime.

(ii) Each row of the elements $\det(B_{q,l}[r_j/\omega_i])$, $\omega_i\in Q_{p,l}$, $r_j\in Q_{p,l(q+1)}$, r_j P-Prime and ω_i-Prime, has only one nonzero element of the form s^k, for some

$k=0,1,\ldots,q$ and all the rest elements are zeroes. Also all these nonzero elements appear in a different position in each row. Thus, each $m_{\omega_i}^{r_j}(s)$ will be of the form:

$$\det \begin{Bmatrix} & \omega_{i_1} & \omega_{i_2} & \cdots & \omega_{i_p} \\ r_{j_1} & s^{k_1} & 0 & \cdots & 0 \\ r_{j_2} & 0 & 0 & s^{k_2} \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_{j_p} & 0 & s^{k_p} & \cdots & 0 \end{Bmatrix}, k_i \in \{0,1,\ldots,q\},\ i=1,2,\ldots,p \tag{4.34}$$

We remark that each r_{j_i} belongs to some $P_1,\ldots,P_{q+1}$, let us say to P_m. The weight of r_{j_i} is equal to $w(r_{j_i})=w(P_m)=q+1-m$. If we expand the determinant (4.34) it is equal to

$$\operatorname{sgn}(r_j)\cdot s^{w(r_{i_1})+\ldots+w(r_{j_p})} = \operatorname{sgn}(r_j)\cdot s^{W(r_j)}$$

■

Combining (4.21), (4.22) and the above Proposition the next result readily follows:

Proposition (4.7): Let $M(s)\in R^{m\times l}[s]$, $m\geq l$, q the maximum degree of the polynomials, be a given polynomial matrix that can be expressed in the form (4.5.1). For $1\leq p\leq l$ let $C_p(M(s))=$

$=[\underline{m}_{\omega_1}(s),\ldots,\underline{m}_{\omega_\sigma}(s)]\in R^{\binom{m}{p}\times\sigma}$, $\sigma=\binom{l}{p}$ be its p-th compound matrix. Then each element of columns $\underline{m}_{\omega_i}(s)=[m_{\omega_{i,1}}(s),\ldots,m_{\omega_{i,k}}(s)]^t\in R^k$ $k=\binom{m}{p}$, is of the form:

$$m_{\omega_{i,j}}(s) = \sum_r \operatorname{sgn}(r)\cdot\det(A[a_j|r])s^{W(r)},\quad j=1,2,\ldots,k \tag{4.35}$$

where $\omega_i \epsilon Q_{p,1}$, $a_j=(a_{j_1},\ldots,a_{j_p}) \epsilon Q_{p,m}$, $r \epsilon Q_{p,1(q+1)}$, P-Prime and ω_i-Prime, $A \epsilon R^{mx1(q+1)}$ the matrix of (4.21).

■

Combining Proposition (4.1) and Remark (4.1) we conclude that for a given sequence $\omega_i \epsilon Q_{p,1}$ the number of P-Prime sequences of $Q_{p,1(q+1)}$ that are ω_i-Prime too, is $(q+1)^p$. Therefore, Proposition (4.7) implies a remarkable reduction in the number of opertions required for the evaluation of $C_p(M(s))$. More explicitly, instead of taking for each of the possible $\binom{1}{p}$ column combinations all the possible $\binom{1(q+1)}{p}$ row combinations, we actually use only the $(q+1)^p$ of them.

Example (4.7): Let

$$M(s) = \begin{bmatrix} s & s & s-1 \\ s^2+s & s^2+2s & s^2-1 \\ 2s^2-2s & s^2-2s & 2s^2-3s+2 \end{bmatrix} \epsilon R^{3x3}[s] \text{ be a polynomial matrix.}$$

We want to calculate $C_2(M(s))=[\underline{m}_{\omega_1}(s), \underline{m}_{\omega_2}(s), \underline{m}_{\omega_3}(s)]$, $\omega_i \epsilon Q_{2,3}$, $i=1,2,\ldots,\binom{3}{2}$

Matrix M(s) can be analysed as:

$$M(s)= \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & -1 \\ 2 & 1 & 2 & -2 & -2 & -3 & 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} s^2 & 0 & 0 \\ 0 & s^2 & 0 \\ 0 & 0 & s^2 \\ \hline s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = A \cdot B_{q,1}$$

where $A\epsilon R^{3x9}$, $B_{q,1}\epsilon R^{9x3}[s]$.

For $\omega_1=(1,2)\epsilon Q_{2,3}$

For $a_1=(1,2)\epsilon Q_{2,3}$

$$m_{\omega_{1,1}}(s) = \sum_r \text{sgn}(r)a_r^{a_1}\cdot s^{W(r)=s^2} \quad \text{where}$$

$r\epsilon\Sigma=\{(1,2), (1,5), (1,8), (2,4), (2,7), (4,5), (4,8), (5,7), (7,8)\}$ the set of all the P-Prime and ω_1-Prime sequences of $Q_{2,9}$

For $a_2=(1,3)\epsilon Q_{2,3}$

$$m_{\omega_{1,2}}(s) = \sum_r \text{sgn}(r)a_r^{a_2}\cdot s^{W(r)} = -s^3, \ r\epsilon\Sigma$$

For $a_3=(2,3)\epsilon Q_{2,3}$

$$m_{\omega_{1,3}}(s) = \sum_r \text{sgn}(r)a_r^{a_3}\cdot s^{W(r)} = -s^4-6s^3+2s^2, \quad r\epsilon\Sigma$$

$$\underline{m}_{\omega_1}(s) = [s^2, -s^3, -s^4-6s^3+2s^2]^t$$

The same process will be repeated for the rest sequences $\omega_2=(1,3)$ and $\omega_3=(2,3)$ of $Q_{2,3}$. Finally we obtain that

$$C_2(M(s))=[\underline{m}_{\omega_1}(s),\underline{m}_{\omega_2}(s),\underline{m}_{\omega_3}(s)]=$$

$$= \begin{bmatrix} s2 & -2s3 & -s2+s \\ -s^3 & -4s^3+3s^2 & s^3 \\ -s^4-6s^3+2s^2 & -7s^3+s^2 & s^4+3s^3-3s^2+2s \end{bmatrix}$$

■

Next we present an algorithm for the evaluation of $C_p(M(s))$.

Algorithm COMPOL3

Let $M(s)\epsilon R^{mx1}[s]$, $m\geq 1$, $1\leq p\leq 1$, q the maximum degree of the polynomials and $A\epsilon R^{mx1(q+1)}$ the matrix of (4.21). The following algorithm evaluates each element of $C_p(M(s))=(comp)_{ij}\epsilon R^{\binom{m}{p}x\binom{1}{p}}[s]$ and directly prints it.

for each sequence $\omega=(\omega_{i_1},\ldots,\omega_{i_p})\epsilon Q_{p,l}$

  for each sequence $a=(a_{i_1},a_{i_2},\ldots,a_{i_p})\epsilon Q_{p,m}$

    for each sequence $r=(r_{i_1},r_{i_2},\ldots,r_{i_p})\epsilon Q_{p,l(q+l)}$

      If r is P-Prime and ω-Prime **then**

        comp := $\mathrm{sgn}(r)\cdot a_r^a\cdot s^{W(r)}$

        print comp

Alg: 4.9

Remark (4.3):

(i) A similar technique for computing the p-th compound matrix of a polynomial matrix $M(s)\epsilon R^{mxl}[s]$, $1\leq p\leq l$ based on analysis (4.27) can be developed.

(ii) All the algorithms discussed so far deal with matrices $M(s)\epsilon R^{mxl}[s]$, $m\geq l$. If we want to evaluate $C_p(M(s))$, of a given matrix $M(s)\epsilon R^{mxl}$, $m<l$, $1\leq p<m$, we form $M^T(s)\epsilon R^{lxm}[s]$, $l\geq m$ and we compute $C_p(M^T(s))$. Then applying Property (4.8) of compound matrices we have:

$$C_p(M(s)) = (C_p(M^T(s))^T$$

(iii) If $M(s)\epsilon R^{mxl}[s]$, $m=l$ then $C_m(M(s))=\det(M(s))$. Thus, one way of computing the determinant of a polynomial mxm matrix is by finding its m-th compound matrix.

■

4.5.3 Applications of the numerical algorithm

The evaluation of the compound of a polynomial matrix can have a lot of interesting applications in various fields. Some of the most useful are:

(I) Computation of Smith Normal Form

We recall [Ayr., 1] that every square polynomial matrix A(s), of rank r can be reduced by elementary transformations to the Smith normal form

$$N(s) = \begin{bmatrix} f_1(s) & 0 & \dots & 0 & \dots & 0 \\ 0 & f_2(s) & \dots & 0 & \dots & 0 \\ . & . & . . . . & 0 & \dots & 0 \\ 0 & 0 & \dots & f_r(s) & \dots & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 \\ . & . & . . . . & . & . & . \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \qquad (4.36)$$

where each $f_i(s)$ is monic and $f_i(s)$ divides $f_{i+1}(s)$, $i=1,2,\dots,r-1$. The polynomials $f_1(s),\dots,f_r(s)$ are called invariant factors of $M(s)$.
One way of achieving the Smith normal form of a polynomial matrix is by using the notion of compound matrices. Next this technique is demonstrated.

Algorithm SMITH

For a given polynomial matrix $M(s)\epsilon R^{l\times l}[s]$, $\rho(M(s))=r$, the following algorithm computes the invariant factors of the Smith normal form of matrix $M(s)$.

$D_0(s) := 1$

for $p := 1,\dots,r$

evaluate $C_p(M(s))=(c_{ij}(s))$, $i,j=1,2,\dots,\binom{l}{p}$

find $D_p(s) := \text{g.c.d.}\{c_{ij}(s),\ i,j=1,2,\dots,\binom{l}{p}\}$

$$f_p(s) := \frac{D_p(s)}{D_{p-1}(s)}$$

Alg: 4.10

Implementation of the algorithm

For the evaluation of $C_p(M(s))$, algorithm **COMPOL3** can be used. The computation of the greatest common divisor (g.c.d.) of the polynomial entries

of each $C_p(M(s))$ can be done by applying the algorithm developed in Chapter 7. In this case, the construction of the basis matrix of each polynomial set $\{(c_{ij}),\ i,j=1,2,\ldots,\binom{l}{p}\}$ requires careful handling.

Example (4.8): Determine the Smith normal form of the matrix:

$$M(s) = \begin{bmatrix} s & s & s-1 \\ s^2+s & s^2+2s & s^2-1 \\ 2s^2-2s & s^2-2s & 2s^2-3s+2 \end{bmatrix} \epsilon R^{3x3}[s],\ \rho(M(s))=3$$

$D_0(s) = 1,\ C_1(M(s)) = M(s),\ D_1(s) = 1,\ f_1(s) = 1.$

From Example (4.7) we have that

$$C_2(M(s)) = \begin{bmatrix} s^2 & -2s^3 & -s^2+s \\ -s^3 & -4s^3+3s^2 & s^3 \\ -s^4-6s^3+2s^2 & -7s^3+s^2 & s^4+3s^3-3s^2+2s \end{bmatrix}$$

$D_2(s) = s,\ f_2(s) = s$

Using Algorithm **COMPOL1** or **COMPOL2** we can find

$$C_3(M(s)) = s^3,\ f_3(s) = s^2$$

Thus, the Smith normal form of M(s) is:

$$N(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s^2 \end{bmatrix}$$

■

(II) Computation of Plucker matrices

Let $M(s) = [\underline{m}_1(s),\ldots,\underline{m}_q(s)]\epsilon R^{pxq}[s],\ p\geq q,\ \sigma = \binom{p}{q},\ \rho_{R(s)}(M(s)) = q$, and assume that M(s) has no finite zeros. If we denote $V_M = \text{col-sp}_{R(s)}\{M(s)\}$, then $\underline{m}_i(s)\wedge\ldots\wedge\underline{m}_q(s) \equiv m(s)\wedge = \underline{g}(V_M)$ is Known as a Grassmann Representative (GR) of V_M [Kar. & Gia., 1].

$\underline{g}(V_M)$ uniquely characterises V_M (mod R(s)) and if $\delta \equiv \deg\{\underline{g}(V_M)\}$, then we may write

$$\underline{g}(V_M) = P_\delta \cdot \underline{e}_\delta(s),\ P_\delta \epsilon R^{\sigma x(\delta+1)},\ \underline{e}_\delta(s)=[1,s,\ldots,s^\delta]^t \tag{4.37}$$

The basis matrix P_δ of $\underline{g}(V_M)$ is referred to as the Plucker matrix of V_M [Kar. & Gia., 1].

Using an analysis similar with that developed in (4.27) we will formulate a procedure achieving the evaluation of Plucker matrices.

a. General formulation

Let $M(s)=[\underline{m}_1(s),\ldots,\underline{m}_q(s)]\epsilon R^{pxq}[s]$, $\rho_C(M(s))=q$, $\forall\ s\epsilon C$, $\deg\{\underline{m}_i(s)\}=\delta_i$, $\underline{m}_i(s)=M_i\underline{e}_{\delta_i}(s)$, $M_i\epsilon R^{\rho x(\delta_i+1)}$ and assume that M(s) is also column reduced and ordered according to ascending degrees (i.e. $0\leq\delta_1\leq\ldots\leq\delta_q$), then

$$M(s)=[M_1,\ldots,M_q]\cdot\begin{bmatrix} \underline{e}_{\delta_1}(s) & & \\ & \ddots & \\ & & \underline{e}_{\delta_q}(s) \end{bmatrix} \equiv T_M\cdot S(s) \tag{4.38}$$

where $T_M\epsilon R^{PxK}$, $k=\sum_{i=1}^{q}(\delta_i+1)$ is the coefficient matrix of M(s) and $S(s)\epsilon R^{kxq}[s]$ is the structure matrix of M(s) defined by the index $J=\{\delta_i: 0\leq\delta_i\leq\ldots\leq\delta_q\}$.

Combining the definition of column Grassmann product of matrix M(s) and (4.38) we conclude that

$$C_q(M(s))=P_M\cdot\underline{e}_\delta(s) = C_q(T_M)C_q(S(s)) \tag{4.39a}$$

Note that

$$C_q(S(s))=\underline{g}(S(s)) = P_S\cdot\underline{e}_\delta(s) \equiv \underline{g}_S(s) \tag{4.39b}$$

where

$$\underline{g}(S(s))\epsilon R^v[s],\ v=\binom{k}{q},\ \deg\{\underline{g}_S(s)\}=\delta=\sum_{i=1}^{q}\delta_i,\ \text{and}\ P_S\epsilon R^{\binom{k}{q}x(\delta+1)}$$

Thus

$$P_M = C_q(T_M)P_S \tag{4.40}$$

Since the structure of P_S will define which part of $C_q(T_M)$ is essential for the structure of P_M, the problem of defining the structure of $g_{S(s)}$ is considered first.

b. Properties and evaluation of $g_{S(s)}$

Every entry in $g_S(s)$ can be parametrised by a sequence $\omega=(\omega_1,\omega_2,\ldots,\omega_q)\epsilon Q_{q,k}$ and thus $g_S(s)$ may be denoted by $g_S(s)=[\ldots g_\omega(s)\ldots]^t$, $\omega=(i_1,i_2,\ldots,i_q)\epsilon Q_{q,k}$. $g_\omega(s)$ are referred to as Plucker coordinates of $S_{(s)}$.

The issues arising are:

i) Define $\omega\epsilon Q_{q,k}$ such that $g_\omega(s)\equiv 0$

ii) Define the form of nonzero $g_\omega(s)$ and their corresponding location (in terms of ω)

We introduce first some notation:

Notation (4.4): The interval of integers $[1,\ldots,k]$ is parti ioned into intervals as shown below:

$$\Delta_1=[1,2,\ldots,\delta_1+1],\ \Delta_2=[\delta_1+2,\ldots,\delta_1+\delta_2+2],\ldots,\Delta_q=[\delta_1+\ldots+\delta_{q-1}+q,\ldots,k] \qquad (4.41)$$

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

If $l\epsilon\Delta_i$, then its stathm $\sigma(l)$ is defined by: $\sigma(l)=k-\sum_{j=1}^{i-1}(\delta_j+1)-1$.

■

Proposition (4.8): The Plucker coordinates $g_\omega(s)$, $\omega=(i_1,..,i_q)\epsilon Q_{q,k}$ have the following properties:

(i) $g_\omega(s)\not\equiv 0$, iff $i_1\epsilon\Delta_1$, $i_2\epsilon\Delta_2,\ldots,i_q\epsilon\Delta_q$.

(ii) $g_\omega(s)\equiv 0$, if at least two indices in ω are taken from the same interval

(iii) If $i_1\epsilon\Delta_1$, $i_2\epsilon\Delta_2,\ldots,i_q\epsilon\Delta_q$, then

$$g_\omega(s) = s^{\sigma(i_1)+\ldots+\sigma(i_q)} \qquad (4.42)$$

■

Definition (4.3): A sequence $\omega=(i_1,\ldots,i_q)\epsilon Q_{q,k}$ for which $i_1\epsilon\Delta_1$, $i_2\epsilon\Delta_2,\ldots,i_q\epsilon\Delta_q$ is called nonsingular; otherwise, i.e. if more than one indices are taken from the same interval is called singular. The set of nonsingular sequences of $Q_{q,k}$ is denoted by $\Omega_{q,k}(\delta_1,\ldots,\delta_q)$.

■

From Proposition (4.8) it is evident that the singular sequences of $Q_{q,k}$ define the zero Plucker coordinates, whereas the nonsingular ones the nonzero elements.

In the sequel, a procedure defining the set $\Omega_{q,k}(\delta_1,\ldots,\delta_q)$ is suggested.

Algorithm NONSINGSEQ

Let q, $0\leq\delta_1\leq\ldots\leq\delta_q$ be given integers, $k=\sum_{i=1}^{q}(\delta_i+1)$. The following algorithm evaluates the nonsingular sequences $\omega\epsilon\Omega_{q,k}(\delta_1,\ldots,\delta_q)$.

STEP 1:Define the index sets Δ_i

for i = 1,...,q

$\Delta_i=\{\delta_1+\ldots+\delta_{i-1}+i,\delta_1+\ldots+\delta_{i-1}+i+1,\ldots\ldots,\delta_1+\delta_2+\ldots\delta_i+i\}$

STEP 2:Each $\omega\epsilon\Omega_{q,k}(\delta_1,\ldots,\delta_q)$ is defined as a path passing once through the elements of each of the Δ_i index sets.

Alg: 4.11

The following algorithm computes the so called stathm representative $\sigma(\omega)$ of of each sequence $\omega=(i_1,i_2,\ldots,i_q)\epsilon\Omega_{q,k}(\delta_1,\ldots,\delta_q)$.

Algorithm STATHMREP

STEP 1: Define the stathm sets $\{\Delta_i\}$

for i = 1,...,q

$\{\Delta_i\}$ = $\{0,1,\ldots,\delta_i\}$

STEP 2: **for** each $\omega=(i_1,\ldots,i_q)\epsilon\Omega_{q,k}(\delta_1,\ldots,\delta_q)$

$\sigma(\omega)$:= $(\sigma(i_1),\sigma(i_2),\ldots,\sigma(i_q))$

Alg: 4.12

There is an one to one mapping between $\Omega_{q,k}(\delta_1,\ldots,\delta_q)$ and $\Sigma\{\Omega_{q,k}(\delta_1,\ldots,\delta_q)\}$ where $\Sigma\{\Omega_{q,k}(\delta_1,\ldots,\delta_q)\}$ is the set of the stathms representatives of all $\omega\epsilon\Omega_{q,k}(\delta_1,\ldots,\delta_q)$.

The vector consisting only from the nonzero Plucker coordinates of the Grassmann vector $\underline{g}_S(s)$ is defined as the

reduced Grassmann vector $\quad \overset{r}{\underline{g}}_S(s)\epsilon R^{\tau},\ \tau=\prod_{i=1}^{q}(\delta_i+1)$.

The submatrix of P_S obtained by deleting the zero rows (that are parametrised by a singular sequence $\omega\epsilon Q_{q,k}$) without changing the relative position of the rows characterised by the nonsingular sequences, is defined as the reduced structure matrix and it is denoted by $\overset{r}{P}_S\epsilon R^{\tau x(\delta+1)}$.

By computing the sets $\{\Delta_i\}$ $i=1,2,\ldots,q$ defined by algorithm **STATHMREP**, we derive the following procedure for the construction of $\overset{r}{\underline{g}}_S(s)$ and $\overset{r}{P}_S$.

Algorithm GRASREP

STEP 1: Define the composition set $\{\Delta_1,\ldots,\Delta_{q-1}\}$

$\{\Delta_1\} := \{\omega_{i(1)}:\omega_{i(1)}=(i(1)-1),\ i(1)\epsilon\underline{\delta_1+1}\}$

$=\{(0),(1),\ldots,(\delta_1)\}$ ordered lexicographically

for m = 2,...,q-1

$\{\Delta_1,\Delta_2,\ldots,\Delta_m\}:=\{\omega_{i(1)\ldots i(m-1)i(m)}:\omega_{i(1)\ldots i(m-1)i(m)}$

$=(\omega_{i(1)\ldots i(m-1)},i(m)-1),$

for all $\omega_{i(1)\ldots i(m-1)}\epsilon\{\Delta_1,\ldots,\Delta_{m-1}\}$ and $i(m)\epsilon\underline{\delta_m+1}\}$

STEP 2: **for** every $\omega_{i(1)\ldots i(q-1)}=\{x_1,\ldots,x_{q-1}\}\epsilon\{\Delta_1,\ldots,\Delta_{q-1}\}$

$$f_{x_1,x_2,\ldots,x_{q-1}} := s^{x_1+\ldots+x_{q-1}}\cdot \underline{e}_{\delta_q}(s)$$

$\longleftarrow\ \delta_{q+1}\longrightarrow$

STEP 3: $g_S^r(s) := [\ldots|\ f_{x_1,x_2,\ldots,x_{q-1}}|\ldots]$

$\uparrow$ $(x_1,\ldots,x_{q-1})$ position

STEP 4: $Z := x_1+x_2+\ldots+x_{q-1}$

$$P_S^r := \delta_{q+1}\uparrow\downarrow \begin{bmatrix} \vdots & & \\ \cdots & \cdots & \cdots \\ 0 & I_{\delta_{q+1}} & 0 \\ \cdots & \cdots & \cdots \\ \vdots & & \end{bmatrix} \longleftarrow (x_1,x_2,\ldots,x_{q-1})\text{position}$$

Alg: 4.13

From the above, the following Proposition readily follows:

Proposition (4.9): If $\omega=(x_1,\ldots,x_q)\epsilon\{\Delta_1,\ldots,\Delta_q\}$ is a stathm representation, then

$$g_\omega(s) = s^{x_1+x_2+\ldots+x_q}$$

■

c. Computation of Plucker matrices

Using the analysis developed in b., we are led to the following formulation.

The submatrix of $C_q(T_M)$ obtained by deleting all columns associated with the singular sequences of $Q_{q,k}$ without changing the relative position of the columns associated with the nonsingular sequences, is called the reduced compound and denoted by $C_q^r(T_M)$. Clearly,

$$P_M = C_q^r(T_M)\cdot P_S^r \qquad (4.43)$$

In the sequel, we present an algorithm for the evaluation of the Plucker matrix P_M.

Algorithm PLUCKER

Let T_M be the matrix of (4.38). The following algorithm evaluates the Plucker matrix $P_M=[p_0,p_1,\ldots,p_\delta]$ using the analysis established in b.

STEP 1: Construct the sets $\Omega_{q,k}(\delta_1,\ldots,\delta_q)$, $\Sigma\{\Omega_{q,k}(\delta_1,\ldots,\delta_q)\}$

STEP 2: **for** $m = 0,\ldots,\delta$

construct the set

$\Omega_m := \{\omega=(x_1,\ldots,x_q):\omega\epsilon\Omega_{q,k}(\delta_1,\ldots,\delta_q),\ x_1+x_2+\ldots+x_q=m\}$

STEP 3: Construct $C_q^{\rho}(T_M)=[\ldots,t_\omega,\ldots]$

STEP 4: **for** $m = 0,\ldots,\delta$

$p_m := \sum_{\omega\epsilon\Omega_m} t_\omega$

Alg: 4.14

Example (4.9): Let

$$M(s)=\begin{bmatrix} 3s+1 & 2s^2 \\ 5s+2 & 2s+3 \\ 3s+2 & 5s^2+1 \end{bmatrix} \epsilon R^{3x2}[s]$$ be a polynomial matrix.

Then, $p=3$, $q=2$, $\delta_1=1$, $\delta_2=2$, $k=5$, $\delta=3$. Matrix $M(s)$ can be expressed as:

$$M(s)=\begin{bmatrix} 1 & 3 & 0 & 0 & 2 \\ 2 & 5 & 3 & 2 & 0 \\ 2 & 3 & 1 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ s & 0 \\ 0 & 1 \\ 0 & s \\ 0 & s^2 \end{bmatrix} = T_M \cdot S_{(s)}$$

Thus, $$P_M = C_2^r\left(\begin{bmatrix} 1 & 3 & 0 & 0 & 2 \\ 2 & 5 & 3 & 2 & 0 \\ 2 & 3 & 1 & 0 & 5 \end{bmatrix}\right)P_S^r$$

Applying algorithm **PLUCKER** and using also algorithms **NONSINGSEQ** and **STATHMREP** we have:

$$\Omega_{2,5}(1,2)=\{(1\ 3),(1\ 4),(1\ 5),(2\ 3),(2\ 4),(2\ 5)\}$$

$$\Sigma\{\Omega_{2,5}(1,2)=\{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2)\}$$

$$\Omega_0=\{(0,0)\},\ \Omega_1=\{(0,1),(1,0)\},\ \Omega_2=\{(0,2),(1,1)\},\ \Omega_3=\{(1,2)\}$$

$$C_2^r(T_M)=\begin{bmatrix} 3 & 2 & -4 & 9 & 6 & -10 \\ 1 & 0 & 1 & 3 & 0 & 9 \\ -4 & -4 & 10 & -4 & -6 & 25 \end{bmatrix}=[\underline{t}_{13}\ \underline{t}_{14}\ \underline{t}_{15}\ \underline{t}_{23}\ \underline{t}_{24}\ \underline{t}_{25}]$$

Finally,

$$P_M = [\underline{p}_0,\underline{p}_1,\underline{p}_2,\underline{p}_3]\ \text{,where}$$

$\underline{p}_0=\underline{t}_{13}$, $\underline{p}_1=\underline{t}_{14}+\underline{t}_{23}$, $\underline{p}_2=\underline{t}_{15}+\underline{t}_{24}$, $\underline{p}_3=\underline{t}_{25}$, thus

$$P_M=\begin{bmatrix} 3 & 11 & 2 & -10 \\ 1 & 3 & 1 & 9 \\ -4 & -8 & 4 & 25 \end{bmatrix}$$

■

Remark (4.4) : If we are given a matrix $M(s)\epsilon R^{pxq}[s]$ that is not column reduced, using appropriate transformations we can modify it to the form required for the above analysis.

■

4.6 CONCLUSIONS

The aim of this Chapter was to develop efficient numerical algorithms for handling computations arising from Exterior Algebra. A brief summary of the

most important results from Exterior Algebra was presented in the beginning and in the sequel were defined some new notions concerning sequences of integers. These definitions were applied for the development of algorithms achieving the computation of compounds of polynomial matrices. Thus, this Chapter serves the following purposes:

(i) It provides a quick review of the most important results from the area of Exterior Algebra.

(ii) It provides efficient algorithms for computing the exterior product of real vectors and the compounds of real matrices.

(iii) It provides efficient algorithms for computing the compounds of polynomial matrices.

(iv) It provides an efficient algorithm for the evaluation of Plucker matrices.

CHAPTER 5

SPECIAL NUMERICAL TECHNIQUES FOR HANDLING NONGENERIC COMPUTATIONS

5.1 INTRODUCTION

In this Chapter special numerical techniques handling nongeneric computations are presented. The notion of nongeneric computations was analytically introduced in Chapter 3. As a matter of fact, a lot of substantial computational problems possess a nongeneric nature (g.c.d. of polynomials, rank deficiency of a matrix). When we are trying to derive numerical algorithms for them, extra care must be taken and special techniques must be deployed in order to end up with a stable algorithm catching correctly the desired solutions.

In the beginning of this Chapter it is briefly reported the most powerful tool, that provides the means for encountering the nongenericity, the Singular Value Decomposition theorem. The most characteristic properties of the singular values are also summarized. In the sequel, other useful numerical techniques are demonstrated. The replacement of the usual notion of the rank of a matrix with the corresponding numerical ε-rank, for a given accuracy ε is introduced. The notion of numerical ε-rank will be used in almost all the proposed methods.

When we are given a set of vectors it is required to determine the relationship between its members. Therefore, the notions of ε-independent, numerically ε-dependent, strongly ε-dependent, fuzzy ε-dependent sets of vectors are introduced. The strongly ε-dependency property of a set provides its unity numerical ε-rank.

Since in many nongeneric computations the appearance of sets of vectors with unity numerical ε-rank takes place, a cautious study of their properties is something indispensable. A useful theorem providing necessary and sufficient conditions between the numerical ε-rank of a strongly ε-dependent set and its singular values is developed. The problem of selecting a "best" representative of such sets is also faced.

The selection of a "best uncorrupted" base for the row space of a matrix is considered next. A useful algorithm combining the notions of Gram matrix and compound matrix is demonstrated analytically.

Furthermore, a detailed survey concerning the most important properties of the Gramian of given vectors and the Schur complement -tools extremely useful due to their ability in handling determinants- is presented.

5.2 THE SINGULAR VALUE DECOMPOSITION (S.V.D.)

One of the basic and most important tools of modern Numerical Analysis, particularly Numerical Linear Algebra, is the singular value decomposition.

Theorem (5.1) [Gol. & Loan, 1]: If $A \epsilon R^{mxn}$ then there exist orthogonal matrices

$$U = [\underline{u}_1,\ldots,\underline{u}_m] \epsilon R^{mxm}$$

and

$$V = [\underline{v}_1,\ldots,\underline{v}_n] \epsilon R^{nxm}$$

such that $$U^T A V = \Sigma \qquad (5.1)$$

where $\Sigma \epsilon R^{mxn}$ a "diagonal" matrix, with nonnegative "diagonal" elements σ_i, i.e. with $\Sigma_{ii}=\sigma_i$ and $\Sigma_{ij}=0 (i \neq j)$, satisfying $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{m,n\}} \geq 0$. ■

The numbers σ_i are the nonnegative square roots of the eigenvalues of AA^T, and hence are uniquely determined. The columns of U are eigenvectors of AA^T and the columns of V are eigenvectors of A^TA (arranged in the same order as the corresponding eigenvalues σ_i^2).

The "diagonal entries" $\sigma_i=\Sigma_{ii}$, i=1,2,..., p=min{m,n} of Σ are known as the singular values of $A \epsilon R^{mxn}$ (sometimes only the nonzero ones are so termed), and the columns of U and V are the (respectively, left and right) singular vectors of A. The factorization (5.1) is known as the Singular Value Decomposition of A (S.V.D.). It is easy to verify that

$$A\underline{v}_i = \sigma_i \underline{u}_i$$

$$A^T\underline{u}_i = \sigma_i \underline{v}_i$$

Some of the most important results characterizing the S.V.D. of a matrix are summarized next.

Corollary (5.1) [Gol. & Loan, 1]: If the S.V.D. of matrix $A \epsilon R^{mxn}$ is given by Theorem (5.1) and $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0$, then

(i) $\rho(A)=r$

(ii) $N(A)=\text{span}\{\underline{v}_{r+1},\ldots,\underline{v}_n\}$

(iii) $R(A)=\text{span}\{\underline{u}_1,\underline{u}_2,\ldots,\underline{u}_r\}$

(iv) $A = \sum_{i=1}^{r} \sigma_i \underline{u}_i \underline{v}_i^t = U_r \Sigma_r V_r^T$ where $U_r=[\underline{u}_1,\ldots,\underline{u}_r]$

$V_r = [\underline{v}_1,\ldots,\underline{v}_r]$ and $\Sigma_r=\text{diag}(\sigma_1,\ldots,\sigma_r)$

(v) $\|A\|_F^2 = \sigma_1^2+\ldots+\sigma_p^2$

(vi) $\|A\|_2 = \sigma_1$.

■

Corollary (5.2) [Gol. & Loan, 1]: Let the S.V.D. of $A \in R^{mxn}$ be given by Theorem (5.1). If $k<r=\rho(A)$ and

$$A_k = \sum_{i=1}^{k} \sigma_i \underline{u}_i \underline{v}_i^t \tag{5.2}$$

then $$\min_{\rho(B)=k} \|A-B\|_2 = \|A-A_k\|_2 = \sigma_{k+1}$$

■

Corollary (5.2) says that the smallest singular value of A is the 2-norm distance of A to the set of all rank-deficient matrices.

Corollary (5.3) [Horn, 1]: Let $A,B \in R^{mxn}$, let E=B-A, and let $p=\min\{m,n\}$. If $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p$ are the singular values of A and, $\tau_1 \geq \tau_2 \geq \ldots \geq \tau_p$ are the singular values of B, then

$$\text{(a)} \quad |\sigma_i - \tau_i| \leq \|E\|_2, \quad i=1,2,\ldots,q \tag{5.3}$$

$$\text{(b)} \quad \left[\sum_{i=1}^{p} (\sigma_i - \tau_i)^2\right]^{1/2} \leq \|E\|_2$$

If $\rho(B)=r$, i.e. $\tau_i=0$, $i=r+1,\ldots,n$ then (b) becomes

$$\sum_{i=r+1}^{p} \sigma_i^2 \leq \|E\|_2^2$$

■

The great virtue of the singular value decomposition is that it enables us to deal sensibly with the concept of matrix rank. The plain fact is, that the simplest of the factorizations (Gaussian elimination, QR) give no reliable indication of the proximity of a matrix to rank deficiency even when rounding errors are not involved. On the contrary, the S.V.D. gives a perfectly reliable indication and it can be performed in such a stable manner that rounding errors do not bring any significant complications.
Notice that the rank r of A is equal to the number of nonzero singular values of A, thus one way to compute the rank of A <u>numerically</u>, is to compute a S.V.D. and take the rank of A to be the number of singular values that are larger than some threshold. The key quantity in rank determination is obviously σ_r. Moreover, this number gives a dependable measure of how far (in a $\|\ \|_2$ sense) a matrix is from matrices of lesser rank. But σ_r alone is

clearly sensitive to scale so that a better measure is $\sigma_r/\|A\|_2$. But $\|A\|_2=\sigma_1$, therefore the important quantity is σ_r/σ_1 which turns out to be the reciprocal of the number $k(A)=\|A\|_2\|A^+\|_2$, the so called "condition number of A". In the case when A is invertible, $k(A)=\|A\|_2\|A^{-1}\|_2$ is the usual spectral condition number. The numerical determination of the rank of A is easier and more accurate if $1/k(A)$ is not near to zero.

The algorithm in most common use for the S.V.D. is due to [Gol. & Rein., 1] and is extremely stable. It can be shown that the computed singular values are the exact singular values of some matrix A+G where

$$\|G\|_2 \leq \beta^{-t}\|A\|_2 \cdot f(n,m)$$

β^{-t} being the computer precision and f(n,m) a very "modest" function of n and m. Hence if A is close to rank degeneracy this will be revealed reliably even by the computed σ_i, i=1,2,...,r. In fact, when A arises from physical measurements, then unless the computer precision is exceptionally low, the equivalent perturbation in A resulting from rounding errors will usually be far smaller than the original perturbation in A arising from inaccuracies in the data.

5.3 RANK DEGENERACY-THE NUMERICAL E-RANK

The usual mathematical notion of rank is not very useful when the matrices in question are not known exactly. For example, suppose that A is a mxn matrix that was originally of rank r<n but whose elements have been perturbed by some small errors (e.g. rounding or measurement errors). It is extremely unlikely that these errors will contribute to keep the rank of A exactly equal to r; indeed what is most likely is that the perturbed matrix will have full rank n. Nonetheless, the nearness of A to a matrix of defective rank will often cause it to behave erratically when it is subjected to statistical and numerical algorithms.

One way of overcoming the difficulties of the mathematical definition of rank is to specify a tolerance and say that A is numerically defective in rank if to within that tolerance it is near a defective matrix [Gol. & Klem. & Stew., 1] . Specifically we might say that A has ε-rank r with respect to the norm $\|\cdot\|$ if

$$r = \inf\{\rho(B):\|A-B\|<\varepsilon\} \qquad (5.4)$$

However, this definition has the defect that a slight increase in ε can decrease the numerical rank. What is needed is an upper bound on the values of ε for which the numerical rank remains at least equal to r. Such a number is provided by any number δ satisfying

$$\varepsilon<\delta\leq\sup\{n:\|A-B\|\leq n\ \rho(B)\geq r\} \tag{5.5}$$

Accordingly we make the following definition.

Definition (5.1) [Gol., Klem. & Stew., 1]: A matrix A has numerical rank (δ,ε,r) with respect to the norm $\|\cdot\|$ if δ,ε, and r satisfy (5.4) and (5.5).

■

When the norm in Definition (5.1) is either the 2-norm or the Frobenius norm, the problem of determining the numerical rank of a matrix can be solved in terms of the singular value decomposition of the matrix.

The following Theorem was stated in [Gol., Klem. & Stew., 1] without proof.

Theorem (5.2) [Gol., Klem. & Stew., 1]: Let $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_p$, $p=\min\{m,n\}$ be the singular values of $A\epsilon R^{mxn}$. Then A has numerical rank (δ,ε,r) with respect to the $\|\ \|_2$ if and only if

$$\sigma_r \geq \delta < \varepsilon \geq \sigma_{r+1} \tag{5.6}$$

Proof: Suppose that (5.6) holds. Then by (5.2) if $\|A-B\|_2<\delta$ we must have $\rho(B)\geq r$. Consequently, δ satisfies (5.5). This also shows that

$$\min\{\rho(B):\|B-A\|\leq\varepsilon\}\geq r$$

But the matrix $B=U\Sigma_1V^T$, where U,V are the orthogonal matrices of the S.V.D. of matrix A (i.e. $A=U\Sigma V^T$) and $\Sigma_1=\mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_r,0,\ldots,0)$ has rank r and satisfies $\|B-A\|_2\leq\varepsilon$. Hence ε satisfies (5.4).

Conversely, suppose δ,ε, and r satisfy (5.4) and (5.5). Then by (5.2), $\delta\leq\sigma_r$. Also $\varepsilon\geq\sigma_{r+1}$; for if not by (5.4) there is a matrix B of rank r satisfying $\|A-B\|<\sigma_{r+1}$ which contradicts (5.5).

■

Remark (5.1): If we use the Frobenius norm, Theorem (5.2) can be stated as: A has numerical rank (δ,ε,r) with respect to the $\|\ \|_F$ if and only if

$$\sigma_r^2+\sigma_{r+1}^2+\ldots+\sigma_p^2 \geq\delta^2 >\varepsilon^2 \geq\sigma_{r+1}^2+\ldots+\sigma_p^2$$

The proof is quite similar with that of the $\| \ \|_2$ norm.

■

Because of the simplicity of the characterization (5.6) we shall restrict ourselves to rank defectiveness measured in terms of the spectral norm.

According to the previous definitions of numerical ε-rank, the notion of numerical ε-nullity ($n_\varepsilon(A)$) can be defined as well.

Definition (5.2). [Fost., 1]: The numerical ε-nullity of a matrix $A\epsilon R^{mxn}$ is defined by:

$$n_\varepsilon(A) = \max\{\text{nullity}(B): \|A-B\|_2 \leq \varepsilon\}. \tag{5.7}$$

■

Given a vector y in Euclidean space R^m, and a subspace S of R^m, we define the distance between y and S by $\text{dist}(y,S)=\min\{\|y-z\|: z\epsilon S\}$. A more simplified condition for the determination of the numerical ε-rank ($\rho_\varepsilon(A)$) and the numerical ε-nullity ($n_\varepsilon(A)$) of a matrix A is given next.

Theorem (5.3) [Fost., 1]: For a matrix $A\epsilon R^{mxn}$, and a specified tolerance ε

$\rho_\varepsilon(A)=\min\{\text{dimension (S): S is a subspace of } R^m \text{ such}$

$$\text{that } x\epsilon R^n, x\neq 0 \ \frac{\text{dist}(Ax,S)}{\|x\|} \leq \varepsilon\} \tag{5.8}$$

$$\rho_\varepsilon(A)=\text{number of singular values of A that are} > \varepsilon \tag{5.9}$$

$$n_\varepsilon(A)=\max_S\{\text{dimension (S): S is a subspace of } R^n \text{ such}$$

$$\text{that } x\epsilon S,\ x\neq 0 \ \frac{\|Ax\|}{\|x\|} \leq \varepsilon\} \tag{5.10}$$

$$n_\varepsilon(A)=\text{number of singular values that are} \leq \varepsilon \tag{5.11}$$

$$\rho_\varepsilon(A)=n-n_\varepsilon(A). \tag{5.12}$$

■

The results (5.8) and (5.10) are geometrical, (5.8) stating that the numerical ε-rank is the smallest dimension of a space that approximates well vectors in the range of A, and (5.10) stating that the numerical ε-nullity is the largest dimension of a space that is approximately annihilated by A. A subspace S of dimension $n_\varepsilon(A)$ that satisfies (5.10) will be called an ε-null space of A. The results (5.9) and (5.11) suggest one method of calculating the numerical rank of A-via the S.V.D.

Let $A=\{\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_n\}$ be a set of n vectors $\underline{a}_i \epsilon R^{mx1}$, i=1,2,...,m. This set can be expressed in terms of a matrix $A=[\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_n]^t \epsilon R^{mxn}$, having as its rows the given vectors. Let $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$, be the singular values of A. It is extremely useful to determine the relationship of vectors $\underline{a}_i$, i=1,2,...,n in terms of the S.V.D.

Definition (5.3): For a given tolerance ε

(i) The set *A* is ε-independent if $\sigma_i > \varepsilon$, i=1,2,...,r i.e. all the singular values are greater than ε.

(ii) The set *A* is numerically ε-dependent if $\sigma_i > \varepsilon$ and $\sigma_j \leq \varepsilon$ for some i,j, i<j i.e. some singular values are greater than ε and others are smaller than ε.

(iii) The set *A* is strongly ε-dependent if $\sigma_1 > \varepsilon$, $\sigma_i \leq \varepsilon$, i=2,3,...,r i.e. the maximal singular value is greater than ε and all the others are less than ε.

(iv) The set *A* is fuzzy ε-dependent if $\sigma_i \leq \varepsilon$, i=1,2,...,r i.e. all the singular values are less than ε.

■

Since scaling affects the singular values of a matrix, the above definition will be more suitable when it is applied in a normalized set of vectors. By normalizing, the given data are kept under control and thus strange situations as fuzzy ε-dependency of vectors, that are mostly encountered when we are dealing with data of extremely low values (e.g. $\leq 10^{-8}$), are avoided. Throughout this work, normalized sets of vectors will be exclusively used in most cases.

Remark (5.2):

(i) If the set *A* is ε-independent, then $\rho_\varepsilon(A)=r$.

(ii) If the set *A* is numerically ε-dependent, then $\rho_\varepsilon(A)=g<r$.

(iii) If the set *A* is strongly ε-dependent, then $\rho_\varepsilon(A)=1$.

■

Due to the fact that the singular values alone are clearly sensitive to scaling, a better measure for vector's dependency might be $\frac{\sigma_i}{\|A\|_2} = \frac{\sigma_i}{\sigma_1}$. Thus, Definition (5.3) can be stated in the following way when a modest constant $c=f(\varepsilon,\sigma_1)$ is used.

Definition (5.4): For a given constant c

(i) The set A is c-independent if $\frac{\sigma_i}{\sigma_1} > c$, $i=1,2,\ldots,r$

(ii) The set A is numerically c-dependent if

$$\frac{\sigma_i}{\sigma_1} > c \text{ and } \frac{\sigma_j}{\sigma_1} \leq c \text{ for some } i,j$$

(iii) The set A is strongly c-dependent if

$$\frac{\sigma_i}{\sigma_1} \leq c < 1, \quad i=2,3,\ldots,r$$

■

Remark (5.3): Definition (5.4) avoids the determination of the particular case of fuzzy ε-dependent sets of vectors and thus it can be considered as an improved version of Definition (5.3).

■

Example (5.1): (i) Let

$A=\{\underline{a}_1=(0.1,1,1,0)^t, \underline{a}_2=(2.01,3,1,0)^t, \underline{a}_3=(0.01,2,3,1)^t\}$

be a set of vectors $\underline{a}_i \epsilon R^{4x1}$, $i=1,2,3$. The corresponding matrix $A=[\underline{a}_1,\underline{a}_2,\underline{a}_3]^t \epsilon R^{3x4}$ has singular values:

$$\sigma_1 \approx 4.99, \quad \sigma_2 \approx 2.25, \quad \sigma_3 \approx 0.31$$

For any tolerance ε, less than some satisfying accuracy (e.g. $\varepsilon \leq 10^{-5}$) the set is ε-independent

(ii) Let

$A=\{\underline{a}_1=(5,2,5,2,0,0)^t, \underline{a}_2=(1,1,1,0,0)^t, \underline{a}_3=(4,1,4,1,0,0)^t, \underline{a}_4=(5,3,5,3,0,0)^t,$

$\underline{a}_5=(6,6,14,6,8,0)^t, \underline{a}_6=(15,0,18,0,3,0)^t, \underline{a}_7=(3,3,7,3,4,0)^t, \underline{a}_8=(0,7,0,15,0,8)^t\}$

be a set of vectors $\underline{a}_i \epsilon R^{6x1}$, $i=1,2,\ldots,8$. The corresponding matrix $A=[\underline{a}_1,\underline{a}_2,\underline{a}_3,\underline{a}_4,\underline{a}_5,\underline{a}_6,\underline{a}_7,\underline{a}_8] \epsilon R^{8x6}$ has singular values:

$$\sigma_1 \approx 0.33*10^2, \quad \sigma_2 \approx 0.192*10^2, \quad \sigma_3 \approx 0.778*10, \quad \sigma_4 \approx 0.297*10,$$
$$\sigma_5 \approx 0.304*10^{-13}, \quad \sigma_6 \approx 0.104*10^{-13}$$

For $\varepsilon = 0.1*10^{-13}$ the set is numerically ε-dependent

(iii) Let

$A=\{\underline{a}_1=(0.000678,\ 0.000339)^t, \underline{a}_2=(0.000013,\ 0.0000065)^t\}$ be a set of vectors $\underline{a}_i \epsilon R^{2x1}$, $i=1,2$. The corresponding matrix $A=[\underline{a}_1,\underline{a}_2] \epsilon R^{2x2}$ has singular values:

$$\sigma_1 \approx 0.757*10^{-3}, \quad \sigma_2 \approx 0.173*10^{-17}$$

For $\varepsilon=0.1*10^{-3}$ the set is strongly ε-dependent

■

5.4 STRONGLY ε-DEPENDENT SETS OF VECTORS

Frequently, in many applications sets of vectors with numerical ε-rank equal to unity appear. Therefore, it is necessary to examine what properties characterize such sets. Since such sets owe their definition to the values of their singular values according to some specified accuracy ε, a first issue that should be examined is whether the above singular values satisfy any other relations too, apart from these derived from Definition (5.3). A thorough investigation of the conditions governing the corresponding singular vectors must be made too. Another subject that should be also examined has to do with the problem of finding a "best", in a sense to be made precise later, representative of such a set. In other words, since all the vectors of a strongly ε-dependent set are very close to each other, it is necessary to determine a vector that could be considered as the "best" approximation of the set.

Some useful theoretical tools for tackling the above mentioned issues are examined below.

Definition (5.6) [Horn, 1]: Let $\underline{a}$, $\underline{b}\in R^n$ be given. The vector $\underline{b}$ is said to majorize the vector $\underline{a}$ if

$$\min\{\sum_{j=1}^{k} b_{i_j} : 1\le i_1<\ldots<i_k\le n\} \ge \min\{\sum_{j=1}^{k} a_{i_j} : 1\le i_1<\ldots<i_k\le n\} \qquad (5.13)$$

for all k=1,2,...,n with equality for k=n. If we arrange the entries of $\underline{a}$ and $\underline{b}$ in increasing order $a_{j_1}\le a_{j_2}\le\ldots\le a_{j_n}$, $b_{m_1}\le b_{m_2}\le\ldots\le b_{m_n}$, the defining inequalities can be restated in the equivalent form

$$\sum_{i=1}^{k} b_{m_i} \ge \sum_{i=1}^{k} a_{j_i} \quad \text{for all} \quad k=1,2,\ldots,n \qquad (5.14)$$

with equality for k=n.

■

Thus, the real vector $\underline{b}$ majorizes the real vector $\underline{a}$ if the sum of the k smallest entries of $\underline{b}$ is greater than or equal to the sum of the k smallest entries of $\underline{a}$ for k=1,2,...,n-1, and the sums of the entries of $\underline{b}$ and $\underline{a}$ are equal. Notice that the entries of $\underline{b}$ and $\underline{a}$ may be permuted arbitrarily without affecting whether $\underline{b}$ majorizes $\underline{a}$.

The notion of majorization arises in many places in matrix theory as the precise relationship between two sets of real numbers. One example of this phenomenon is the following theorem of Schur (1923).

Theorem (5.4) [Horn, 1]: Let $A \epsilon C^{nxn}$ be Hermitian. The vector of diagonal entries of A majorizes the vector of eigenvalues of A.

■

The following result was suggested in [Horn, 1] without proof.

Proposition (5.1): Let $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$. Order the set of Euclidean norms of the rows $\{\|\underline{r}_i^t\|_2 : i=1,2,\ldots,m\}$ in increasing order and denote the resulting ordered values by $R_1 \leq R_2 \leq \ldots \leq R_m$. The singular values of A are ordered with $\sigma_m \leq \sigma_{m-1} \leq \ldots \leq \sigma_1$. Then,

$$\sum_{i=1}^{k} \sigma_{m-i+1}^2 \leq \sum_{i=1}^{k} R_i^2 \quad \text{for} \quad k=1,2,\ldots,m \tag{5.15}$$

Proof: We form the matrix $A \cdot A^T = C \epsilon R^{mxm}$. Matrix C is symmetric with diagonal entries $c_{ii}=\|\underline{r}_i^t\|_2^2$, $i=1,2,\ldots,m$. From Theorem (5.4), the vector of diagonal entries of C majorizes the vector of eigenvalues of C. If the eigenvalues of C are ordered with $\lambda_m \leq \lambda_{m-1} \leq \ldots \leq \lambda_1$ and let $c_{j_1 j_1} \leq c_{j_2 j_2} \leq \ldots \leq c_{j_m j_m}$ be a rearrangement of the diagonal entries of C into increasing order, then

$$\sum_{i=1}^{k} c_{j_i j_i} \geq \sum_{i=1}^{k} \lambda_{m-i+1}$$

But $\lambda_i=\sigma_i^2$, $i=1,2,\ldots,m$,thus

$$\sum_{i=1}^{k} \sigma_{m-i+1}^2 \leq \sum_{i=1}^{k} R_i^2 \quad \text{for} \quad k=1,2,\ldots,m$$

■

Remark (5.4): Proposition (5.1) shows that if matrix A has a "small" row, then it must also have a "small" singular value.

■

If $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$, then the normalization of A is a matrix $A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t \epsilon R^{mxn}$ with the property: $\underline{v}_i=\underline{r}_i/\|\underline{r}_i\|_2$, i=1,2,...,m; it is obvious that $\underline{v}_i \epsilon R^{nx1}$, i=1,2,...,m are unit length vectors ($\|\underline{v}_i\|_2=1$).

Applying Proposition (5.1) to the case of normalized matrices we readily obtain the following result.

Proposition (5.2): Let $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$, $m \leq n$ be a normalized matrix with singular values $\sigma_m \leq \sigma_{m-1} \leq \ldots \leq \sigma_1$. Then,

$$\sigma_i \leq \sqrt{m-i+1}, \quad i=1,2,\ldots,m. \tag{5.16}$$

■

Actually Proposition (5.2) provides an upper bound for the singular values of a normalized matrix.

More results concerning normalized matrices are given next.

Proposition (5.3): A matrix $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$, $\underline{r}_i \neq \underline{0}$, i=1,2,...,m has $\rho(A)=1$, if and only if the normalization of A is of the form $A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t \epsilon R^{mxn}$, where $\underline{v}_i=a\cdot\underline{v}_1$, i=2,3,...,m and a=+1,-1.

Proof: Suppose that matrix $A \epsilon R^{mxn}$ has $\rho(A)=1$. Then, only one row of A is linearly independent. Without loss of generality we can suppose that $\underline{r}_1^t$ is that row. Then,

$$\underline{r}_i^t = a_i \cdot \underline{r}_1^t, \quad a_i \epsilon R, \quad i=1,2,\ldots,m \tag{5.17}$$

The normalized A_N is of the form:

$$A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t, \quad \underline{v}_i^t = \frac{\underline{r}_i^t}{\|\underline{r}_i^t\|_2}, \quad i=1,2,\ldots,m \tag{5.18}$$

From relation (5.17) we conclude that

$$\|\underline{r}_i^t\|_2=\|a\cdot\underline{r}_1^t\|_2=|a_i|\cdot\|\underline{r}_1^t\|_2, \quad i=1,2,\ldots m \tag{5.19}$$

Combining (5.17), (5.18), (5.19) we have

$$\underline{v}_i^t = \frac{a_i\cdot\underline{r}_1^t}{|a_i|\cdot\|\underline{r}_1^t\|_2} = \frac{a_i}{|a_i|}\cdot\underline{v}_1^t, \quad i=2,3,\ldots,m$$

which can be expressed as:

$$\underline{v}_i^t = a\cdot\underline{v}_1^t, \quad a\epsilon R, \quad a=\pm 1 \tag{5.20}$$

Conversely, let $A_N = [\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t \epsilon R^{mxn}$ the normalization of $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t$. If $\underline{v}_i^t = a\cdot\underline{v}_1^t$, i=2,3,...,m, a=±1 then,

$$\frac{\underline{r}_i^t}{\|\underline{r}_i^t\|_2} = a\cdot\frac{\underline{r}_1^t}{\|\underline{r}_1^t\|_2} \quad \text{which is equivalent to}$$

$$\underline{r}_i = a\cdot\frac{\|\underline{r}_i^t\|_2}{\|\underline{r}_1^t\|_2}\cdot\underline{r}_1, \quad i=2,3,\ldots,m. \quad \text{Thus,}$$

$$\underline{r}_i^t = a_i\cdot\underline{r}_1^t, \quad i=2,3,\ldots,m, \quad a_i = \pm\frac{\|\underline{r}_i^t\|_2}{\|\underline{r}_1^t\|_2} \tag{5.21}$$

Relation (5.21) shows that all the rows of A are linear combinations of the first row, therefore $\rho(A)=1$.

■

The following Proposition will be used in the sequel.

Proposition (5.4): For a,bεR, a,b≠0 the matrix

$$A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_n]^t = \begin{bmatrix} a+b & a & a & \ldots & a \\ a & a+b & a & \ldots & a \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ a & a & a & \ldots & a+b \end{bmatrix} \epsilon R^{nxn}$$

has $\det(A)=b^{n-1}\cdot(na+b)$.

Proof: We apply to matrix A the following transformation:

STEP 1: **for** i := 1,2,...,n-1

$$\underline{r}_i^t := \underline{r}_i^t - \underline{r}_{i+1}^t$$

Thus,

$$A = \begin{bmatrix} b & -b & 0 & \dots & 0 \\ 0 & b & -b & \dots & 0 \\ . & . & . & \dots & . \\ 0 & 0 & . & ..b & -b \\ a & a & . & ..a & a+b \end{bmatrix}$$

STEP 2: **for** i := 1,2,...,n-1

$$\underline{r}_n^t := \underline{r}_n^t - i\frac{a}{b}\,\underline{r}_i^t$$

Thus, A becomes:

$$A = \begin{bmatrix} b & -b & 0 & . & . & 0 \\ 0 & b & -b & . & . & 0 \\ 0 & 0 & b & . & . & 0 \\ . & . & . & . & . & . \\ 0 & 0 & . & . & b & -b \\ 0 & 0 & . & . & 0 & na+b \end{bmatrix}$$

From this upper triangular form we conclude that $\det(A)=b^{n-1}(na+b)$.

■

From the above results we may derive a criterion for the singular values of normalized strongly ε-dependent sets.

Theorem (5.5): Let $A=[\underline{r}_1,\underline{r}_2,\dots,\underline{r}_m]^t \epsilon R^{mxn}$, $m\leq n$, $\underline{r}_i \neq \underline{0}$, i=1,2,...,m. Then $\rho(A)=1$, if and only if the singular values $\sigma_m \leq \sigma_{m-1} \leq \dots \leq \sigma_1$ of the normalization $A_N=[\underline{v}_1,\underline{v}_2,\dots,\underline{v}_m]^t \epsilon R^{mxn}$ of A satisfy the conditions:

$$\sigma_1 = \sqrt{m}, \quad \sigma_i = 0, \quad i = 2,3,\dots,m \tag{5.22}$$

Proof: Suppose that $\rho(A)=1$. Then according to Proposition (5.3) the normalization A_N of A is of the form:

$$A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t \epsilon R^{mxn},\ \underline{v}_i=a\cdot\underline{v}_1,\ i=2,3,\ldots,m,\ a=\pm 1 \tag{5.23}$$

where $\|\underline{v}_i\|_2=1$. This condition yields $(\underline{v}_i\cdot\underline{v}_i)^{1/2}=1$ or

$$\underline{v}_i^t \cdot \underline{v}_i = 1 \tag{5.24}$$

The matrix $A_N\cdot A_N^T = C\epsilon R^{mxn}$ is real and symmetric with real eigenvalues. From (5.23), (5.24) we conclude that

$$|c_{ij}| = 1,\quad i,j=1,2,\ldots,m \tag{5.25}$$

In order to evaluate the eigenvalues of C we form the determinant:

$$D = \det(\lambda I-C) \tag{5.26}$$

Using relation (5.25) and the fact that if any row or column of a matrix is multiplied by a scalar d then the determinant of this matrix is multiplied by d, relation (5.26) yields:

$$D = \det\left\{\begin{bmatrix} \lambda-1 & -1 & \ldots & -1 \\ -1 & \lambda-1 & \ldots & -1 \\ . & . & \ldots & . \\ -1 & -1 & \ldots & \lambda-1 \end{bmatrix}\right\} \tag{5.27}$$

Applying Proposition (5.4) we have: $D=\lambda^{m-1}(\lambda-m)$.

From $\lambda^{m-1}(\lambda-m)=0$, it follows that the eigenvalues of C are: m, 0 with multiplicity m-1. The singular values of A_N are thus: $\sigma_1=\sqrt{m}$, $\sigma_i=0$, $i=2,3,\ldots,m$.

Suppose now that the singular values of $A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t\epsilon R^{mxn}$ are $\sigma_1=\sqrt{m}$, $\sigma_i=0$, $i=2,3,\ldots,m$. Then, the eigenvalues of $C=A_N\cdot A_N^T \epsilon R^{mxm}$, are m, 0 with multiplicity m-1.

Thus $\det(\lambda I-C)$ must be of the form:

$$D = \lambda^{m-1}(\lambda-m) \tag{5.28}$$

From (5.27), Proposition (5.4) and (5.28) we conclude that:

$$D = \det\left\{\begin{bmatrix} \lambda-1 & -1 & \dots & -1 \\ -1 & \lambda-1 & \dots & -1 \\ . & . & \dots & . \\ -1 & -1 & \dots & \lambda-1 \end{bmatrix}\right\} \tag{5.29}$$

Therefore $C\epsilon R^{mxm}$ has $|c_{ij}|=1$, $i,j=1,2,\dots,m$ with $c_{ij}=(\underline{v}_j\cdot\underline{v}_i)$ and consequently

$$|\underline{v}_i^t \cdot \underline{v}_j|=1, \quad i,j = 1,2,\dots,m \tag{5.30}$$

(5.24) and (5.30) yields $\underline{v}_1^t\cdot\underline{v}_1=1=|\underline{v}_i^t\cdot\underline{v}_1|$, $i=2,3,\dots,m$ which implies that

$$\underline{v}_i^t = a\cdot\underline{v}_1^t, \quad i=2,3,\dots,m, \quad a=\pm 1 \tag{5.31}$$

From (5.31) and Proposition (5.3) we conclude that $\rho(A)=1$.

■

The numerical version of Theorem (5.5) may be stated as:

Theorem (5.6): Let $A=[\underline{r}_1,\underline{r}_2,\dots,\underline{r}_m]^t\epsilon R^{mxn}$, $m\leq n$, $\underline{r}_i\neq 0$, $i=1,2,\dots,m$. Then $\rho_\varepsilon(A)\approx 1$, if and only if the singular values $\sigma_m\leq\sigma_{m-1}\leq\dots\leq\sigma_1$ of the normalization $A_N=[\underline{v}_1,\underline{v}_2,\dots,\underline{v}_m]^t\epsilon R^{mxn}$ of A satisfy the conditions:

$$\sigma_1\approx\sqrt{m}, \quad \sigma_i\leq\varepsilon, \quad i=2,3,\dots,m \quad \text{for some } \varepsilon. \tag{5.32}$$

■

From the previous results we derived conditions that hold for the singular values of normalized strongly ε-dependent set of vectors. The next step is to search for any conditions holding for the corresponding singular vectors.

Proposition (5.5): Let $A=[\underline{r}_1,\underline{r}_2,\dots,\underline{r}_m]^t\epsilon R^{mxn}$, $m\leq n$, $\underline{r}_1\neq 0$, $\rho(A)=1$. Let $A_N=[\underline{v}_1,\underline{v}_2,\dots,\underline{v}_m]^t\epsilon R^{mxn}$ be the normalization of A with S.V.D. of the form $U\Sigma W^T$, where $U=[\underline{u}_1,\underline{u}_2,\dots,\underline{u}_m]\epsilon R^{mxm}$, $W=[\underline{w}_1,\underline{w}_2,\dots,\underline{w}_n]\epsilon R^{nxn}$ orthogonal matrices and $\Sigma\epsilon R^{mxn}$ with $\Sigma_{ii}=\sigma_i$ and $\Sigma_{ij}=0$ $(i\neq j)$. Then, the first column $\underline{u}_1$ of matrix U is defined by:

$$u_{i1} = \pm\frac{\sqrt{m}}{m}, \quad i=1,2,\dots,m \tag{5.33}$$

Proof: Let $C=A\cdot A^T\epsilon R^{mxm}$ a real symmetric matrix. From Theorem (5.5) we have that the eigenvalues of C are m and 0 with multiplicity m-1. The first column

$\underline{u}_1$ of matrix U is the eigenvector of C corresponding to the eigenvalue $\lambda=m$. Vector $\underline{u}_1$ satisfy the equation:

$$C\cdot\underline{u}_1 = \lambda\cdot\underline{u}_1 \tag{5.34}$$

or in matrix terms:

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ . & . & \dots & . \\ c_{m1} & c_{m2} & \dots & c_{mm} \end{bmatrix} \cdot \begin{bmatrix} u_{11} \\ u_{21} \\ . \\ u_{m1} \end{bmatrix} = m\cdot \begin{bmatrix} u_{11} \\ u_{21} \\ . \\ u_{m1} \end{bmatrix} \tag{5.35}$$

(5.35) can be written as:

$$\sum_{j=1}^{m} c_{ij}\cdot u_{j1} = m\cdot u_{i1}, \quad i=1,2,\dots,m \tag{5.36}$$

From Theorem (5.5) it is known that $|c_{ij}|=1$.
Thus, after taking absolute values in (5.36) we obtain the following system of equations:

$$\sum_{j=1}^{m} |u_{j1}| = m|u_{i1}|, \quad i=1,2,\dots,m \tag{5.37}$$

or explicitly

$$|u_{11}|+\dots+|u_{i1}|+\dots+|u_{m1}|=m|u_{i1}|, \quad i=1,2,\dots,m \tag{5.38}$$

Taking into account the orthogonality of vector $\underline{u}_1$ e.g.

$u_{11}^2+u_{21}^2+\dots+u_{m1}^2=1$, the equations (5.36) are true for

$$|u_{11}|=|u_{21}| = \dots = |u_{m1}| = \frac{\sqrt{m}}{m} \tag{5.39}$$

Therefore, the coordinates of vector $\underline{u}_1$ satisfy:

$$u_{i1} = \pm \frac{\sqrt{m}}{m}, \quad i=1,2,\dots,m \tag{5.40}$$

■

Remark (5.5): The result of Proposition (5.5) can also be stated as: For a given normalized matrix $A\in R^{mxn}$ with $\rho(A)=1$, the coordinates of the left

singular vector corresponding to the largest singular value σ_1 have value $\pm\frac{\sqrt{m}}{m}$

■

The numerical version of Proposition (5.5) may be stated as:

Proposition (5.6): Let $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$, $m\leq n$, $\underline{r}_i \neq \underline{0}$, $\rho_\varepsilon(A)=1$ for some ε. Let $A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t \epsilon R^{mxn}$ be the normalization of A with S.V.D. of the form $U\Sigma W^T$, where $U=[\underline{u}_1,\underline{u}_2,\ldots,\underline{u}_m]\epsilon R^{mxm}$, $W=[\underline{w}_1,\underline{w}_2,\ldots,w_n]\epsilon R^{nxn}$ orthogonal matrices and $\Sigma \epsilon R^{mxn}$ with $\Sigma_{ii}=\sigma_i$ and $\Sigma_{ij}=0$ $(i\neq j)$. Then, the first column $\underline{u}_1$ of matrix u is defined by:

$$u_{i1} \approx \pm \frac{\sqrt{m}}{m} \quad , \quad i=1,2,\ldots,m \tag{5.41}$$

■

Example (5.2):

Let $A = \begin{bmatrix} 0.365 & 0.730 & 0.548 & 0.183 \\ -0.049 & -0.098 & -0.073 & -0.024 \end{bmatrix} \epsilon R^{2x4}$ be a given matrix

The normalization of A is :

$$A_N = \begin{bmatrix} 0.365 & 0.730 & 0.548 & 0.183 \\ -0.365 & -0.730 & -0.548 & -0.183 \end{bmatrix} \epsilon R^{2x4}$$

The S.V.D. of A_N is: $A_N=U\Sigma W^T$, with $U=[\underline{u}_1,\underline{u}_2]\epsilon R^{2x2}$,

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \epsilon R^{2x4}, \; W^T \epsilon R^{4x4} \text{ and } \sigma_1 \approx 0.141421 \cdot 10 \approx \sqrt{2},$$

$\sigma_2 \approx 0.158051 \cdot 10^{-11} < \varepsilon = 0.1 \cdot 10^{-10}$, thus $\rho_\varepsilon(A)=1$. The column $\underline{u}_1$ of U is :

$\underline{u}_1=(-0.7071\ 0.7071)^t \approx (-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})^t$.

■

5.5 APPROXIMATION OF MATRICES

Let $A,B\epsilon R^{nxn}$ be given matrices and suppose we wish to determine whether A was produced by a two-sided "rotation" of B; that is, is A=UBV for some orthogonal matrices $U\epsilon R^{mxm}$, $V\epsilon R^{nxn}$? More generally, if we consider all the possible two-sided "rotations" UBV of the given matrix B, how well can we approximate A in the sense of least squares?

We seek to choose orthogonal matrices $U\epsilon R^{mxm}$ and $V\epsilon R^{nxn}$ to minimize $\|A-UBV\|_F$.

It can be proved [Horn, 1] that for $A,B\epsilon R^{mxn}$ and $p=\min\{m,n\}$

$$\min\{\|A-UBV\|_F: U\epsilon R^{mxm} \text{ and } V\epsilon R^{nxn} \text{ orthogonal}\}=\left[\sum_{i=1}^{p}[\sigma_i(A)-\sigma_i(B)]^2\right]^{1/2} \qquad (5.42)$$

In particular, A is a "two-sided rotation" of B if and only if A and B have the same set of singular values.

This result can be applied when it is desired to find a matrix $A_1\epsilon R^{mxn}$ that has $\rho(A_1)=k_1<k$ and most closely approximates a matrix $A\epsilon R^{mxn}$, $\rho(A)=k$ in the Frobenius norm.

Let $A=V\Sigma W^T$ be a singular value decomposition of A. Let Σ_1 be the same as Σ except that only $\sigma_1,\ldots,\sigma_{k_1}$ are used; the remaining $n-k_1$ "diagonal" entries of Σ_1 are zero. Then the matrix $A_1=V\Sigma_1W^T$ has the required property.

In fact, this is true because:

$$\min\{\|A-A_1\|_F\}=\min\{\|A-V\Sigma_1W^T\|_F: V\epsilon R^{mxm} \text{ and } W\epsilon R^{nxn} \text{ orthogonal}\}=$$

$$\left[\sum_{i=1}^{k}[\sigma_i(A)-\sigma_i(\Sigma_1)]^2\right]^{1/2}=\left[\sum_{i=1}^{k}[\sigma_i(\Sigma)-\sigma_i(\Sigma_1)]^2\right]^{1/2}=\sigma_{k_1+1}(\Sigma)+\sigma_{k_1+2}(\Sigma)+\ldots\sigma_k(\Sigma) \qquad (5.43)$$

The value attained in (5.43) is actually the smallest value that relation (5.42) can take in that specific case.

The preceding result can be stated as:

Proposition (5.7): Let $A\epsilon R^{mxn}$, $\rho(A)=k>0$, $A=V\Sigma W^T$ the singular value decomposition of A, $k_1<k$. The matrix $A_1\epsilon R^{mxn}$ with $\rho(A_1)=k_1$ that most closely approximates A in the Frobenius norm is given by $A_1=V\Sigma_1W^T$, when Σ_1 is the same as Σ except that only $\sigma_1,\ldots,\sigma_{k_1}$ are used; the remaining $n-k_1$ "diagonal" entires of Σ_1 are zero.

■

The following question arises immediately from Proposition (5.7).

Is the given approximation "best" only for the Frobenius norm or and for other norms as well?

The following tools are useful for obtaining an answer to the previous question.

Notation (5.1) [Horn, 1]: We recall that a vector norm $\|\cdot\|$ on R^{mxn} is said to be unitarily invariant if

$$\|UAV\| = \|A\|$$

for all $A\epsilon R^{mxn}$ and for all orthogonal matrices $U\epsilon R^{mxm}$, $V\epsilon R^{nxn}$.

If $A\epsilon R^{mxn}$ is a given matrix and if $A=V\Sigma W^T$ is a singular value decomposition of A, then $\|A\|=\|V\Sigma W^T\|=\|\Sigma\|$ for any unitarily invariant norm $\|\cdot\|$. Thus, a unitarily invariant norm of a matrix of a given size depends only on the set of singular values of the matrix.

Two familiar examples of unitarily invariant norms are the Frobenius (Euclidean norm) and the spectral norm. If the singular values of $X=[x_{ij}]\epsilon R^{mxn}$ are $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_p\geq 0$, $p=\min\{m,n\}$, then

$$\|X\|_F = \left(\sum_{j=1}^{n}\sum_{i=1}^{m}|x_{ij}|^2\right)^{1/2} = \left(\sum_{i=1}^{p}\sigma_i^2\right)^{1/2}$$

and

$$\|X\|_2 \equiv \max_{y\neq 0}\frac{\|Xy\|_2}{\|y\|_2} = [\varphi(X^TX)]^{1/2} = \sigma_1 = \max\{\sigma_1,\ldots,\sigma_p\}$$

■

Theorem (5.7) [Horn, 1]: Let $A,B\epsilon R^{mxn}$, be given matrices with singular value decompositions $A=V_1\Sigma(A)W_1^T$ and $B=V_2\Sigma(B)W_2^T$ with orthogonal $V_1,V_2\epsilon R^{mxm}$ and orthogonal $W_1,W_2\epsilon R^{nxn}$ and in which the "diagonal" elements of both $\Sigma(A)$ and $\Sigma(B)$ are arranged in decreasing order. Then $\|A-B\|\geq\|\Sigma(A)-\Sigma(B)\|$ for every unitarily invariant norm $\|\cdot\|$ on R^{mxn}.

■

One consequence of Therorem (5.7) is a generalization of the problem of finding a best (in the sense of least squares) rank k_1 approximation to a given matrix $A\epsilon R^{mxn}$ $\rho(A)=k_1>0$, considered before for the Frobenius norm. If $\|\cdot\|$ is a unitarily invariant norm and if $A_1\epsilon R^{mxn}$ has rank k_1, then

$\sigma_1(A_1)\geq\sigma_2(A_1)\geq\ldots\geq\sigma_{k_1}(A_1)>0=\sigma_{k_1+1}(A_1)=\ldots=\sigma_p(A_1)$, where $p=\min\{m,n\}$. Thus,

$\|A-A_1\| \geq \|\Sigma(A)-\Sigma(A_1)\| =$

$$= \|\mathrm{diag}(\sigma_1(A)-\sigma_1(A_1),\ldots,\sigma_{k_1}(A)-\sigma_{k_1}(A_1),\sigma_{k_1+1}(A_1),\ldots,\sigma_k(A))\| \geq$$

$$\|\mathrm{diag}(0,0,\ldots,\sigma_{k_1+1}(A),\ldots,\sigma_k(A))\|$$

where we have used the fact that a unitarily invariant norm on diagonal matrices is a monotone norm because it is a symmetric gauge function of the diagonal entries. [Horn, 1].
Furthermore, equality is possible for $A_1=V\Sigma_1 W^T$, where $A=V\Sigma W^T$ is a singular value decomposition for A and $\Sigma_1=\mathrm{diag}(\sigma_1(A),\ldots,\sigma_{k_1}(A),0,\ldots,0)$.

Thus for any $A\epsilon R^{mxn}$, $\rho(A)=k$ and any $A_1\epsilon R^{mxn}$ of rank $k_1<k$, we have the bounds

$$\|A-A_1\| \geq \|\mathrm{diag}(0,\ldots,0,\sigma_{k_1+1}(A),\ldots,\sigma_k(A))\| \geq$$

$$\sigma_k(A)\|\mathrm{diag}(0,\ldots 0,1,\ldots,1)\|$$

for any unitarily invariant norm (there are k_1 zero terms on the diagonal of the last expression) in which the first inequality, but not generally the second, is sharp.
The second inequality (which follows solely from monotonicity of symmetric gauge functions if A is nonsingular and is trivial if A is singular) has the advantage that its dependence on the norm is a function of k only and not of A. In particular, this says that for any nonsingular matrix $A\epsilon R^{mxn}$, $\rho(A)=k$ and any unitarily invariant norm $\|\cdot\|$, we have the sharp bound

$$\|A-A_1\| \geq \sigma_k(A)\|\mathrm{diag}(0,\ldots,0,1)\| \qquad (5.44)$$

for the distance between A and any singular matrix A_1; that is, the minimum distance from A to the closed set of singular matrices (with respect to the unitarily invariant norm $\|\cdot\|$) is $\sigma_k(A)\|\mathrm{diag}(0,\ldots,0,1)\|$ and

$$\|X\|_2 \equiv \max_{y\neq 0}\frac{\|Xy\|_2}{\|y\|_2} = [\rho(X^TX)]^{1/2} = \sigma_1 = \max\{\sigma_1,\ldots,\sigma_p\}$$

■

The case of approximating a given matrix $A\epsilon R^{mxn}$ by a rank 1 matrix A_1 occurs frequently enough in the applications that it deserves special mention. Applying Proposition (5.7) for $k_1=1$ we have that the matrix $A_1\epsilon R^{mxn}$ with $\rho(A_1)=k_1$ that most closely approximates A in the Frobenius norm is given by $A_1=V\Sigma_1 W^T=V\cdot\mathrm{diag}(\sigma_1,0,\ldots,0)\cdot W^T=\sigma_1\underline{v}\cdot\underline{w}^t$, where σ_1 is the largest singular value of A, and $\underline{v}$ and $\underline{w}$ are the first columns of the orthogonal matrices V and W in a singular value decomposition of A, respectively. A useful observation about

$\underline{v}$ and $\underline{w}$ is that they are unit vector solutions of the pair of the symmetrical eigenvalue-eigenvector problems

$$AA^T\cdot\underline{v} = \sigma_1^2\cdot\underline{v}, \quad A^TA\cdot\underline{w} = \sigma_1^2\cdot\underline{w}$$

where σ_1^2 is the largest eigenvalue of the positive semidefinite matrix A^TA (and AA^T). This observation does not uniquely determine $\underline{v}$ and $\underline{w}$, of course; one difficulty is that the eigenspaces associated with σ_1^2 need not be one-dimensional. If σ_1^2 is a simple eigenvalue of A^TA (and hence of AA^T), however, the eigenvectors $\underline{v}$ and $\underline{w}$ are determined up to scalar factors of modulus 1 and must therefore be scalar multiples of the respective first columns of the orthogonal matrices V and W in a singular value decomposition $A=V\Sigma W^T$.

When we have to do with strongly ε-dependent sets of vectors the following problem arises:

Given a normalized matrix $A\epsilon R^{mxn}$ with $\rho_\varepsilon(A)\approx 1$ find a "best" rank 1 approximation to A.

According to the previous results, if $A=V\Sigma W^T$ is a singular value decomposition of A, then a "best" rank 1 approximation to A in the Frobenius norm is given by $A_1=\sigma_1\cdot\underline{v}\cdot\underline{w}^t$, where σ_1 is the largest singular value of A that is greater than the given accuracy ε, and $\underline{v}$ and $\underline{w}$ are the first columns of the orthogonal matrices V and W of the singular value decomposition of A, respectively.

Thus, $A_1=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t=\sigma_1\cdot\underline{v}\cdot\underline{w}^t$ =

$$\sigma_1\cdot\begin{bmatrix} v_{11}\\ v_{21}\\ \cdot\\ \cdot\\ \cdot\\ v_{m1}\end{bmatrix}\cdot[w_{11}\ w_{21}\ \ldots\ w_{n1}]=\begin{bmatrix} \sigma_1v_{11}w_{11} & \sigma_1v_{11}w_{21} & \ldots & \sigma_1v_{11}w_{n1}\\ \sigma_1v_{21}w_{11} & \sigma_1v_{21}w_{21} & \ldots & \sigma_1v_{21}w_{n1}\\ \cdot & \cdot & \ldots & \cdot\\ \cdot & \cdot & \ldots & \cdot\\ \cdot & \cdot & \ldots & \cdot\\ \sigma_1v_{m1}w_{11} & \sigma_1v_{m1}w_{21} & \ldots & \sigma_1v_{m1}w_{n1}\end{bmatrix} \qquad (5.45)$$

and taking into account Proposition (5.5) we conclude that $|\underline{r}_1|=|\underline{r}_2|=\ldots=|\underline{r}_m|$, which actually shows that $\rho(A_1)=1$.

All the above results can be applied in order to encounter the following issue:

Problem: Given a set of m unit length vectors $\underline{a}_i \epsilon R^{nx1}$, i=1,2,...,m that for a given tolerance ε are strongly ε-dependent, which is the "best" representative of the set?

The notion of "best" is considered in the following sense. The "best" representative of a set is considered to be a vector that forms equal angles with all the vectors of the set.

The vectors $\underline{a}_i$, i=1,2,...,m form the normalized matrix $A=[\underline{a}_1,\underline{a}_2,...,\underline{a}_m]^t$ with $\rho_\varepsilon(A)\approx 1$. This matrix is approximated bythe rank 1 matrix $A_1=\sigma_1 \cdot \underline{v}_1 \cdot \underline{w}_1^t$, where σ_1 is the largest singular value of A that is greater than ε, and $\underline{v}_1$, $\underline{w}_1$ are the first columns of the orthogonal matrices V and W of the singular value decomposition $V\Sigma W^T$ of A, respectively. From (5.45) we deduce that the rows $\underline{a}_i$, i=1,2,...,m of matrix A are approximated by:

$$\underline{a}_i = \sigma_1 \cdot v_{i1} \cdot \underline{w}_1 \tag{5.46}$$

Generally, we can say that each vector $\underline{a}_i$, i=1,2,...,m of the original set is approximated by (5.46).

Let $\underline{w}_1$ the right singular vector that corresponds to the singular value σ_1. Then,

$$<\underline{w}_1,\underline{a}_i> = \underline{a}_i^t \cdot \underline{w}_1 = \sigma_1 v_{i1} \underline{w}_1^t \cdot \underline{w}_1, \quad i=1,2,...,m \tag{5.47}$$

(5.47) can be written as

$$[\sigma_1 v_{i1} w_{11} \ \sigma_1 v_{i1} w_{21} \ \dots \ \sigma_1 v_{i1} w_{n1}] \cdot \begin{bmatrix} w_{11} \\ w_{21} \\ \cdot \\ \cdot \\ \cdot \\ w_{n1} \end{bmatrix} = \sigma_1 v_{i1} w_{11}^2 + \sigma_1 v_{i1} w_{21}^2 + \dots + \sigma_1 v_{i1} w_{n1}^2 =$$

$$= \sigma_1 v_{i1} (w_{11}^2 + w_{21}^2 + \dots + w_{n1}^2) = \sigma_1 v_{i1}, \quad i=1,2,...,m \tag{5.48}$$

From Proposition (5.5) we conclude that:

$$<\underline{w}_1, \underline{a}_i> = \pm \sigma_1 \frac{\sqrt{m}}{m}, \quad i=1,2,...,m \tag{5.49}$$

Finally the angles θ_i between the vector $\underline{w}_1$ and the vectors $\underline{a}_i$ are given by:

$$\cos\theta_i = \frac{|<\underline{w}_1,\underline{a}_i>|}{\|\underline{w}_1\|_2 \|\underline{a}_i\|_2} = \sigma_1 \frac{\sqrt{m}}{m}, \quad i=1,2,...,m \tag{5.50}$$

From (5.50) it is evident that the vector $\underline{w}_1$ forms equal angles with the approximations of the original vectors thus, it can be considered as the "best" representastive of the given set.

Remark (5.6): For a given strongly ε-dependent set of vectors with corresponding matrix $A=V\Sigma W^T$, where $V\Sigma W^T$ the S.V.D. of A, the "best" representative of the set is the singular vector $\underline{w}_1$ of W corresponding to the singular value σ_1.

■

Example (5.4):

Let $A=\{\underline{a}_1=(0.6571407, -0.7118352, 0.2463886, -0.0273753)^t,$

$\underline{a}_2=(-0.6571219, 0.7118482, -0.2464010, 0.0273773)^t\}$

be a set of normalized vectors. The corresponding matrix A has S.V.D. of the

form $A=V\Sigma W^T$ with $V\epsilon R^{2x2}$, $\Sigma=\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}\epsilon R^{2x4}$, $W = [\underline{w}_1,\underline{w}_2,\underline{w}_3,\underline{w}_4]\epsilon R^{4x4}$

where $\sigma_1 \approx 0.141421\cdot 10 = \sqrt{2}$, $\sigma_2 \approx 0.18446936\cdot 10^{-4} \leq 0.1\cdot 10^{-3}$.
Evidently, for $\varepsilon=0.1\cdot 10^{-3}$ the set is strongly ε-dependent. The best representative of the set is given by:
$\underline{w}_1 = (-0.6571313, 0.7118417, -0.2463948, 0.0273763)^t$.

■

5.6 THE GRAMIAN OF GIVEN VECTORS AND THE SCHUR COMPLEMENT

Two useful tools that can be successfully used when we are dealing with nongeneric computations are the Gramian of given vectors and the Schur complement. Both of them have widespread applications in several areas of matrix theory and a brief presentation of them is given in the sequel.

Definition (5.7) [Gant., 1]: Let $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$ vectors ϵR^n. The matrix

$$G = \begin{bmatrix} (\underline{x}_1\cdot\underline{x}_1)(\underline{x}_1\cdot\underline{x}_2) & \cdots & (\underline{x}_1\cdot\underline{x}_m) \\ (\underline{x}_2\cdot\underline{x}_1)(\underline{x}_2\cdot\underline{x}_2) & \cdots & (\underline{x}_2\cdot\underline{x}_m) \\ \cdots\cdots\cdots\cdots\cdots\cdots & & \\ (\underline{x}_m\cdot\underline{x}_1)(\underline{x}_m\cdot\underline{x}_2) & \cdots & (\underline{x}_m\cdot\underline{x}_m) \end{bmatrix} \epsilon R^{mxm} \qquad (5.51)$$

is called the Gram matrix of the vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$ and the determinant $G_m \equiv G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m)=\det(G)$ is called the Gramian of the vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$.

■

Characteristic Properties of the Gramian and the Gram matrix

(I) One of the most important abilities of Gramian is that it provides us with an important criterion about the linear dependency of vectors.

Theorem (5.8) (Gram's criterion) [Gant., 1]: The vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$ are linearly independent if and only if their Gramian is not equal to zero.

■

The following Corollary is implied from the preceding Theorem.

Corollary (5.4) [Gant., 1]: If any principal minor of the Gramian is zero, then the Gramian is zero.

■

(II) Gram matrices are related with positive definite matrices. In fact, a positive definite Hermitian matrix and a Gram matrix can be made equivalent.

Theorem (5.9) [Horn, 1]: Let $G\epsilon C^{kxk}$ be the Gram matrix of the vectors $\{\underline{w}_1,\underline{w}_2,\ldots,\underline{w}_k\}\subset C^n$ with respect to a given inner product (.,.), and let $W=[\underline{w}_1,\underline{w}_2,\ldots,\underline{w}_k]\epsilon C^{nxk}$. Then

(a) G is positive semidefinite;

(b) G is nonsingular if and only if the vectors $\underline{w}_1,\underline{w}_2,\ldots,\underline{w}_k$ are independent;

(c) There exists a positive definite matrix $A\epsilon C^{nxn}$ such that $G=W^*AW$;

(d) $\rho(G)=\rho(W)$=maximum number of independent vectors in the set $\{\underline{w}_1,\underline{w}_2,\ldots,\underline{w}_k\}$

■

Corollary (5.5) [Horn, 1]: Let $A\epsilon C^{nxn}$ be a given matrix. Then A is positive semi-definite with rank $r\leq n$ if and only if there is a set of vectors $S=\{\underline{w}_1,\underline{w}_2,\ldots,\underline{w}_n\}\subset C^n$ containing exactly r independent vectors such that A is the Gram matrix of S with respect to the Euclidean inner product.

■

Corollary (5.5) proves that a substantial characterization of positive semidefinite matrices is that they are always Gram matrices.

(III) Orthogonal projections of vectors into subspaces can be expressed using the Gramian. [Gant., 1]
Let $\underline{x}$ be an arbitrary vector in a unitary or Euclidean space R and S an m-dimensional subspace with a basis $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$. It can be shown that $\underline{x}$ can be represented (and moreover, represented uniquely) in the form

$$\underline{x} = \underline{x}_S + \underline{x}_N \tag{5.52}$$

where

$$\underline{x}_S \epsilon S \quad \text{and} \quad \underline{x}_N \perp S$$

(orthogonality to a subspace means orthogonality to every vector of the subspace) $\underline{x}_S$ is the orthogonal projection of $\underline{x}$ onto S, $\underline{x}_N$ the projecting vector. $\underline{x}_S$ and $\underline{x}_N$ can be expressed in the terms of the given vector $\underline{x}$, the basis of S and the Gramian $G_m \equiv G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m)$ as follows:

$$\underline{x}_S = -\frac{\begin{vmatrix} & & \underline{x}_1 \\ & G_m & \vdots \\ & & \underline{x}_m \\ (\underline{x}\cdot\underline{x}_1)\ldots(\underline{x}\cdot\underline{x}_m) & & 0 \end{vmatrix}}{G_m} \tag{5.53}$$

$$\underline{x}_N = \underline{x} - \underline{x}_S = \frac{\begin{vmatrix} & & \underline{x}_1 \\ & G_m & \vdots \\ & & \underline{x}_m \\ (\underline{x}\cdot\underline{x}_1)\ldots(\underline{x}\cdot\underline{x}_m) & & \underline{x} \end{vmatrix}}{G_m} \tag{5.54}$$

We draw attention to another important formula. We denote by h the length of the vector $\underline{x}_N$. Then, by (5.52) and (5.54),

$$h^2 = \frac{G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m,\underline{x})}{G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m)} \tag{5.55}$$

The quantity h can also be interpreted in the following way:
Let the vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$, $\underline{x}$ issue from a single point and construct on these vectors as edges an (m+1)-dimensional parallelepiped. Then h is the

height of this parallelepiped measured from the end of the edge $\underline{x}$ to the base S that passes through the edges $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$.

Mostly based on (5.55), it can be proved the next extremely useful Proposition.

Proposition (5.8) [Gant., 1]: The Gramian of linearly independent vectors is positive, that of linearly dependent vectors is zero. Negative Gramians do not exist, e.g. for arbitrary vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$

$$G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m) \geq 0$$

■

(IV) The Geometrical meaning of the Gramian and some inequalities are considered next.
Let $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$ be arbitrary vectors. We call V_m the volume of the m-dimensional parallelepiped spanned by the vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$. Then,

$$\sqrt{G_m} = V_m \tag{5.56}$$

We denote by $x_{1k},x_{2k},\ldots,x_{nk}$ the coordinates of $\underline{x}_k$, k=1,2,...,m in an orthonormal basis of R and set $X=[x_{ik}]$, i=1,2,...,n; k=1,2,...,m. Due to the fact that if x_i, i=1,2,...,n are the coordinates of a vector $\underline{x}$ in an orthonormal basis then

$(\underline{x}\cdot\underline{x})=\sum_{i=1}^{n}|x_i|^2$, the Gramian G_m, can be writen as:

$$G_m = |X^T \bar{X}|$$

and therefore

$$V_m^2=G_m=\sum_{1\leq i_1<i_2<\ldots<i_m\leq n} \mathrm{mod}\begin{vmatrix} x_{i_1 1} & x_{i_1 2} & \cdots & x_{i_1 m} \\ x_{i_2 1} & x_{i_2 2} & \cdots & x_{i_2 m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i_m 1} & x_{i_m 2} & \cdots & x_{i_m m} \end{vmatrix}^2 \tag{5.57}$$

This equation has the following geometric meaning:

The square of the volume of a parallelepiped is equal to the sum of the squares of the volumes of its projections on all the m-dimensional coordinate subspaces.

From relations (5.52) and (5.55) it is easily obtained [Gant., 1] the so-called Hadamard inequality

$$G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m) \leq G(\underline{x}_1)G(\underline{x}_2)\ldots G(\underline{x}_m) \tag{5.58}$$

where the equality sign holds if and only if the given vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$ are pairwise orthogonal. The inequality (5.58) expresses the following fact, which is geometrically obvious:
The volume of a parallelepiped does not exceed the product of the lengths of its edges and is equal to it only when the parallelepiped is rectangular.
As an immediate consequence of Hadamard's inequality, for any given set of unit length vectors $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m$ the following inequality always holds

$$G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m) \leq 1 \tag{5.59}$$

In the sequel, a generalization of Hadamard's inequality is given:

$$G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m) \leq G(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_p)\cdot G(\underline{x}_{p+1},\ldots,\underline{x}_m) \tag{5.60}$$

The inequality (5.60) has the following geometric meaning:
The volume of a parallelepiped does not exceed the product of the volumes of two complementary "faces" and is equal to this product if and only if these faces are orthogonal or at least one of them has volume zero.

(V) The well known inequalities of Schwarz and Bessel can be easily derived using the Gramian. For arbitrary vectors $\underline{x},\underline{y}\in R$ Schwarz's inequality states that

$$|(\underline{x}\cdot\underline{y})|^2 \leq (\underline{x}\cdot\underline{x})\cdot(\underline{y}\cdot\underline{y}) \tag{5.61}$$

The validity of Schwarz's inequality follows simply from the inequality established above

$$G(\underline{x},\underline{y}) = \begin{vmatrix} (\underline{x}\cdot\underline{x}) & (\underline{x}\cdot\underline{y}) \\ (\underline{y}\cdot\underline{x}) & (\underline{y}\cdot\underline{y}) \end{vmatrix} \geq 0$$

For an orthonormal sequence of vectors $\underline{z}_1,\underline{z}_2,\ldots,(z)$ the so-called Bessel inequality is produced in the following way: Let $\underline{x}$ be an arbitrary vector. We denote by ξ_p the projection of $\underline{x}$ onto $\underline{z}_p$:

$$\xi_p = (\underline{x}\cdot\underline{z}_p), \quad p=1,2,\ldots$$

Then the projection of x onto the subspace $s_p=[\underline{z}_1,\underline{z}_2,\ldots,\underline{z}_p]$ can be represented in the form (see (5.53))

$$\underline{x}_{s_p} = \xi_1\underline{z}_1+\xi_2\underline{z}_2+\ldots+\xi_p\underline{z}_p, \quad p=1,2,\ldots$$

But $(\underline{x}_{s_\rho}\cdot\underline{x}_{s_\rho})=|\xi_1|^2+|\xi_2|^2+\ldots+|\xi_p|^2 \leq (\underline{x}\cdot\underline{x})$

Therefore, for every p,

$$|\xi_1|^2+|\xi_2|^2+\ldots+|\xi_p|^2 \leq (\underline{x}\cdot\underline{x}) \tag{5.62}$$

This is Bessel's inequality.

In the case of a space of finite dimension n, this inequality has a completely obvious geometrical meaning. For p=n it goes over into the theorem of Pythagoras

$$|\xi_1|^2+|\xi_2|^2+\ldots+|\xi_y|^2=|x|^2$$

Although most of the recent books establish Schwarz's inequality independently, it can be derived from Bessel's inequality. In [Ever. & Rys., 1] it is proved that Schwarz's inequality is a trivial case of the Bessel inequality.

(VI) Gramian is connected with the singular values of a given set of vectors and satisfies certain inequalities according to the numerical ε-rank of this set.

Let $A=\{\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_n\}$ be a set of n vectors $\underline{a}_i\epsilon R^m$, i=1,2,...,m with corresponding matrix $A=[a_{ij}]\epsilon R^{mxn}$. Let $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_r$, r=min{m,n} be the singular values of matrix A. Then, the Gramian of the vectors $\underline{a}_i$, i=1,2,...,n is given by the next formula:

$$G_n=G(\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_n)=\det(A^T\cdot A)=\prod_{i=1}^{r}\sigma_i^2 \tag{5.63}$$

It is apparent that the next equality is always satisfied.

$$G_n \geq r\cdot\sigma_1^2 \tag{5.64}$$

In virtue of (5.63) and from Definition (5.3) and Remark (5.3), the following bounds for the Gramian of dependent sets of vectors can be derived.

(i) If the set A is ε-independent with $\rho_\varepsilon(A)=r$, then

$$G_n > r\cdot\varepsilon^2 \tag{5.65}$$

(ii) If the set A is numerically ε-dependent with $\rho_\varepsilon(A)=g<r$ then,

$$G_n \leq g\cdot\sigma_1^2(n-g)\varepsilon^2 \tag{5.66}$$

(iii) If the set A is strongly ε-dependent with $\rho_\varepsilon(A)=1$ then,

$$G_n \leq (n-1)\varepsilon^2\cdot\sigma_1^2 \tag{5.67}$$

Due to Theorem (5.5), if $\|a_i\|_2=1$ relation (5.67) can be written as:

$$G_n \leq (n-1)\sqrt{m}\ \varepsilon^2 \tag{5.68}$$

From the above, it is evident that the Gram matrix of given vectors and their Gramian, have widespread applications and it can be used in many cases. More properties of the Gramian and two methods for expanding it as well, one based on Pythagorean equality and another based on Gram-Schmidt orthonormalizing process of elements in a Hilbert space, can be found in [Wong., 1].

(b) The Schur complement

Definition (5.8) [Carl., 1]: Let

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in F^{mxn} \tag{5.69}$$

where F is any arbitrary field, with $A\in F^{kxk}$ and nonsingular. The classical Schur complement of A in M is the matrix $S\in F^{(m\times k)\times(n\times k)}$ given by the formula

$$S = D-CA^{-1}B \tag{5.70}$$

and is denoted by (M/A).

■

The idea of the Schur complement matrix goes back to Sylvester (1851). It is well known that the entry s_{ij} of S, i=1,...,m-k, j=1,...,n-k, is the minor of M determined by rows 1,...,k,k+i and columns 1,...,k,k+j, a property which was used by Sylvester as his definition.
When $M\in C^{nxn}$ is Hermitian, then $C=B^*$ and

$$S = D - B^{*}A^{-1}B \tag{5.71}$$

is also Hermitian.

The name Schur is suggested by the well known determinantal formula (for the case where M is square) [Cotl., 1]

$$\det M = \det A \cdot \det(D - CA^{-1}B) \tag{5.72}$$

Matrices of the form $D-CA^{-1}B$ are very common; perhaps their most frequently encountered manifestation is in ordinary or "generalized" Gaussian elimination.

(b.1) Classical Results [Carl., 1]

In this section we derive some of the classical uses of the Schur complement formula (5.70). First, applying Gaussian elimination (without pivoting) successively to the first k rows of the matrix (5.69) (with A nonsingular) yields the matrix

$$\begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & S=D-CA^{-1}B \end{bmatrix} \tag{5.73}$$

Thus, the Schur complement arises naturally in discussions of Gaussian elimination, and in mathematical programming. Applying column operations to (5.73) to eliminate B yields

$$\begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \tag{5.74}$$

from which several facts are clear:

(i) $\rho(M)=\rho(A)+\rho(S)$

(ii) If M is nxn, then $\det M=\det A \cdot \det S$

(iii) If M is nxn and nonsingular, then S is also nonsingular

(iv) If M is nxn and nonsingular, then using (5.74) to write M as a product of three matrices, and inverting, we obtain the formula of Banachiewicz for M^{-1}:

$$M^{-1} = \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} =$$

$$\begin{bmatrix} A^{-1}+A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix} \tag{5.75}$$

(v) If F=C and M is Hermitian, then InM=InA+InS, where InM is the inertia of M.
(vi) If F=C and M is Hermitian, then M is nonnegative definite if and only if A and S are.

All the above results were including evaluations with partitioned matrices. In such cases the following theorem always hold.

Theorem (5.10) [Gant., 1]: If to the a-th row (column) of the blocks of the partitioned matrix A we add the β-th row (column) multiplied on the left (right) by a rectangular matrix X of the corresponding dimensions, then the rank of A remains unchanged under this transformation and, if A is a square matrix, the determinant of A is also unchanged.

■

From Theorem (5.10) there follows also another classical result concerning the Schur complement.

Theorem (5.11) [Gant., 1]: If a rectangular matrix R is represented in partitioned form

$$R = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{5.76}$$

where A is a square nonsingular matrix of order n ($|A| \neq 0$), then the rank of R is equal to n if and only if

$$D = CA^{-1}B \tag{5.77}$$

■

From Theorem (5.11) there follows an algorithm for the construction of the inverse matrix A^{-1} [Gant., 1] and, more generally, the product $CA^{-1}B$, where B and C are rectangular matrices of dimensions nxp and qxn.

(b.2) The quotient property

A nice property of the Schur complement called the quotient property is concerned next. [Cotl., 1]

$$\text{If } M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ and } A = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

where A and E are nonsingular, then

$$(M/A) = [(M/E)/(A/E)] \tag{5.78}$$

The nonsingularity of (A/E) follows from that of A and E via Schur's determinantal formula (5.72). Moreover, it turns out that (A/E) is the leading block of (M/E) so the grand Schur complement on the right-hand side of equation (5.78) is well-defined. One implication of the quotient property is that under the given hypotheses on A and E, the calculation of the Schur complement of A in M can be carried out in two stages. Also, the quotient property is a very useful tool in providing or verifying a special kind of formulas [Temp., 1]:

Given a matrix A and an algebraic relation between entries which all can be viewed as Schur-complements, each of a submatrix of A with respect to another submatrix of A, choose as matrix E the "greatest common submatrix" of all the submatrices mentioned above.

By the quotient property, each coefficient of the formula may now be interpreted as a Schur-complement of a submatrix of A/E with respect to another submatrix of A/E. Since the dimension of A/E is smaller than of A, the representation of the coefficients has become simpler, so one may verify the formula to be proved by elementary calculations. Tempelmeier has used this method to simplify and unify the proofs of a number of important extrapolation algorithms. In [Temp., 1] he proves the cross-rule for Wynn's ε-Algorithm applying the Schur complement method.

(b.3) Usage of the Schur complement in a Determinantal test [Cotl., 1]

In some circumstances, it is desirable to know whether the leading principal minors of a square matrix $M=[m_{ij}]$ are all nonzero (or, of a particular sign, say positive). This can be determined by pivoting and use of the quotient formula.

Let M be of order n and denote by M[1,...,k] its leading principal submatrix of order k where k=1,...,n:

$$M[1,\ldots,k] = \begin{bmatrix} m_{11} & \cdots & m_{1k} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ m_{k1} & \cdots & m_{kk} \end{bmatrix}$$

Now suppose m_{11} is nonzero (positive). Using m_{11} as the pivot leads to the Schur complement

$$M^{(1)} = (M/m_{11}),$$

in which the leading entry is

$$m_{11}^{(1)} = m_{22} - \frac{m_{12}m_{21}}{m_{11}} = (M[1,2]/M[1])$$

Moreover,

$$m_{11}^{(1)} = \det m_{11}^{(1)} = \frac{\det M[1,2]}{\det M[1]}$$

In general, if the procedure is not interrupted by the discovery of a leading entry (i.e. leading principal minor) of zero (nonpositive) value, then after $k<n$ steps

$$m_{11}^{(k)} = (M[1,\ldots,k+1]/M[1,\ldots,k])$$

and

$$m_{11}^{(k)} = \det m_{11}^{(k)} = \frac{\det M[1,\ldots,k+1]}{\det M[1,\ldots,k]}$$

It should be emphasized here that at each stage, the entries of the new Schur complement are easily computed from the matrix currently at hand. This is done by pivoting just as in the case of Gaussian elimination. With $M=M^{(0)}$, the individual entries of $M^{(k)}$ are given by the formula

$$m_{ij}^{(k)} = m_{i+1,j+1}^{(k-1)} - \frac{m_{i+1,1}^{(k-1)}\, m_{1,j+1}^{(k-1)}}{m_{11}^{(k-1)}}$$

The procedure above has an obvious application to the well known determinantal test for positive definiteness.

More applications of Schur complement in computing inertias of matrices, covariance matrices of conditional distributions can be found in [Cotl., 1].

Also applications to eigenvalue problems are given in [Hayn., 1]. Interesting results concerning generalized inverses, solutions of optimal rank problems, shorted operators are cited in [Carl., 1].

5.7 "BEST UNCORRUPTED" BASES OF SETS OF VECTORS

5.7.1 Introduction

The already known methods for finding bases, orthogonal or not, for given sets of vectors are based on the fact that they virtually transform the original data by using mostly Gaussian or orthogonal techniques. Evidently, they obtain new sets and amongst the new vectors they choose the required ones that span the original set. Thus, the base will be consisted from vectors completely different from the given ones. The following problem is considered next: For a given set of vectors we would like to choose a "best uncorrupted" base, the "best" in a sense to be made precise later. By the term "uncorrupted" we mean that we want to find a base for this set without transforming the original data and evidently introducing roundoff-error even before the method starts. Of course, especially when orthogonal techniques are used for the transformation of the given data the roundoff-error is not actually remarkable. But when we are dealing with nongeneric computations and the evaluation of a base out of a given set of vectors is required, it is important to select this base from the original set of data for the following reasons:

1) If we are given for example a numerically ε-dependent set of vectors by using orthogonal techniques the original set will be transformed and the new set might not have the initial property any more.

2) When we are interested in evaluating the g.c.d. of a given set, it is extremely important to begin the calculating process using the concrete set of data or a subset of it.

3) The singular values of the original set will be altered when orthogonal techniques are used, therefore the computations will start with a different set of singular values.

Let $A=\{\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m\}$ be a set of m given vectors $\underline{r}_i\epsilon R^n$, i=1,2,...,m. This set can be expressed in terms of a matrix $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t\epsilon R^{mxn}$. Then, the problem of finding a "best uncorrupted" base for the set A is transferred into finding a "best uncorrupted" base for the row space of matrix A.

For the evaluation of an uncorrupted base without the restriction of being in a sense "best", the row-searching algorithm [Chen, 1] can be used. A better stable algorithm that applies Gaussian elimination with partial

pivoting or Householder transformations to the columns of the given matrix can also be found in [Wilk., 2], [Chen, 1].

In the sequel, we develop a method for the selection of a "best uncorrupted" base of a given set of vectors.

5.7.2 A method of selecting a "best uncorrupted" base for the row space of a matrix

The following problem is encountered.

For a given matrix $A\epsilon R^{mxn}$ with $\rho(A)=r\leq\min\{m,n\}$ we want to find a "best" ,in some sense, uncorrupted base for its row space.

Combining the preceding theory and the theory developed in Chapter 4 about the compound matrices, we can easily prove the following important Proposition, which gives us an effective algorithm for the selection of an uncorrupted base for the row space of a matrix.

Proposition (5.9): Let $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t\epsilon R^{mxn}$, $\rho(A)=r\leq\min\{m,n\}$, $A_N=[\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t\epsilon R^{mxn}$ the normalization of A. Suppose $G\epsilon R^{mxm}$ the Gram matrix of the vectors $\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m$ and $C_r(G)=[c_{ij}]\epsilon R^{\binom{m}{r}x\binom{m}{r}}$ the r-th compound matrix of G. If $c_{ii}=\det(G[a/a])$, $a=(i_1,i_2,\ldots,i_r)\epsilon Q_{r,m}$ is the maximum diagonal element of $C_r(G)$, then a most orthogonal uncorrupted base for the row space of A, consisits from the vectors:

$$\{\underline{r}_{i_1},\underline{r}_{i_2},\ldots,\underline{r}_{i_r}\}.$$

■

The proof of the above result readily follows from the relationship between $C_r(G)$ and the Gramian.

Proposition (5.9) provides us an uncorrupted base for the row space of a given matrix and if we regard the rows of this matrix as given vectors we obtain an uncorrupted base for this set of vectors. Due to the constructive process, this base contains vectors that are mostly orthogonal, therefore if we define the notion of a "best base" as a base consisting of vectors that are mostly orthogonal, the previous defined base satisfies this definition.

Remark (5.7): The most orthogonal uncorrupted base may not be uniquely defined. In the following, any such base will be referred to, as a best uncorrupted base.

■

Example (5.4): Let $A=\{\underline{a}_1=(3,1,0)^t,\ \underline{a}_2=(-3,2,1)^t,\ \underline{a}_3=(6,5,1)^t\}$ be a given set of vectors. We want to find a best uncorrupted basis for this set.

The corresponding matrix to the given set A is:

$$A=[\underline{a}_1,\underline{a}_2,\underline{a}_3]^t = \begin{bmatrix} 3 & 1 & 0 \\ -3 & 2 & 1 \\ 6 & 5 & 1 \end{bmatrix} \in R^{3x3},\ \rho(A)=2\leq 3$$

The normalization of A is:

$$A_N = [\underline{v}_1,\underline{v}_2,\underline{v}_3]^t = \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} & 0 \\ -3/\sqrt{14} & 2/\sqrt{14} & 1/\sqrt{14} \\ 6/\sqrt{62} & 5/\sqrt{62} & 1/\sqrt{62} \end{bmatrix}$$

The Gram matrix of vectors $\underline{v}_1,\underline{v}_2,\underline{v}_3$ is:

$$G = A_N \cdot A_N^T = \begin{bmatrix} 1 & -7/(\sqrt{10}\sqrt{14}) & 23/(\sqrt{10}\sqrt{62}) \\ -7/(\sqrt{10}\sqrt{14}) & 1 & -7/(\sqrt{14}\sqrt{62}) \\ 23/(\sqrt{10}\sqrt{62}) & -7/(\sqrt{14}\sqrt{62}) & 1 \end{bmatrix}$$

In the sequel, we evaluate $C_2(G)\in R^{\binom{3}{2}x\binom{3}{2}}$

$$C = C_2(G) \approx \begin{bmatrix} 0.65 & 0.30914 & -0.78 \\ 0.30914 & 0.15 & -0.37 \\ -0.78 & -0.37 & 0.94 \end{bmatrix}$$

The maximum diagonal element of $C_2(G)$ is c_{33} and is given by:

$$c_{33} = \det(G[a/a),\quad a = (2,3)\in Q_{2,3}$$

Therefore, a best uncorrupted base for the set A is consisted from the vectors $\underline{a}_2,\underline{a}_3$ and analytically is:

$$B_u = \{\underline{b}_1=(-3,2,1)^t,\ \underline{b}_2 = (6,5,1)^t\}.$$

■

5.7.3 The numerical algorithm of the method and its analysis

a. The numerical algorithm

Given $A=[\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_m]^t \epsilon R^{mxn}$ and ε a specified tolerance, the following algorithm overwrites A by the row independent matrix.

Algorithm UNCBAS

STEP 1: A_N := the normalization of A

$A_N := [\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m]^t \epsilon R^{mxn}$

$\rho := \rho_\varepsilon(A_N)$

G := the Gram matrix of the vectors $\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_m$

$$numb := \frac{m!}{\rho!(m-\rho)!}$$

STEP 2: **for** i = 1,...,numb

construct all the sequences

$r_i = (r_{i_1},r_{i_2},\ldots,r_{i_p}) \epsilon Q_{p,m}$

STEP 3: **for** i = 1,...,numb

$$c_i := g_{r_i}^{r_i}$$

$ind_i := i$

STEP 4 : Reorder the pairs (c_i,ind_i), i=1,2,...,numb such as: $c_1 \geq c_2 \geq \ldots \geq c_{numb}$

STEP 5: The row independent matrix is given by

for i = 1,...,ρ

for j = 1,...,n

$a_{ij} := a_{r_{ind_{1,i}},j}$

Alg: 5.1

Algorithm **UNCBAS** is consisted from some elementary algorithms that are developed seperately.

a.1 Algorithm NORMAL

Given $A \epsilon R^{mxn}$, the following algorithm computes the normalization A_N of A. Matrix A is overwritten by A_N.

```
for i = 1,...,m
        n
    Σ := Σ  a_ij^2
       j=1
    for j = 1,...,n
        a_ij := a_ij/√Σ
```

Alg: 5.2

Computational complexity: 2mn flops

Error Analysis of algorithm NORMAL

Let $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_1 \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \dots & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \epsilon R^{mxn}$ be a given matrix

and let $A_N \epsilon R^{mxn}$ be its normalized form. Using floating point arithmetic and according to backward error analysis we shall prove that:

(i) $fl(A_N)=A_N+E_N$, where E_N a matrix accounting for the roundoff error.
(ii) Matrix E_N is suitably bounded.

Let us develop the above assumptions more analytically:

(i) Each element v_{ij} of A_N is equal to $\dfrac{a_{ij}}{\left(\sum_{j=1}^{n} a_{ij}^2\right)^{\frac{1}{2}}}$. Thus we must

analyse the floating point evaluation $fl\left(\frac{a_{ij}}{\left(\sum_{j=1}^{n} a_{ij}^2\right)^{\frac{1}{2}}}\right)$. This computation is performed executing the following steps.

STEP 1 : Evaluate $a = fl_2\left(\sum_{j=1}^{n} a_{ij}^2\right)$

STEP 2 : Evaluate $b = fl(a^{1/2})$

STEP 3 : Evaluate $c = fl\left(\frac{a_{ij}}{b}\right)$, $i=1,2,\ldots,m$, $j=1,2,\ldots,n$.

Following an analysis cited in [Wilk., 1] we carry out the succeeding evaluations.

STEP 1 : $a = fl_2(a_{i1}^2+a_{i2}^2+\ldots+a_{in}^2) \equiv a_{i1}^2(1+\varepsilon)+\ldots+a_{in}^2(1+\varepsilon)$,

$|\varepsilon|<n\cdot u_3$, where $u_3=\left(1+\frac{1}{\beta}\right)\frac{1}{2}\cdot\beta^{1-2t_2}$, $2t_2=2t-0.08406$,

β is the machine base, t is the number of digits of machine word.
It can be easily proved that:

$$a = fl_2(a_{i1}^2+a_{i2}^2+\ldots+a_{in}^2) \equiv (a_{i1}^2+\ldots+a_{in}^2)(1+\varepsilon), \quad |\varepsilon|<1.00001\cdot u \tag{5.79}$$

where $u = \begin{cases} \frac{1}{2}\beta^{1-t} & \text{if rounded arithmetic is used} \\ \beta^{1-t} & \text{if chopped arithmetic is used} \end{cases}$ the unit roundoff error

It is important the above computation $\sum_{j=1}^{n} a_{ij}^2$ to be accumulated with a 2t-digit mantissa and thus we used $fl_2()$ computation for its evaluation.

STEP 2 : $b = fl(a^{1/2}) \equiv a^{1/2}(1+n) = \sqrt{a_{i1}^2+\ldots+a_{in}^2}\ \sqrt{1+\varepsilon}\ (1+n)$,

$$|n| < 1.00001\cdot u \tag{5.80}$$

We set $\sqrt{1+\varepsilon}\ (1+n)=1+\xi$ and finally we have

$$b = fl(a^{1/2}) = a^{1/2}(1+\xi), \quad |\xi| < 2.00002 \cdot u \tag{5.81}$$

STEP 3 : $c = fl(\frac{a_{ij}}{b}) \equiv \frac{a_{ij}}{a^{1/2}(1+\xi)}(1+k), \quad |k| < u$

We set $1+\theta_{ij} = \frac{1+k}{1+\xi}$ and from (5.80), (5.81) we derive that $|\varepsilon_{ij}| \leq 3.003 \cdot u$

$$c = fl(\frac{a_{ij}}{b}) = \frac{a_{ij}}{b}(1+\theta_{ij}), \quad |\theta_{ij}| \leq 3.003 \cdot u \tag{5.82}$$

Combining (5.80), (5.81), (5.82) we remark that

$$fl(v_{ij}) = v_{ij}(1+\theta_{ij}), \quad |\theta_{ij}| \leq 3.003 \cdot u$$

Therefore it is evident that $fl(A_N) = A_N + E_N$, where

$$E_N = \begin{bmatrix} v_{11}\theta_{11} & v_{12}\theta_{12} & \cdots & v_{1n}\theta_1 \\ v_{21}\theta_{21} & v_{22}\theta_{22} & \cdots & v_{2n}\theta_{2n} \\ . & . & \cdots & . \\ v_{m1}\theta_{m1} & v_{m2}\theta_{m2} & \cdots & v_{mn}\theta_{mn} \end{bmatrix} \in R^{mxn} \text{ is the error matrix}$$

(ii) In order to bound matrix E_N we set $\varepsilon_{ij} = v_{ij}\theta_{ij}$.
We remark that each $v_{ij} \leq 1$, thus

$$|\varepsilon_{ij}| = |v_{ij}\theta_{ij}| \leq 3.003 \cdot u$$

Consequently,

$$|E_N| \leq 3.003 \cdot u \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ . & . & \cdots & . \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{5.83}$$

If the $\| \ \|_\infty$ is used, $\|E_N\|_\infty \leq 3.003n \cdot u$.

a.2 Algorithm RANKMA

Given $A \epsilon R^{mxn}$, $r=\min\{m,n\}$, σ_i, $i=1,2,\ldots,r$ $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$ the singular values of A, ε a tolerance, the following algorithm computes the numerical ε-rank, $\rho_\varepsilon(A)$ of matrix A.

```
rank := 0
for i = 1,...,r
    if σ_i > ε then
        rank := rank+1
```

Alg: 5.3

a.3 Algorithm GRAM

Given $A=[\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_m]^t \epsilon R^{mxn}$, the following algorithm evaluates the Gram matrix $G \epsilon R^{mxm}$ of vectors $\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_m$.

```
for i = 1,...,m
    for j = i,...,m
        g_ij := a_i · a_j
        g_ji := g_ij
```

Alg: 5.4

Computational complexity: $\int_1^m (m-i)n \, d_i = \frac{m^2n}{2} - mn + \frac{n}{2} \approx O(\frac{m^2n}{2})$ flops

Error Analysis of algorithm GRAM

Let $\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_m$ be a given set of vectors with $\underline{a}_i \epsilon R^n$ and let $G=(g_{ij}) \epsilon R^{mxm}$ be their Gram matrix. Using floating point arithmetic and according to backward error analysis we shall prove that:

(i) fl(G)=G+E, where E a matrix accounting for the roundoff error.
(ii) Matrix E is suitably bounded.

Let us discuss the above assumptions more specifically.

(i) From Definition (5.7) $g_{ij} = \underline{a}_i^t \cdot \underline{a}_j$, i,j=1,2,...,m. Thus, $fl(g_{ij}) = fl(\underline{a}_i^t \cdot \underline{a}_j)$
Since each g_{ij} is actually an inner product, from [Wilk., 1] we conclude that

$$fl(\underline{a}_i^t \cdot \underline{a}_j) = \underline{a}_i^t \cdot \underline{a}_j + \varepsilon_{ij} \text{ , where} \tag{5.84}$$

$$|\varepsilon_{ij}| \leq u_1 \{(n+1)|a_{i1}| \cdot |a_{j1}| + n|a_{i2}| \cdot |a_{j2}| + \ldots + 2|a_{in}| \cdot |a_{jn}|\}$$

$$u_1 = \begin{cases} \frac{1}{2}\beta^{1-\tau} & \text{if rounded arithmetic is used} \\ \beta^{1-\tau} & \text{if chopped arithmetic is used} \end{cases}, \quad \tau = t - \log_\beta(1.01),$$

β is the machine base and t is the number of digits of machine word.

Therefore $fl(G)=G+E$, $E=(\varepsilon_{ij}) \in R^{mxm}$

(ii) From (5.84) we remark that

$$|E| \leq (n+1) \cdot u_1 \cdot \begin{bmatrix} |\underline{a}_1|^t \cdot |\underline{a}_1| & |\underline{a}_1|^t \cdot |\underline{a}_2| & \ldots & |\underline{a}_1|^t \cdot |\underline{a}_m| \\ |\underline{a}_2|^t \cdot |\underline{a}_2| & |\underline{a}_2|^t \cdot |\underline{a}_2| & \ldots & |\underline{a}_2|^t \cdot |\underline{a}_m| \\ \vdots & \vdots & \ldots & \vdots \\ |\underline{a}_m|^t \cdot |\underline{a}_1| & |\underline{a}_m|^t \cdot |\underline{a}_2| & \ldots & |\underline{a}_m|^t \cdot |\underline{a}_m| \end{bmatrix} \tag{5.85}$$

The relative error is given by the expression

$Rel = \dfrac{|G - fl(G)|}{|G|}$. Taking into account (5.85), we have

$$\mathrm{Rel} \leq (n+1)\cdot u_1 \cdot \frac{\begin{bmatrix} |\underline{a}_1|^t\cdot|\underline{a}_1| & |\underline{a}_1|^t\cdot|\underline{a}_2| & \cdots & |\underline{a}_1|^t\cdot|\underline{a}_m| \\ |\underline{a}_2|^t\cdot|\underline{a}_1| & |\underline{a}_2|^t\cdot|\underline{a}_2| & \cdots & |\underline{a}_2|^t\cdot|\underline{a}_m| \\ \vdots & \vdots & \cdots & \vdots \\ |\underline{a}_m|^t\cdot|\underline{a}_1| & |\underline{a}_m|^t\cdot|\underline{a}_2| & \cdots & |\underline{a}_m|^t\cdot|\underline{a}_m| \end{bmatrix}}{\begin{bmatrix} |\underline{a}_1^t\cdot\underline{a}_1| & |\underline{a}_1^t\cdot\underline{a}_2| & \cdots & |\underline{a}_1^t\cdot\underline{a}_m| \\ |\underline{a}_2^t\cdot\underline{a}_1| & |\underline{a}_2^t\cdot\underline{a}_2| & \cdots & |\underline{a}_2^t\cdot\underline{a}_m| \\ \vdots & \vdots & \cdots & \vdots \\ |\underline{a}_m^t\cdot\underline{a}_1| & |\underline{a}_m^t\cdot\underline{a}_2| & \cdots & |\underline{a}_m^t\cdot\underline{a}_m| \end{bmatrix}} \qquad (5.86)$$

If $|\underline{a}_i^t\cdot\underline{a}_j| << |\underline{a}_i|^t|\underline{a}_j|$ for i,j=1,2,...,m then the relative error in fl(G) may not be small and evidently algorithm **GRAM** may not be stable.

b. <u>Implementation of the algorithm</u>

For the implementation of algorithm **UNCBAS**, some subroutines performing basic tasks are required. Such subroutines can be found in various libraries and can be taken from them.

For the evaluation of the singular values of a matrix $A\epsilon R^{mxm}$ we use subroutine F02WCF of NAG Library. The computational complexity of this subroutine is approximately proportional to:

$8n^2(m+n/2)$ flops, $m\geq n$

$10m^2(n+m/2)$ flops, $m<n$

For the evaluation of the determinant of a given matrix $A\epsilon R^{mxm}$, subroutine F03AAF of NAG Library is used. The computational complexity of F03AAF is proportional to $O(m^3)$ flops.

The selection of all the sequences $r_i\epsilon Q_{p,m}$, $i=1,2,\ldots,\binom{m}{p}$ is achieved by using algorithm **CONSEQ** that was developed in Chapter 4 and is accomplished in $O\{\binom{m}{p}\}$ flops.

In the sequel, we discuss few things about the space complexity of **UNCBAS**. Basically, algorithm **UNCBAS** needs one mxn array for storing the original matrix and two mxm arrays for the Gram matrix and the determinantal evaluations. Moreover, it requires an $\binom{m}{p}$xp array for keeping in memory all the sequences of $Q_{p,m}$. Also, it uses an $\binom{m}{p}$ array for storing the diagonal elements of the constructed compound matrix. These extra requirements can be avoided if immediately after the evaluation of each sequence r_i of $Q_{p,m}$, $i=1,2,\ldots,\binom{m}{p}$, we compute the corresponding element $g_{r_i}^{r_i}$ of the compound matrix. We compare this element with the previous one and keep in memory only the largest one. Therefore, except the basic memory requirements, we need only one more array of dimension p.

Algorithm **UNCBAS** was programmed [Mit. & Kar., 1] and tested for several cases. Some numerical examples illustrating its application are presented in Appendix B.

5.7.4 Further Comments

In some cases it is required to choose an uncorrupted base out of a given set of vectors that will satisfy some concrete initial restrictions. Therefore, sometimes we are more interested in finding an uncorrupted base fulfilling certain conditions no matters whether it is "best" or not. Next we develop such a case.

Suppose we are given a set of vectors $\{\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m\}$ with corresponding matrix $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$, $\rho(A)=q\leq\min\{m,n\}$ and we want to find the row independent matrix under the restriction that between all the possible combinations of linearly independent rows, the one with the smallest row indices is preferred. e.g. For m=6, n=8, q=4, between the possible sets (r_1,r_3,r_5,r_7), (r_1,r_5,r_7,r_8) the set (r_1,r_3,r_5,r_7) is preferred. This restriction can be applied in the case of evaluating the g.c.d. of several polynomials (Chapter 7). Before starting the evaluation process, sometimes it is required to select a base amongst them. Then, taking as vectors the polynomials ordered in increasing order and as matrix A their coefficient matrix, we want to find the row independent matrix A under the previous condition. The base selected in that way contains as more polynomials as

possible of lowest degree and evidently the computational process will be more simplified.

Algorithm RINDMA

Let $A=[\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_m]^t \epsilon R^{mxn}$ be a given matrix, $\rho(A)=q\leq\min\{m,u\}$, ε a given tolerance. The following algorithm finds the row independent matrix $A=[\underline{r}_{1_1},\underline{r}_{i_2},\ldots,\underline{r}_{i_q}]^t \epsilon R^{qxn}$ under the restriction that $\{i_1,i_2,\ldots,i_q\}$ is the smallest possible combination of row indices. The variables rind and rdep count the number of independent and dependent rows of matrix A, respectively.

```
          rind := 0
          rdep := 0
          for i = 1,...,m-1

STEP 1:   Compute the Gram matrix G
          of vectors r1,r2,...,ri+1
          if G(r1,r2,...,ri+1)>ε then
             rind := rind+1
             if rind = q then
                quit
           else
             rdep := rdep+1
             throw row ri+1 from A
             A: = [r1,r2,...,rm-rdep]^t
             repeat STEP 1
```

Alg: 5.5

5.8 CONCLUSIONS

The aim of this Chapter was to provide efficient numerical techniques dealing with issues in nongeneric computations. The new notions of ε-independent, numerically ε-dependent, strongly ε-dependent, fuzzy ε-dependent

sets of vectors were defined and a necessary and sufficient condition relating the numerical ε-rank of a strongly ε-dependent set and its singular values was proved. The substantial problems of choosing a "best" representative of a strongly ε-dependent set and of selecting a "best uncorrupted" base for the row space of a matrix were also considered. Therefore, this Chapter serves the following purposes:

(i) It provides a detailed survey concerning the most characteristic properties of the Gramian of given vectors and the Schur complement.

(ii) It provides efficient numerical tools for handling nongeneric computations.

(iii) It provides a stable algorithm for selecting a "best uncorrupted" base for the row space of a matrix.

CHAPTER 6

SURVEY OF METHODS FOR FINDING THE GREATEST COMMON DIVISOR OF POLYNOMIALS

6.1 INTRODUCTION

The problem of finding the greatest common divisor (g.c.d.) of two or more members of a Euclidean ring has interested mathematicians for a very long time, and has widespread applications. In the theory of linear multivariable control systems, when the elements are polynomials, g.c.d. determination arises in the computation of the Smith form of a polynomial matrix and associated problems such as minimal realization of a transfer function matrix [Kal., Fal. & Arb., 1], [Pac. & Bar., 1]. There are also applications in network theory [Fry., 1]. When the ring is the set of integers, construction of g.c.d. occurs in solution of systems of linear diophantine equations and integer linear programming [Pac. & Bar., 1].

Since the existence of a common divisor of polynomials is a nongeneric property, small errors in its computation can lead to incorrect results. Therefore, extra care is needed in order to develop efficient algorithms calculating correctly the required g.c.d.

The algorithm associated with Euclid is the oldest known solution to the greatest common divisor problem. This algorithm computes the positive greatest common divisor of two given positive integers. However, it is readily generalized to apply to any other Euclidean ring and thus to polynomials in any number of variables over any unique factorization domain in which greatest common divisors can be computed.

In nineteenth century the developing apparatus of determinants was applied to the greatest common divisor problem by J.J. Sylvester. Other methods that came out later on, were either a variation of Euclid's algorithm or they used Sylvester's Resultant on different guises.

The purpose of this Chapter is to give a comprehensive survey of algorithms for computing the g.c.d. of many polynomials. The algorithms presented here are divided into two categories.

The first category consists of algorithms which are based on Euclid's algorithm. Examples of these include Collin's algorithm, generalized Euclid's algorithm, and Routh's algorithm. The algorithms in the above category are also extended to unique factorization domain.

In the second category, some well known matrix methods for computing the g.c.d. are presented. These include methods due to Blankiship, Sylvester, and Barnett.

In both categories, numerous examples are presented to aid the underlying theories in the g.c.d. calculations.

6.2 EUCLID AND RELATED ALGORITHMS FOR COMPUTING THE G.C.D. OF POLYNOMIALS [Knu., 1]

Let F field and G[s] the ring of polynomials over F. Let a(s), b(s) be two given polynomials ϵF[s] with $b(s)\neq 0$. Then, it is well known that there exists unique polynomials q(s), r(s)ϵF[s] such that $a(s)=q(s)b(s)+r(s)$ with either $r(s)=0$ or $0<\deg\{r(s)\}<\deg\{q(s)\}$.

The following algorithm may be used to determine q(s) and r(s)

Algorithm DIV (Division of polynomials over a field)

Given polynomials

$$a(s)=a_m s^m+a_{m-1}s^{m-1}+\ldots+a_1 s+a_0,\quad b(s)=b_n s^n+b_{n-1}s^{n-1}+\ldots+b_1 s+b_0$$

over a field F, where $b_n\neq 0$ and $m\geq n\geq 0$, this algorithm finds the polynomials

$$q(s) = q_{m-n}s^{m-n}+\ldots+q_0,\quad r(s) = r_{n-1}s^{n-1}+\ldots+r_0 \qquad (6.1)$$

over F with either $r(s)=0$ or $0<\deg\{r(s)\}<\deg\{q(s)\}$.

```
for k = m-n, m-n-1,...,0
    q_k := a_{n+k}/b_n
    for j = n+k-1, n+k-2,...,k
        a_j := a_j - q_k b_{j-k}
```

Alg: 6.1

When the algorithm is terminated $a_{n-1}=r_{n-1},\ldots,a_0=r_0$. The computational complexity of this algorithm is proportional to n(m-n+1). For some reason this procedure has become known as "synthetic division" of polynomials. Note that explicit division of coefficients is only done by b_n; so if b(s) is a monic polynomial (with $b_n=1$), there is no actual division at all.

If b_n has a very small absolute value, large errors may arise on a digital computer implementation of this algorithm. Hence this method may not be numerically stable. (This situation is similar to the Gaussian elimination without any pivoting).

Definition (6.1) [God., 1]: Let $p_1(s),p_2(s),\ldots,p_n(s)\epsilon$F[s] be given polynomials. The polynomial d(s)ϵF[s] which satisfies:

(i) $d(s)|p_i(s)$, $i=1,2,\ldots,n$

(ii) If $\varphi(s)|p_i(s)$, $i=1,2,\ldots,n$ then $\varphi(s)|d(s)$

(iii) $d(s)$ is monic

is called the greatest common divisor (g.c.d.) of $p_i(s)$, $i=1,2,\ldots,n$ and we denote it by $d(s)=(p_1(s),p_2(s),\ldots,p_n(s))$. If $d(s)$ is a nonzero constant (independent of s), then the polynomials are said to be relatively prime or coprime.

■

Proposition (6.1) [God., 1]: Let $a(s),b(s)\epsilon F[s]$ be given polynomials. Then they have a uniquely defined g.c.d. $d(s)\epsilon F[s]$ which can be expressed as

$$d(s) = \sigma(s)a(s)+\tau(s)b(s) \tag{6.2}$$

for $\sigma(s)$, $\tau(s)\epsilon F[s]$.

■

Proposition (6.2) [God., 1]: Let $p_1(s),p_2(s),\ldots,p_n(s)\epsilon F[s]$ be given polynomials. If $d_1(s)=p_1(s)$, $d_2(s)=(d_1(s),p_2(s))$, $d_3(s)=(d_2(s),p_3(s)),\ldots,d_n(s)=(d_{n-1}(s), p_n(s))$, then

$$d_n(s) = (p_1(s),p_2(s),\ldots,p_n(s)) \tag{6.3}$$

■

We present now a method for the evaluation of the g.c.d. Given two polynomials a(s) and b(s), by a sequence of long divisions often called the Euclidean algorithm, we can write

$$\begin{aligned}
a(s) &= q_1(s)b(s)+r_1(s) && \deg\{r_1(s)\}<\deg\{b(s)\}\\
b(s) &= q_2(s)r_1(s)+r_2(s) && \deg\{r_2(s)\}<\deg\{r_1(s)\}\\
r_1(s) &= q_3(s)r_2(s)+r_3(s) && \deg\{r_3(s)\}<\deg\{r_2(s)\}\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots && \\
r_{p-2}(s) &= q_p(s)r_{p-1}(s)+r_p(s) && \deg\{r_p(s)\}<\deg\{r_{p-1}(s)\}\\
r_{p-1}(s) &= q_{p+1}(s)r_p(s)+0 &&
\end{aligned} \tag{6.4}$$

This process will eventually stop because the degree of $r_i(s)$ decreases at each step. We claim that $r_p(s)=(a(s),b(s))$.

From the first equation we see that $(a(s),b(s))|r_1(s)$. Using this fact and the second relation we have $(a(s),b(s))|r_2(s)$. Proceeding downward in the same way, finally we conclude that

$$(a(s),b(s))|r_p(s) \tag{6.5}$$

From the last equation we have: $r_p(s)|r_{p-1}(s)$. Proceeding upward we have $r_p(s)|r_{p-2}(s),\ldots,r_p(s)|b(s)$, $r_p(s)|a(s)$. Thus,

$$r_p(s)|(a(s),b(s)) \tag{6.6}$$

From (6.5), (6.6) we infer that $r_p(s)=(a(s),b(s))$.

Therefore for $a(s)$, $b(s)\epsilon F[s]$ given polynomials with $\deg\{a(s)\}\geq \deg\{b(s)\}$ the Euclidean algorithm produces successive divisions according to the formula:

$$r_i(s) = q_i(s)r_{i+1}(s)+r_{i+2}(s), \quad i=1,2,\ldots \tag{6.7}$$

where $r_1(s)=a(s)$ and $r_2(s)=b(s)$. The polynomials q_i, r_{i+2} are unique and $\deg\{r_{i+2}(s)\}<\deg\{r_{i+1}(s)\}$.

The set of polynomials $c_3\cdot r_3(s)$, $c_4\cdot r_4(s)$, $c_5\cdot r_5(s),\ldots$. where c_i are arbitrary nonzero constants is called a Polynomial Remainder Sequence (P.R.S.) and the essential feature is constructed as part of the process of obtaining a g.c.d.

Let $d_i=\deg\{c_i\cdot r_i(s)\}$, $i=3,4,\ldots$ and note that $d_3>d_4>d_5>\ldots\geq 0$. Let $\delta_i = d_i-d_{i+1}>0$, $i=3,4,\ldots$. If $\delta_i=1$ for all $i>3$, the P.R.S. is called normal otherwise it is called abnormal.

The above method can be formulated into the next algorithm:

Algorithm EUCLID (G.C.D. of polynomials over a field)

Given polynomials $a(s)$, $b(s)\epsilon F[s]$, $\deg\{a(s)\}\geq\deg\{b(s)\}$ this algorithm finds the greatest common divisor $d(s)\epsilon F[s]$ of them. This procedure is called Euclid's algorithm for polynomials over a field; it was first used by S. Stevin in 1585.

STEP 1: **if** $b(s)=0$ **then**

$d(s) := a(s)$

quit

Calculate the remainder $r(s)$ using Algorithm **DIV**

(It is unnecessary to calculate the quotient polynomial $q(s)$)

```
        if r(s) = 0  then
           d(s) := b(s)
           quit
        else if deg{r(s)} = 0 then
                Replace b(s) by the constant polynomial "1"
                d(s) := b(s)
                quit

STEP 2: a(s) := b(s)
        b(s) := r(s)
        Repeat STEP 1
```

Alg: 6.2

Using algorithm **EUCLID** and Proposition (6.2) we can calculate the g.c.d. of a set of polynomials.

Algorithm SETPOL (G.C.D. of set of polynomials over a field)
Given polynomials $p_1(s),p_2(s),\ldots,p_n(s)\epsilon F[s]$ with $\deg\{p_1(s)\}\geq\deg\{p_2(s)\})\geq\ldots\geq\deg\{p_n(s)\}$, this algorithm finds $d(s) = (p_1(s),p_2(s),\ldots,p_n(s))\epsilon F[s]$. The concrete algorithm uses the fact that if two of the n polynomials are coprime, then all of them are coprime.

```
d1(s) := p1(s)
for  i = 2,3,...,n
     Using algorithm EUCLID evaluate
     di(s) := (di-1(s), pi(s))
     if deg{di(s)}=0 then
        d(s) := 1
        quit
d(s) := dn(s)
```

Alg: 6.3

Example (6.1): Let us find the g.c.d. of three polynomials:

$$p_1(s)=s^4+s^3-s-1,\ p_2(s)=s^3+s^2-s-1,\ p_3(s)=s^3+2s^2-s-2$$

$$d_1(s)=p_1(s)=s^4+s^3-s-1$$

$$d_2(s)=(d_1(s),p_2(s))=(s^4+s^3-s-1,s^3+s^2-s-1)=s^2-1$$

$$d_3(s)=(d_2(s),p_3(s))=(s^2-1,s^3+2s^2-s-2)=s^2-1$$

$$d(s)=d_3(s)=s^2-1$$

■

In Appendix C, the extension of Euclid's algorithm to unique factorization domains is presented.

6.3 ROUTH ARRAY METHOD OF CALCULATING THE GCD OF POLYNOMIALS

6.3.1 Background [Fry., 1]

Consider two polynomials $a(s)$, $b(s)\epsilon F[s]$

$$\begin{aligned} a(s) &= a_1s^m+a_2s^{m-1}+\ldots+a_ms+a_{m+1} \\ b(s) &= b_1s^n+b_2s^{n-1}+\ldots+b_ns+b_{n+1} \end{aligned} \tag{6.8}$$

with $m\geq n$.

A technique which is essentially equivalent to Euclid's algorithm uses the Routh array. This has the form:

1st row	$[r_{1j}]$	a_1	a_2	a_3	.	.	.	a_m	a_{m+1}
2nd row	$[r_{2j}]$	b_1	b_2	b_3	.	.	.	b_n	b_{n+1}
3rd row	$[r_{3j}]$	r_{31}	r_{32}	r_{33}	.	.	.	r_{3j+1}	
4th row	$[r_{4j}]$	r_{41}	r_{42}	r_{43}	.	.	.	r_{4j+1}	
.		.		.				.	
.		.		.				.	
.		.		.				.	
i-2 row	$[r_{i-2,j}]$	$r_{i-2,1}$	$r_{i-2,2}$	.	.	.	.	$r_{i-2,j+1}$	
i-1 row	$[r_{i-1,j}]$	$r_{i-1,1}$	$r_{i-1,2}$	.	.	.	.	$r_{i-1,j+1}$	
ith row	$[r_{ij}]$	r_{i1}	r_{i2}	.	.	.		r_{ij}	

The third, and each subsequent row is evaluated from the preceeding two rows by means of a systematic form of calculation

$$r_{ij} \equiv -\frac{\begin{vmatrix} r_{i-2,1} & r_{i-2,j+1} \\ r_{i-1,1} & r_{i-1,j+1} \end{vmatrix}}{r_{i-1,1}}, \quad i=3,4,5,... \tag{6.9}$$

where $r_{1j}=a_j$, $r_{2j}=b_j$, $j=1,2,3,.....$

Each row is constructed from the preceeding two rows, by forming all 2x2 determinants involving the first column.

The process terminates when a row consists entirely of zeros the previous row being composed of the coefficients of the g.c.d. There is a possibility of course that certain elements in the array vanish. Unless the first column-member vanishes, the computation proceeds without difficulty: the array becomes roughly triangular and terminates with a row having only one element. The vanishing of the first column-member creates a special case since, in the formula (6.9) for the next row, division by zero would be required.

The numbers in the two first rows of Routh's array are the coefficients of the original polynomials. The numbers in the other rows also corresponds to coefficients of polynomials, and it is easy to show the relationship of the various rows ,regarded as polynomials, to one another. Consider the long handed division of a(s) by b(s) in (6.8), process being stopped after one subtraction (one cycle):

$$\begin{array}{l|l} a_1 s^m + a_2 s^{m-1} + \ldots\ldots\ldots\ldots\ldots + a_m s + a_{m+1} & b_1 s^n + b_2 s^{n-1} + \ldots + b_n s + b_{n+1} \\ \hline -a_1 s^m - \frac{a_1}{b_1} b_2 s^{m-1} - \frac{a_1}{b_1} b_2 s^{m-2} - \ldots\ldots\ldots\ldots & \frac{a_1}{b_1} s^{m-n} \\ \hline (a_2 - \frac{a_1}{b_1} b_2) s^{m-1} + (a_3 - \frac{a_1}{b_1} b_2) s^{m-2} + \ldots\ldots\ldots & \end{array}$$

The coefficients in the first remainder are exactly the members of the third row of Routh's array. Thus if the rows are regarded as polynomials, the third row is the first remainder when the second row is divided into the first row, and the numerical coefficients of the single quotient term is the ratio of $l(a)=a_1$ to $l(b)=b_1$. More generally, the nth row is the first remainder, when the $(n-1)^{th}$ row is divided into $(n-2)^{th}$ row, and the numerical coefficients of the quotient term is the ratio of the leading term in the (n-2)th row to that of the (n-1)th row. The Routh's algorithm therefore proves to be a method for making a repeated division process, where each division consists of only one cycle, that is, one quotient term and one subtraction.

Therefore, the above process is analogous to the previously described Euclid's algorithm. More specifically, considering the rows $[r_{ij}]$ as polynomials in s we have:

$$r_{ij}(s) = q_i(s) r_{i+1,j} + r_{i+2,j}(s), \quad i=1,2,... \tag{6.10}$$

The following p.r.s. is produced:

$$g_i(s)=r_{2i-2,1}s^{n-i+1}+r_{2i-2,2}s^{n-i}+\ldots+r_{2i-2,n-i},\ i=3,4,\ldots \qquad (6.11)$$

The relationship between the g_i and the Euclidean remainders in (6.7) is given by:

$$\begin{aligned} r_3(s) &= \frac{r_{31}}{r_{21}}\, g_3(s) \\ r_i(s) &= \frac{r_{2i-6,1}\cdot r_{2i-3,1}}{r_{2i-r,1}\cdot r_{2i-4,1}}\, g_i(s), \quad i>3 \end{aligned} \qquad (6.12)$$

apart from possible differences in sign.

Example (6.2): Consider the polynomials

$a(s) = s^4-3s^3+5s^2+7s+2,\quad b(s) = 2s^3+5s^2+s+3$

From (6.7), $r_3(s)=\frac{73}{4}s^2+\frac{33}{4}s+\frac{41}{4},\quad r_4(s)=-\frac{10524}{5329}s+\frac{3728}{5329}$

The first six rows of the array defined by (6.9) are:

Table 6.1

$[r_{1j}]$	1	-3	5	7	2
$[r_{2j}]$	2	5	1	3	
$[r_{3j}]$	$\frac{11}{2}$	$-\frac{9}{2}$	$-\frac{11}{2}$	-2	
$[r_{4j}]$	$-\frac{73}{11}$	-3	$-\frac{41}{11}$		
$[r_{5j}]$	$\frac{510}{73}$	$\frac{627}{73}$	2		
$[r_{6j}]$	$-\frac{2631}{510}$	$\frac{932}{510}$			

Thus, from (6.11) $g_3(s) = -\frac{73}{11}s^2-3s-\frac{41}{11},\ g_4(s) = -\frac{2631}{510}s+\frac{932}{510}$

and from (6.12) it is easily verified that

$$r_3(s)=-(\tfrac{11}{4})g_3(s),\ r_4 = (\tfrac{2040}{5329})g_4(s)$$

■

6.3.2 The numerical algorithm

Algorithm ROUTH

Let us consider the two polynomials of (6.8). This algorithm calculates a greatest common divisor d(s) of a(s) and b(s).

STEP 1: Construct the two first rows

for j = 1,2,...,m+1

$r_{1j} := a_j$

for j = 1,2,...,n+1

$r_{2j} := b_j$

for j = n+2,...,m+1

$r_{2j} := 0$

row1 := row2 := m+1

i := 2

STEP 2: i:= i+1

Construct the new row

for j = 1,2,...,row2-1

$r_{ij} := -(r_{i-2,1} \cdot r_{i-1,j+1})/r_{i-1,1} + r_{i-2,j+1}$

if $r_{ij}=0$, $\forall j$ **then**

row r_{i-1} contains the coefficients of the g.c.d. d(s)

quit

else if $r_{i1}=0$ **then**

shift row r_i to the left until $r_{i1} \neq 0$

fill with zeros the empty positions

if $r_{ij}=0$, $\forall j \neq 1$ **then**

d(s) :=1

quit

row1 := row2

if $r_{i-1,row2} \neq 0$ **then**

$r_{i,row2} := 0$

else

row2 := row2-1

Alg: 6.4

Implementation of the algorithm

We should point out that in order to construct the elements of the i-th row from formula (6.9) the two former rows i-2 and i-1 must contain the same number of elements. Taking into account that the method starts with $m \geq n$ which means that always the first row will generally contain more elements than the second row and that the number of elements of each new row is always one less than the elements of the previous one, it is possible that row i-2 will have one element less than row i-1. In that case if the extra element of the i-1 row is different than zero, we add a zero in the last position of row r_i. (If the extra element is zero we actually delete it from row r_{i-1} because its useless in formula (6.9).)

Example (6.3): Suppose we want to find the g.c.d. of the following polynomials:

$$p_1(s) = s^5+2s^4+3s^3+3s^2+2s+1,\ p_2(s) = s^2+3s+2$$

we form the Routh table by using (6.9)

Table (6.2)

1st row	1	2	3	3	2	1
2nd row	1	3	2	0	0	0
3rd row	-1	1	3	2	1	
4th row	4	5	2	1	0	
5th row	$\frac{9}{4}$	$-\frac{7}{2}$	$-\frac{9}{4}$	1		
6th row	$-\frac{11}{9}$	-2	$-\frac{7}{9}$	0		
7th row	$-\frac{2}{11}$	$\frac{9}{11}$	1			
8th row	$-\frac{15}{2}$	$-\frac{15}{2}$	0			
9th row	1	1				
10th row	0					

Thus we conclude that the g.c.d. of $p_1(s)$ and $p_2(s)$ is:

$$d(s) = s+1$$

■

Example (6.4): Find the g.c.d. of the following two polynomials:

$p_1(s) = s^4+2s^3+s^2+14s+6$, $p_2(s) = 2s^3+4s^2+s+21$

Using (6.9) we have:

Table (6.3)

1st row	1	2	1	14	6
2nd row	2	4	1	21	0
3rd row	0	1	7	6	
		2	2		
	1	7	12	0	
4th row	-10	-23	21		
5th row	$\frac{47}{10}$	$\frac{141}{10}$	0		
6th row	$\frac{329}{47}$	$\frac{987}{47}$			
7th row	0				

We conclude that the g.c.d. of $p_1(s)$ and $p_2(s)$ is:

$$d(s) = \frac{329}{47}s+\frac{987}{47} = \frac{329}{47}(s+3)$$

■

Remark (6.1): It should be clear that any row of the array r_{ij} can be multiplied by any arbitrary nonzero scaling factor before the next row is computed, without doing more than introducing an extra constant factor in the form of the resulting g.c.d. Therefore, we can multiply any complete row either by a negative or positive constant. For instance in the example above, the row written (1/2,7/2,6,0) would more conveniently be expressed as (1,7,12,0).

■

Example (6.5): Find the g.c.d. of the following two polynomials:

$p_1(s) = s^3+6s^2+11s+6$, $p_2(s) = s^3+11s^2+38s+40$

Using (6.9) we have the following Table:

Table (6.4)

1st row	1	6	11	6
2nd row	1	11	38	40
3rd row	-5	-27	-34	0
4th row	$\frac{28}{5}$	$\frac{156}{5}$	40	
5th row	$\frac{24}{28}$	$\frac{48}{28}$	0	
6th row	20	40		
7th row	0			

Finally, the g.c.d. of $p_1(s)$ and $p_2(s)$ is: $d(s) = 20s+40 = 20(s+2)$ ■

In Appendix C, the extension of Routh's algorithm to unique factorization domains is presented.

6.4 MATRIX METHODS FOR COMPUTATION OF THE G.C.D.

6.4.1 Companion matrix method for finding the G.C.D.

(a) Theoretical background

Let $$a(s)=s^m+a_{m-1}s^{m-1}+a_{m-2}s^{m-2}+\ldots+a_1s+a_0 \tag{6.13}$$

$$b(s)=b_ns^n+b_{n-1}s^{n-1}+b_{n-2}s^{n-2}+\ldots+b_1s+b_0 \tag{6.14}$$

be two given polynomials. In this method we assume a(s) to be monic e.g. $a_m=1$ if this is not the case, it can be made so by dividing each coefficient of a(s) by a_m.

Definition (6.2) [Lips., 1]: A companion matrix associated with a(s) is an mxm matrix of the form:

$$C \equiv C_m \equiv \begin{bmatrix} 0 & 1 & 0 & & & & & & \\ 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ -a_0 & -a_1 & -a_2 & \cdot & \cdot & \cdot & -a_{m-2} & -a_{m-1} & \end{bmatrix} \tag{6.15}$$

or

$$C \equiv C_m \equiv \left[\begin{array}{c|ccc} 0 & & I_{m-1} & \\ \hline -a_0 & -a_1 & \cdots & -a_{m-1} \end{array}\right]$$

where I_{m-1} denotes (m-1x(m-1) unity identity matrix.

■

Theorem (6.1) [Lips., 1]: The characteristic polynomial of C_m is a(s) e.g.

$$\det(sI_m - C_m) = s^m + a_{m-1}s^{m-1} + \ldots + a_1 s + a_0. \tag{6.16}$$

■

Remark (6.2): (i) Let e_i denote the i^{th} row of the identity matrix I_m. From (6.15) we conclude that the $(i-1)^{th}$ row of C is e_i, for i=2,3,...,m. This property can be conveniently written as:

$$e_{i-1} \cdot C = e_i, \quad i = 2,3,\ldots,m \tag{6.17}$$

(pre-multiplication of any matrix by e_{i-1} simply picks out its $(i-1)^{th}$ row)

(ii) The first rows of $C, C^2, \ldots, C^{m-1}$ are $e_2, e_3, \ldots, e_m$.

(iii) The identity (6.16) shows that the roots $s_1, s_2, \ldots, s_m$ of a(s) are the eigenvalues of C, so in particular $\det C = s_1 \cdot s_2 \cdot \cdots \cdot s_m$, showing that C is singular if and only if a(s) has a zero root.

(iv) If we transpose C, we obtain a different form of companion matrix C^T, given below

$$C^T \equiv \left[\begin{array}{c|c} 0 & -a_0 \\ \hline & -a_1 \\ & \vdots \\ I_{m-1} & \vdots \\ & -a_{m-1} \end{array}\right] \tag{6.18}$$

that has the same properties as C.

■

A key feature in the development of the g.c.d. problem, is the study of polynomials in matrix C, that is if b(s) is the polynomial in (6.14) we construct,

$$b(C) \equiv b_n C^n + b_{n-1} C^{n-1} + b_{n-2} C^{n-2} + \ldots + b_1 C + b_0 I_m \tag{6.19}$$

Let r_i, $i=1,2,\ldots,m$ denote the rows of matrix $b(C)$. If $\deg\{b(s)\}=n<m$, then from Remark (6.2) the first row r_1 of $b(s)$ is:

$$r_1=b_n e_{n+1}+b_{n-1}e_n+\ldots+b_0 e_1=[b_0,b_1,\ldots,b_{n-1},b_n] \tag{6.20}$$

For the remaining rows we have by definition:

$$r_i = e_i \cdot b(C) \tag{6.21}$$

and substitution of (6.17) into (6.21) gives,

$$r_i = e_{i-1}\cdot C\cdot b(C) = e_{i-1}\cdot b(C)\cdot C \tag{6.22}$$

using the fact that C commutes with $b(C)$.
Combining (6.21) and (6.22) shows that (6.22) can be written as:

$$r_i = r_{i-1}\cdot C, \quad i = 2,3,\ldots m \tag{6.23}$$

Since r_1 is given by (6.20), the recurrence relation (6.23) provides a simple way of constructing $b(C)$.

Another way of writing $b(C)$ is obtained by noting that (6.23) implies that $r_2=r_1\cdot C$, $r_3=r_2\cdot C=r_1\cdot C^2$, and so on, so that we have established:

Theorem (6.2) [Kal., 3]: For $b(s)$ in (6.14) with $\deg\{b(s)\}=n<m$ and C in (6.15), the matrix $b(C)$ has rows

$$r_1,\ r_1\cdot C,\ r_1\cdot C^2,\ldots,r_1\cdot C^{m-1}$$

where r_1 is given by (6.20). ■

The resultant of $a(s)$ and $b(s)$ ($R(a,b)$) can be expressed in the terms of this companion matrix formulation.
If s_i, $i=1,2,\ldots,m$ are the roots of $a(s)$ and τ_j, $j=1,2,\ldots,n$ are the roots of $b(s)$ then

$$R(a,b) = b_n^m \prod_{i=1}^{m}\prod_{j=1}^{n} (s_i-\tau_j) \tag{6.24}$$

Using the facts that if $s_1,s_2,\ldots,s_m$ are the eigenvalues of C, then those of $b(C)$ are $b(s_1),b(s_2),\ldots,b(s_m)$ and that the determinant of a matrix is equal to the product of its eigenvalues, (6.24) can be written as:

$$R(a,b) = b_n \prod_{j=1}^{n} (s_1-\tau_j)\cdot b_n \prod_{j=1}^{n} (s_2-\tau_j)\ldots b_n \prod_{j=1}^{n} (s_m-\tau_j)$$

$$= b(s_1)\cdot b(s_2)\cdots b(s_m) = \det(b(C))$$

Theorem (6.3) [Mac., 1]: The resultant of a(s) and b(s) in (6.13) and (6.14) respectively is given by

$$R(a,b) = \det(b(C))$$

where C is the matrix of (6.15). In other words, b(C) is nonsingular if and only if these polynomials are coprime.

■

Example (6.6): Let $a(s)=s^3-3s^2+s+5$, $b(s)=2s^2-3s+1$ be two given polynomials. The companion matrix (6.15) of a(s) is,

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -1 & 3 \end{bmatrix}$$

Matrix $b(C)=2C^2-3C+1 \in R^{3x3}$ and has rows r_1, r_2, r_3.
From (6.20) we have r_1 = [1, -3, 2], and from (6.23) $r_2=r_1 \cdot C$= [-10,-1,3], $r_3=r_2 \cdot C$=[-15, -13, 8].
Hence

$$b(C) = \begin{bmatrix} 1 & -3 & 2 \\ -10 & -1 & 3 \\ -15 & -13 & 8 \end{bmatrix}$$

■

Note that since the first row of b(C) is given by (6.20), it follows that b(C)=0 if and only if r_1=0, which is equivalent to having b(s)=0. Thus there is no nontrivial polynomial of degree less than m such that b(C)=0.

Theorem (6.1) and the Cayley-Hamilton theorem together imply

$$a(C) = C^m+a_{m-1} \cdot C^{m-1}+\ldots+a_1 \cdot C+a_0 = 0 \qquad (6.25)$$

In Theorem (6.2) it was assumed that deg{b(s)}<deg{a(s)} but (6.25) gives us way of removing this. Suppose deg{b(s)=n≥m; then we can divide a(s) by b(s) to give

$$b(s) = a(s)q_1(s)+r(s) \qquad (6.26)$$

where deg{r(s)}<m. Since above is an identity, we can replace s by C to give

$$b(C) = a(C)q_1(C)+r(C) = r(C) \tag{6.27}$$

by virtue of (6.25).

Hence to compute b(C) when deg{b(s)}≥deg{a(s)}, simply divide b(s) by a(s), and the desired expression is obtained by applying Theorem (6.2) to the remainder polynomial r(C). In particular, when deg{a(s)}=deg{b(s)}=n then,

$$r(s)=(b_{n-1}-b_n a_{n-1})s^{n-1}+(b_{n-2}-b_n a_{n-2})s^{n-2}+\ldots+(b_0-b_n a_0)$$

so b(C) has first row

$$r_1 = [b_0-b_n a_0,\ldots\ldots, b_{n-2}-b_n a_{n-2}, b_{n-1}-b_n a_{n-1}] \tag{6.28}$$

In addition (6.27) reveals that if a(s) divides b(s), then $\Upsilon(s)=0$.

Example (6.7): Let $a(s)=s^3-3s^2+s+5$, $b(s)=s^4+5s^3-20s^2+6s+44$ be two given polynomials. The companion matrix is the same as above. Since deg{b(s)}>deg{s(s)} we first do the division by a(s), which gives:

$$b(s) = (s+8)a(s)+(3s^2-7s+4)$$

so $b(C)=3C^2-7C+4I_3$. Theorem (6.2) shows that b(C) has rows

$$r_1 = [4, -7, 3], \quad r_2 = r_1\cdot C = [-15, 1, 2],$$

$$r_3 = r_2\cdot C = [-10, -17, 7]$$

■

(b) G.C.D. by using companion matrix

It has been shown above that the det(b(C)) forms the resultant of a(s) and b(s). If the polynomials a(s) and b(s) are not coprime, then their g.c.d. itself can be obtained in an easy way from the matrix b(C). In the discussion below the companion form C^T will be used. It follows that

$$R_0 \equiv b(C^T) = [b(C)]^T$$

so the columns $e_1,e_2,\ldots,e_m$ of R_0 are equal to the transpose of the rows of b(C). If deg{b(s)}=m<n=deg{a(s)} then (6.20) gives

$$e_1 = [b_0,b_1,\ldots,b_{n-1},b_n]^t \tag{6.29}$$

(6.22) and (6.23) become

$$e_i = C^T\cdot e_{i-1}, \quad i=2,3,\ldots,m \tag{6.30}$$

and Theorem (6.2) implies that

$$R_0 \equiv b(C^T) = [e_1,C^Te_1,\ldots,(C^T)^{m-1}e_1] \tag{6.31}$$

Theorem (6.4) [Bar., 1]: The degree of the g.c.d. d(s) of a(s) and b(s), with deg{a(s)}≥deg{b(s)}, is equal to m-ρ(R_0).

■

Let $$d(s)=s^k+d_{k-1}s^{k-1}+d_{k-2}s^{k-2}+\ldots+d_1s+d_0, \tag{6.32}$$

be the g.c.d. of a(s) and b(s), k<m=deg{a(s)}.

Theorem (6.5) [Bar., 1]: The last m-k rows of $R_0=b(C^T)$ are linearly independent and if $r_i=\sum_{j=k+1}^{m} x_{ij}\cdot r_j$, i=1,2,...,k where r_i denotes the i-th row of R_0, then

$$d_{k-p} = x_{k+1-p,k+1}, \quad p=1,2,\ldots,k \tag{6.33}$$

■

Example (6.8): Let us find the g.c.d. of the following two polynomials:

$$a(s) = s^3-3s^2+s+5, \quad b(s) = s^2-4s+5$$

Form the companion matrix from the polynomial a(s):

$$C^T = \begin{bmatrix} 0 & 0 & -5 \\ 1 & 0 & -1 \\ 0 & 1 & 3 \end{bmatrix}$$

Now from (6.29), (6.30) and (6.31) construct matrix R_0

$$R_0=b(C^T) = \begin{bmatrix} 5 & -5 & 5 \\ -4 & 4 & -4 \\ 1 & -1 & 1 \end{bmatrix} \begin{matrix} \longrightarrow r_1 \\ \longrightarrow r_2 \\ \longrightarrow r_3 \end{matrix}$$

ρ(R_0)=1, therefore the degree k of the g.c.d. is equal to m-ρ(R_0) = 3-1=2. The g.c.d. d(s) of a(s), b(s) is of the form: $d(s)=s^2+d_1s+d_0$.
From Theorem (6.5) we have:

$$r_1 = x_{13}\cdot r_3$$

$$r_2 = x_{23}\cdot r_3$$

It is evident that $x_{13}=5$ and $x_{23}=-4$. Thus,

$$d_1=x_{23}=-4, \quad d_0=x_{13}=5 \quad \text{and} \quad d(s)=s^2-4s+5.$$

■

Remark (6.3): (i) If b(C) is used instead of $b(C^T)$, the rows r_i in the statement of Theorem (6.5) become the columns c_i of b(C). In particular, (6.33) is replaced by,

$$c_i = \sum_{j=k+1}^{m} x_{ij} \cdot c_j, \quad i=1,2,\ldots,k$$

(ii) The solution of system (6.33) is not always as simple as in Example (6.8), thus a numerical technique must be used for the evaluation of x_{ij}. ■

In the sequel, we present the application of the above method to more than two polynomials and the numerical technique used for the solution of the corresponding linear system.

Let Π be a set of polynomials. It can be assumed that, after division if necessary, one polynomial can be denoted by

$$a(s) = s^m + a_{m-1}s^{m-1} + \ldots + a_1 s + a_0 \tag{6.34}$$

and the remaining members of Π by

$$b_i(s) = b_{im-1}s^{m-1} + b_{im-2}s^{m-2} + \ldots + b_{i0}, \quad i=1,2,\ldots,n$$

where the b_{im-1} need not, of course, be nonzero. Let

$$C^T = \begin{bmatrix} 0 & 0 & \ldots & -a_0 \\ 1 & 0 & \ldots & -a_1 \\ 0 & 1 & \ldots & -a_2 \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ 0 & 0 & \ldots & -a_{m-1} \end{bmatrix} \tag{6.35}$$

be a companion matrix of a(s), and form the matrix polynomials

$$b_i(C^T) = b_{im-1}(C^T)^{m-1} + b_{im-2}(C^T)^{m-2} + \ldots + b_{i0}I_m, \quad i=1,2,\ldots,n$$

Theorem (6.6) [Bar., 2]: The degree of the g.c.d. of the set Π is $k=m-\rho(R)$, where

$$R = [b_1(C^T), b_2(C^T), \ldots, b_n(C^T)] \tag{6.36}$$

■

Corollary (6.1): The polynomials a(s), $b_1(s),\ldots,b_n(s)$ are relatively prime if and only if the matrix R has rank m.

■

Theorem (6.7) [Bar., 2]: If the rows of R are denoted by $r_1, r_2, \ldots, r_m$ then $r_{k+1}, r_{k+2}, \ldots, r_m$ are linearly independent and if

$$r_i = \sum_{j=k+1}^{m} x_{ij} \cdot r_j, \quad i=1,2,\ldots,k \tag{6.37}$$

then the unique monic g.c.d. of Π is $d(s)=s^k+d_{k-1}s^{k-1}+\ldots+d_1s+d_0$, where

$$d_{k-p} = x_{k+1-p,k+1}, \quad p=1,2,\ldots,k$$

■

The usefulness of Theorems (6.7) and (6.8) can be increased by establishiing a simple expression for the matrix R in (6.36). It is easily proved [Bar., 2] that R can be expressed as

$$R = [B,\ C^TB,\ (C^T)^2B,\ldots,\ (C^T)^{m-1}B], \tag{6.38}$$

where

$$B = \begin{bmatrix} b_{10} & b_{20} & \cdots & b_{n0} \\ b_{11} & b_{21} & \cdots & b_{n1} \\ \vdots & \vdots & & \vdots \\ b_{1m-1} & b_{2m-1} & & b_{nm-1} \end{bmatrix}$$

The matrix on the right in (6.38) is recognized as the well-known controllability matrix [Chen, 1] for the constant linear control system $\dot{x}=Ax+Bu$. Thus the controllability criterion for the above system is equivalent to the Corollary (6.1).

Computationally the solution of (6.37) is encountered by using the following numerical technique [Pac. & Bar., 1]. If R is partitioned as

$$R = \begin{bmatrix} R_1 & R_2 \\ \hline R_3 & R_4 \end{bmatrix} \begin{matrix} k \\ m-k \end{matrix} \tag{6.39}$$

$$\qquad\quad mn-m+k \qquad m-k$$

then R can be decomposed into the product of two matrices by performing a type of Gaussian elimination, i.e. determining (mxm) matrices J_i such that

$$J_{m-k}\, J_{m-k-1} \ldots J_2 J_1 R = L \tag{6.40}$$

where, typically

$$J_i = \begin{bmatrix} 1 & 0 & & -x_{1,m+1-i} & & 0 \\ & 1 & & \vdots & & \vdots \\ & & \ddots & \vdots & & \vdots \\ & & & -x_{m-i,m+1-i} & & \\ & & & 1 & & \vdots \\ & & & & \ddots & \vdots \\ & & & & & \vdots \\ & 0 & & & & 1 \end{bmatrix} \tag{6.41}$$

and

$$L = \begin{bmatrix} 0 & 0 \\ - - & - - \\ L_1 & L_2 \end{bmatrix} \begin{matrix} k \\ \\ m-k \end{matrix}$$
$$\begin{matrix} mn-m+k & m-k \end{matrix}$$

In [Pac. & Bar., 1] it is proved that

$$R=[J_{m-k}J_{m-k-1}\ldots J_2J_1]^{-1}\cdot L = \begin{bmatrix} I & U_1 \\ - - - & - - - \\ 0 & U_2 \end{bmatrix} \begin{matrix} k \\ \\ m-k \end{matrix} \tag{6.42}$$
$$\begin{matrix} k & m-k \end{matrix}$$

It is also proved in [Pac. & Bar.1] that the first column of U_1 gives the coefficients of the monic g.c.d. of Π.

Example (6.9): Let $a(s) = s^5+s^4-9s^3-5s^3+16s+12$, $b(s) = s^4-3s^3+s^2+3s-2$ be given polynomials. We want to evaluate their g.c.d. using the companion method.

Form the companion matrix corresponding to polynomial a(s)

$$C^T = \begin{bmatrix} 0 & 0 & 0 & 0 & -12 \\ 1 & 0 & 0 & 0 & -16 \\ 0 & 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{6.43}$$

Construct matrix R_0

$$R_0 = b(C^T) = \begin{bmatrix} -2 & -12 & 48 & -168 & 504 \\ 3 & -18 & 52 & -176 & 504 \\ 1 & 8 & -38 & 122 & -386 \\ -3 & 10 & -28 & 88 & -256 \\ 1 & -4 & 14 & -42 & 130 \end{bmatrix}$$

$\rho(R_0)=3$ thus the degree of the g.c.d. is $m-\rho(R_0)=5-3=2$. In the sequel, we perform the UL-decomposition defined by (6.42) to matrix R_0. Matrices U,L will be of the form

$$U = \left[\begin{array}{c|c} I & U_1 \\ \hline 0 & U_2 \end{array}\right]\begin{matrix}2\\3\end{matrix}, \qquad L = \left[\begin{array}{c|c} 0 & 0 \\ \hline L_1 & L_2 \end{array}\right]\begin{matrix}2\\3\end{matrix} \qquad (6.44)$$

$$\begin{matrix} & 2 & & 3 \end{matrix}$$

In fact,

$$U = \left[\begin{array}{cc|ccc} 1 & 0 & -2 & 10 & 2 \\ 0 & 1 & -1 & 3 & 3 \\ \hline 0 & 0 & 1 & -6 & 1 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right], L = \left[\begin{array}{cc|ccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 32 & -64 & 268 \\ 0 & -2 & 14 & -38 & 134 \\ 1 & -4 & 14 & -42 & 130 \end{array}\right] \qquad (6.45)$$

A comparison of (6.44) and (6.45) shows that

$U_1 = \begin{bmatrix} -2 & 10 & 2 \\ -1 & 3 & 3 \end{bmatrix}$ and thus its first column contains the

coefficients of the g.c.d. Therefore, the g.c.d. of the given polynomials is $d(s)=s^2-s-2$.

■

Example (6.10): Let have the set

$\Pi = \{a(s) = s^5+s^4-9s^3-5s^2+16s+12,\ b_1(s) = s^4-3s^3+s^2+3s-2,\ b_2(s) = s^2+5s-14\ \}$

We want to calculate their g.c.d. using the companion method.

The companion matrix C^T corresponding to polynomial a(s) is given by (6.43). Matrix R_0 is equal to [B, $(C^T)B$, $(C^T)^2B$, $(C^T)^3B$, $(C^T)^4B$]. More analytically,

$$R_0=\left[\begin{array}{cc|cc|cc|cc|cc} -2 & -14 & -12 & 0 & 48 & 0 & -168 & -12 & 504 & -48 \\ 3 & 5 & -18 & -14 & 52 & 0 & -176 & -16 & 504 & -76 \\ 1 & 1 & 8 & 5 & -38 & -14 & 122 & 5 & -368 & 4 \\ -3 & 0 & 10 & 1 & -28 & 5 & 88 & -5 & -256 & 41 \\ 1 & 0 & -4 & 0 & 14 & 1 & -42 & 4 & 130 & -9 \end{array}\right] \qquad (6.46)$$

Notice that each 5x2 block of matrix R_0 is obtained by premultiplying the preceeding block by C^T. $\rho(R_0)=4$ and thus the degree of the g.c.d. is m-$\rho(R_0)$=5-4=1. Matrices U,L of the UL-decomposition of matrix R_0 will be of the form

$$U=\left[\begin{array}{c|c} I & U_1 \\ \hline 0 & U_2 \end{array}\right]\begin{matrix}1\\4\end{matrix}, \qquad L=\left[\begin{array}{c|c} 0 & 0 \\ \hline L_1 & L_2 \end{array}\right]\begin{matrix}1\\4\end{matrix} \qquad (6.47)$$

$$\begin{matrix} 1 & 4 \end{matrix} \qquad \begin{matrix} 1 & 4 \end{matrix}$$

More precisely,

$$U=\left[\begin{array}{c|cccc} 1 & -2 & -\frac{10}{11} & 10 & -2 \\ \hline 0 & 1 & -\frac{17}{11} & 3 & 3 \\ 0 & 0 & 1 & -6 & 1 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \qquad (6.48)$$

$$L=\left[\begin{array}{cccccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{72}{11} & 0 & 0 & \frac{192}{11} & -\frac{1882}{11} & -\frac{384}{11} & \frac{192}{11} & \frac{2034}{11} & -\frac{1306}{11} \\ 0 & 1 & 0 & 11 & 32 & -189 & -64 & 43 & 306 & 17 \\ 0 & 0 & -2 & 1 & 14 & -25 & -38 & 7 & 134 & 14 \\ 1 & 0 & -4 & 0 & 14 & 1 & -42 & 4 & 130 & -9 \end{array}\right]$$

A comparison of (6.47) and (6.48) shows that

$U_1=[-2\ -\frac{10}{11}\ 10\ -2]$ and thus the g.c.d. of the given set Π is $d(s)=s-2$.

■

The above analysed method of finding the g.c.d. of several polynomials using the companion matrix, is summarized to the following algorithm.

Algorithm COMPANGCD

Let Π be a given set consisting of the subsequent polynomials

$a(s) = s^m+a_{m-1}s^{m-1}+\ldots+a_1s+a_0$

$b_i(s) = b_{im-1}s^{m-1}+b_{im-2}s^{m-2}+\ldots+b_{i1}s+b_{i0},\quad i=1,2,\ldots,n$

The following algorithm evaluates the g.c.d. of the above set using the companion matrix method.

STEP 1: Construct the companion matrix C^T of $a(s)$

STEP 2: Form matrix $B=[\underline{b}_1,\ldots,\underline{b}_n]\in R^{mxn}$, where

$\underline{b}_i = [b_{i0}\ b_{i1}\ \ldots b_{im-1}]^t$, $i=1,2,\ldots,n$.

STEP 3: Construct matrix $R_0=[B,\ C^TB,\ (C^T)^2B,\ldots,(C^T)^{m-1}B]$

$k := \rho(R_0)$

STEP 4: Perform the decomposition $R_0=U\cdot L$ where

U,L are given from (6.42)

The first column of the (m-k)xk part of U contains

the coefficients of the g.c.d. of Π

Alg: 6.5

Comments about the implementation and the computational complexity of algorithm **COMPANGCD** can be found in [Pac. & Bar.,1].

6.4.2 Sylvester's Resultant matrix method for finding the G.C.D.

(a) Theoretical background

Let

$a(s) = a_0 s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n$, $a_0 \neq 0$, $n>0$

$b(s) = b_0 s^m + b_1 s^{m-1} + \ldots + b_{m-1} s + b_m$, $b_0 \neq 0$, $m>0$

be two given polynomials.

Definition (6.3) [Boch., 1]: The following (m+n)x(m+n) matrix

$$S = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_n & 0 & \ldots & 0 & \ldots & 0 \\ 0 & a_0 & a_1 & a_2 & \ldots & a_n & \ldots & 0 & \ldots & 0 \\ 0 & 0 & a_0 & a_1 & a_2 & \ldots & a_n & \ldots & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \ldots & 0 & a_0 & \ldots & \ldots & a_{n-1} & a_n \\ b_0 & b_1 & b_2 & \ldots & \ldots & b_m & 0 & \ldots & \ldots & 0 \\ 0 & b_0 & b_1 & b_2 & & & b_m & \ldots & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdot & \cdot & \cdot & & \cdot & b_0 & b_1 & \ldots & b_m \end{bmatrix} \qquad (6.49)$$

is called Sylvester's matrix and its determinant $R\left(\begin{smallmatrix} a_0,\ldots,a_n \\ b_0,\ldots,b_m \end{smallmatrix}\right)$ is called Sylvester's Resultant.

■

Theorem (6.8) [Boch., 1]: A necessary and sufficient condition for two polynomials to be relatively prime is that their resultant do not vanish.

■

In some applications it is more convenient to use a slightly different form of S in which the last n rows are in reversed order. In this case, construct centrally situated sub-matriceds by successively deleting a row and a column all the way round. For example, if n=3, m=2 we get

$$S = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & 0 \\ 0 & a_0 & a_1 & a_2 & a_3 \\ 0 & 0 & b_0 & b_1 & b_2 \\ 0 & b_0 & b_1 & b_2 & 0 \\ b_0 & b_1 & b_2 & 0 & 0 \end{bmatrix}$$

Definition (6.4) [Boch., 1]: By the i-th subresultant R_i of two polynomials is understood the determinant obtained by striking out the first i and the last i rows and also the first i and the last i columns from the resultant of these polynomials.

■

Thus if n=3, m=2 R is a determinant of fifth order, R_1 of third and R_2 of first as indicated below:

$$R = \begin{vmatrix} a_0 & a_1 & a_2 & a_3 & 0 \\ 0 & a_0 & a_1 & a_2 & a_3 \\ 0 & 0 & \boxed{b_0} & b_1 & b_2 \\ 0 & b_0 & b_1 & b_2 & 0 \\ b_0 & b_1 & b_2 & 0 & 0 \end{vmatrix}$$

(R_1 = the 3×3 block, R_2 = the 2×2 block)

Theorem (6.9) [Boch., 1]: The degree of the greatest common divisor of a(s) and b(s) is equal to the subscript of the first of the subresultants R_0=R, R_1, R_2,... which does not vanish.

■

Theorem (6.10) [Boch., 1]: If i is the degree of the greatest common divisor of two polynomials a(s) and b(s), then this greatest common divisor may be obtained from the ith subresultant of a and b by replacing the last element in the last row of coefficients of a by a(s), the element just above this by s a(s), the element above this by s^2a(s), etc.; and replacing the last element in the first row of coefficients of b by b(s), the element below this by s·b(s) etc.

■

Example (6.11): Let $a(s) = s^3-3s^2+s+5$ and $b(s)=s^2-4s+5$ be two given polynomials. We want to calculate their g.c.d.

The Sylvester matrix for the above two polynomials is given below:

$$S = \begin{bmatrix} 1 & -3 & 1 & 5 & 0 \\ 0 & 1 & -3 & 1 & 5 \\ 1 & -4 & 5 & 0 & 0 \\ 0 & 1 & -4 & 5 & 0 \\ 0 & 0 & 1 & -4 & 5 \end{bmatrix}$$

The first non vanishing subresultant is $R_2=5$ thus the degree of the g.c.d. is 2. We replace 5 by $b(s)=s^2-4s+5$ and therefore the g.c.d. of a(s) and b(s) is equal to s^2-4s+5.

■

(b) G.C.D. of several polynomials via Sylvester's matrix

Consider a set Π of m+1 polynomials whose maximum degree is n, so that one polynomial can be written without loss of generality as

$$a(s) = s^n+a_1s^{n-1}+\ldots a_n \tag{6.50}$$

and the remaining members of Π as

$$b_i(s) = b_{i0}s^n+b_{i1}s^{n-1}+\ldots+b_{in},\ i=1,2,\ldots,m \tag{6.51}$$

Suppose that the maximum degree amongst the polynomials $b_1(s),\ldots,b_m(s)$ in (6.51) is $p\leq n$, i.e. $b_{i,n-p}\neq 0$ for at least one i, but $b_{ij}=0$ for $j<n-p$, all s. Define a px(n+p) matrix associated with a(s)

$$S_0 = \begin{bmatrix} 1 & a_1 & a_2 & \ldots & a_n & 0 & .. & 0 & 0 \\ 0 & 1 & a_1 & \ldots & a_{n-1} & a_n & .. & 0 & 0 \\ . & . & . & \ldots & . & . & .. & . & . \\ 0 & 0 & 0 & 1 & . & . & .. & a_{n-1} & a_n \end{bmatrix} \tag{6.52}$$

and an nx(n+p) matrix associated with $b_i(s)$:

$$S_i = \begin{bmatrix} b_{i,n-p} & b_{i,n-p+1} & \ldots & b_{in} & 0 & .. & 0 & 0 \\ 0 & b_{i,n-p} & \ldots & b_{i,n-1} & b_{in} & .. & 0 & 0 \\ . & . & \ldots & . & . & .. & . & . \\ 0 & 0 & 0 & . & b_{i,n-p} & . & . & b_{i,n-1} & b_{in} \end{bmatrix} \tag{6.53}$$

for i=1,2,...,m.

The extended Sylvester resultant matrix for the set Π can then be defined by

$$S = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ . \\ . \\ . \\ S_m \end{bmatrix} \tag{6.54}$$

and has dimensions (mn+p)x(n+p). When m=1, S is the classical Sylvester matrix.

Theorem (6.11) [Bar., 4]: The degree of the g.c.d. of the set Π is equal to the rank defect of S.

■

More explicitly, Theorem (6.11) states that if k is the degree of the g.c.d. of Π, then $k=(n+p)-\rho(S)$.

Corollary (6.2) [Bar., 4]: The polynomials $a(s)$, $b_1(s),\ldots,b_m(s)$ are relatively prime if and only if S has rank n+p.

■

Theorem (6.12) [Bar., 4]: If the extended resultant matrix (6.54) is put into row echelon form using row transformations only, then the last nonvanishing row gives the coefficient of a g.c.d.

■

A different proof of the above Theorem, for the case of m=1, can be found in [Lai,1]. Another version of Theorem (6.12) can be also found in [Vard. & Sto., 1].

Example (6.12): Let us find the g.c.d. of the following three polynomials using the Sylvester's matrix method.

$a(s) = s^3+6s^2+11s+6$

$b_1(s) = s^2+2s-3$

$b_2(s) = s^2+s-6$

The highest degree polynomial is chosen as a(s), thus n=3 and p=2. In the sequel we form matrices S_0, S_1, S_2 as follows:

$$S_0 = \begin{bmatrix} 1 & 6 & 11 & 6 & 0 \\ 0 & 1 & 6 & 11 & 6 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 2 & -3 & 0 & 0 \\ 0 & 1 & 2 & -3 & 0 \\ 0 & 0 & 1 & 2 & -3 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 1 & 1 & -6 & 0 & 0 \\ 0 & 1 & 1 & -6 & 0 \\ 0 & 0 & 1 & 1 & -6 \end{bmatrix}$$

The extended resultant matrix is of the form:

$$S = \begin{bmatrix} 1 & 6 & 11 & 6 & 0 \\ 0 & 1 & 6 & 11 & 0 \\ 1 & 2 & -3 & 0 & 0 \\ 0 & 1 & 2 & -3 & 0 \\ 0 & 0 & 1 & 2 & -3 \\ 1 & 1 & -6 & 0 & 0 \\ 0 & 1 & 1 & -6 & 0 \\ 0 & 0 & 1 & 1 & -6 \end{bmatrix}$$

$\rho(S)=4$. Thus the degree of the g.c.d. is 5-4=1. The above matrix is reduced to row echelon form and the resulting matrix is given below:

$$S_{ech} = \begin{bmatrix} 1 & 6 & 11 & 6 & 0 \\ 0 & 1 & 6 & 11 & 6 \\ 0 & 0 & 1 & 2 & -3 \\ 0 & 0 & 0 & -6 & -18 \\ - & - & - & - & - \\ & & 0 & & \end{bmatrix}$$

Thus the g.c.d. is given by the last nonzero row:

$$-6s-18 = -6(s+3).$$

■

The above analysed method of finding the g.c.d. of several polynomials using the Sylvester's Resultant matrix, is summarized to the following algorithm.

Algorithm SYLVESTER

Let Π be a given set of m+1 polynomials. The following algorithm evaluates the g.c.d. of the above set using the Sylvester's Resultant matrix method.

STEP 1: Choose a(s) the highest degree polynomial of Π

$n := \deg(a(s))$

$p :=$ the maximum degree amongst the rest polynomials of Π

STEP 2: From formulas (6.52), (6.53) form matrices

$S_0 \epsilon R^{p \times (n+p)}$, $S_i \epsilon R^{n \times (n+p)}$, $i=1,2,\ldots,m$

$S := [S_0, S_1, \ldots, S_m]^T$

STEP 3: Reduce matrix S to its row echelon form S_{ech}

STEP 4: The last nonzero row of Sech contains the coefficients of the g.c.d. of Π.

Alg: 6.6

Useful comments about the implementation of algorithm **SYLVESTER** can be found in [Pac. & Bar., 1]. An advantage of using Sylvester's matrix method for finding the g.c.d. of polynomials is that S can be immediately written down, compared with the construction of $b(C^T)$. However, a disadvantage is that S has dimensions (n+p)x(n+p), compared only with nxn for $b(C^T)$ in the case of two polynomials. This becomes more important when we have more than two polynomials.

6.4.3 Blankiship's method for calculating the G.C.D. of polynomials [Blan., 1]

Given a set of polynomials $\Phi=\{p_1(s), p_2(s), \ldots, p_n(s)\}$ the previous developed Euclid's algorithm provides a method of computing the g.c.d., d(s), of these polynomials. If the steps performed during the algorithm are traced back it is possible to deduce multipliers $x_1(s), x_2(s), \ldots, x_n(s)$ such that:

$$d(s) = p_1(s)x_1(s)+p_2(s)x_2(s)+\ldots+p_n(s)x_n(s).$$

Blankiship's method uses a trick, well-known to the computing trade, of carrying along a matrix to keep track of the operations which have been performed. The g.c.d. and the multipliers are determined by performing elementary row operations on the matrix $B=[P \mid I_n]$, where

$$P^T = [p_1(s),\ p_2(s),\ldots,p_n(s)]$$

and I_n is an nxn identity matrix. The operations are performed on this matrix until there is only one nonzero element in the first column. If we refer to the first element of a row as the leader of that row, the algorithm can be summarized as follows:

Algorithm BLANKISHIP (Outline)

STEP 1: For a given set of polynomials

$\Phi = \{p_1(s),\ p_2(s),\ldots,p_n(s)\}$ construct matrix

$$B = \left[\begin{array}{c|c} \begin{matrix} p_1(s) \\ p_2(s) \\ \cdot \\ \cdot \\ \cdot \\ p_n(s) \end{matrix} & I_n \end{array}\right]$$

STEP 2: Select the row of B with the leader of the lowest degree and call it the "operator".

STEP 3: Select any other row with a nonzero leader and call it the "operand".

if there is not such row **then**

the remaining row with non-zero leader is

$d(s)\ x_1(s)\ x_2(s)\ .\ .\ .\ x_n(s)$ and

$d(s)=p_1(s)x_1(s)+p_2(s)x_2(s)+\ldots+p_n(s)x_n(s)$

quit

STEP 4: Divide the leading term of the leader of the operator into the leading term of the leader of the operand, ignoring the remainder, and call the quotient q.

STEP 5: Subtract q times the operator from the operand, recording the result as a new row and striking out the operand.

Repeat **STEP 2**.

Alg: 6.7

The fact that the process terminates is easily seen by noting that every time **STEP 5** is performed, the degree of column leader decreases but never becomes negative. Hence the sum of the degrees of column leaders is a strictly decreasing positive integer. Since it cannot decrease more than $\sum_i \text{degp}_i(s)$ times, the process must terminate.

We next note that elementary row operations (such as in **STEP 5**) preserve the greatest common divisor of the polynomials; that is

$$\text{g.c.d.}[p_1(s),p_2(s),\ldots,p_n(s)]=\text{g.c.d.}[p_1(s)+ap_j(s),\ldots,p_n(s)]$$

for any integer a and $j \leq n$ different from 1.

When the last step is reached all the leaders are zero except that of the previous operand and that polynomial must be the g.c.d. of the original set of leaders. These operations can be represented by nonsingular matrices M_i such that the final result is of the form

$$\ldots M_3M_2M_1[P|I_n] = \left[\begin{array}{c|ccc} 0 & & & \\ \vdots & & & \\ 0 & & & \\ d(s) & & M & \\ 0 & & & \\ \vdots & & & \\ 0 & & & \end{array}\right] = [M\cdot P|M] \qquad (6.55)$$

where M is the produce of the M_i. If the g.c.d. occurs in the jth row then from (6.55)

$$d(s) = r_j[M]P,$$

where $r_j[X]$ denotes jth row of X, and M is now overwritten on the unit matrix. This completes the proof that the algorithm works. It is also useful to

remember that a row of the matrix B at any stage of the algorithm, represents a linear equation relating the leader of that row to a linear combination of the original polynomials. Thus performing elementary row operations on matrix B until only one nonzero element remains in the first column produces a g.c.d. (not necessarily monic) and a set of multipliers.

Example (6.13): Let us compute the g.c.d. of the set

$\phi=\{p_1(s)=s^4+4s^3+4s^2+4s+3,\ p_2(s)=2s^3+11s^2+16s+3,\ p_3(s)=s^2+5s+6\}$

Matrix B is of the form $\left[\begin{array}{c|c} \begin{matrix} p_1(s) \\ p_2(s) \\ p^3(s) \end{matrix} & I_3 \end{array}\right]$ and more explicitly

$$B = \left[\begin{array}{c|ccc} s^4+4s^3+4s^2+4s+3 & 1 & 0 & 0 \\ 2s^3+11s^2+16s+3 & 0 & 1 & 0 \\ s^2+5s+6 & 0 & 0 & 1 \end{array}\right]$$

Iteration 1: "operator": s^2+5s+6

"operand" : $s^4+4s^3+4s^2+4s+3$

$q = s^4/s^2=s^2$

$$B = \left[\begin{array}{c|ccc} -s^3-2s^2+4s+3 & 1 & 0 & -s^2 \\ 2s^3+11s^2+16s+3 & 0 & 1 & 0 \\ s^2+5s+6 & 0 & 0 & 1 \end{array}\right]$$

Iteration 2: "operator": s^2+5s+6

"operand": $-s^3-s^2+4s+3$

$q = -s^3/s^2=-s$

$$B = \left[\begin{array}{c|ccc} 3s^2+10s+3 & 1 & 0 & -s^2+s \\ 2s^3+11s^2+16s+3 & 0 & 1 & 0 \\ s^2+5s+6 & 0 & 0 & 1 \end{array}\right]$$

Iteration 3: "operator": s^2+5s+6

"operand": $2s^3+11s^2+16s+3$

$q=2s^3/s^2=2s$

$$B = \left[\begin{array}{c|ccc} 3s^2+10s+3 & 1 & 0 & -s^2+s \\ s^2+4s+3 & 0 & 1 & -2s \\ s^2+5s+6 & 0 & 0 & 1 \end{array}\right]$$

Iteration 4: "operator": s^2+4s+3

"operand": $3s^2+10s+3$

$q=3s^2/s^2=3$

$$B = \left[\begin{array}{c|ccc} -2s-6 & 1 & -3 & -s^2+7s \\ s^2+4s+3 & 0 & 1 & -2s \\ s^2+5s+6 & 0 & 0 & 1 \end{array}\right]$$

Iteration 5: "operator: $-2s-6$

"operand": s^2-4s+3

$q=s^2/-2s=-1/2s$

$$B = \left[\begin{array}{c|ccc} -2s-6 & 1 & -3 & -s^2+7s \\ s+3 & s/2 & 1-3/2s & -2s-\frac{s^3}{2}+\frac{7s^3}{2} \\ s^2+5s+6 & 0 & 0 & 1 \end{array}\right]$$

Iteration 6: "operator": $s+3$

"operand": s^2+5s+6

$q=s^2/s=s$

$$B = \left[\begin{array}{c|ccc} -2s-6 & 1 & -3 & -s^2+7s \\ s+3 & s/2 & 1-3/2s & -2s-\frac{s^3}{2}+\frac{7s^2}{2} \\ 2s+6 & -s2/2 & -s+\frac{3s^2}{2} & 1+2s^2+\frac{s^4}{2}-\frac{7s^3}{2} \end{array}\right]$$

Iteration 7: "operator": s+3

"operand": -2s+6

$$B = \left[\begin{array}{c|ccc} 0 & 1+s & -1-3s & -s^3+6s^2+3s \\ s+3 & s/2 & 1-3/2s & -2s-\frac{s^3}{2}+\frac{7s^2}{2} \\ 2s+6 & -s^2/2 & -s+3s^2/2 & 1+2s^2+\frac{s^4}{2}--\frac{7s^3}{2} \end{array}\right]$$

Iteration 8: "operator": s+3

"operand": 2s+6

$$B = \left[\begin{array}{c|ccc} 0 & 1+s & -1-3s & -s^3+6s^2+3s \\ s+3 & s/2 & 1-3/2s & -2s-\frac{s^3}{2}+\frac{7s^2}{2} \\ 0 & -s^2/2-s & \frac{3s^2}{2}+2s-2 & s^3-\frac{7s^3}{2}-5s^2+\frac{s^4}{2}+4s+1 \end{array}\right]$$

Therefore the g.c.d. of the set Φ is s+3. At the same time we have obtained a set of multipliers, that is,

$$x_1(s)=s/2, \quad x_2(s)=1-\frac{3}{2}s, \quad x_3(s)=-2s-\frac{s^3}{2}+\frac{7s^2}{2}$$

and we have,

$$s+3 = p_1(s)x_1(s)+p_2(s)x_2(s)+p_3(s)x_3(s).$$

■

Implementation of Blankiship's algorithm

In order to implement Blankiship's method to a computer, we must represent the polynomials by vectors consisting of its coefficients. The entries of the unit matrix must be replaced too by corresponding vectors e.g. when the maximum degree of the polynomials is 4, entry 1 of the unit matrix will be replaced by the vector $(0,0,0,0,1)^t$ that represents the polynomial $0\cdot s^4+0\cdot s^3+0\cdot s^2+0\cdot s+1\cdot s^0$. Consequently, if the number of given polynomial is n, matrix B will have the form $B=[B_1|B_2|\cdots|B_n]$ where submatrix B_1 represents the coefficients of the polynomials and submatrices $B_2,\ldots,B_n$ represent the corresponding unit matrix. The steps of the algorithm will now become:

STEP 1: For a given set of n polynomials

Construct matrix $B=[B_1|B_2|\cdots|B_n]$

STEP 2: **if** the first entries of some rows = 0 **then**

apply shifting to these rows transferring

the zeros to the end of them.

if the last columns of all submatrices$\{B_i, i=1,2,\ldots,n\}=0$ **then**

delete them

STEP 3: Select the row with the smallest nonzero leader and

call it the "operator"

STEP 4: Select any other row with a nonzero leader and call it

the "operand".

if there is no such row **then**

the remaining one with nonzero leader

gives the coefficients of the g.c.d. and

of the multipliers.

quit

STEP 5: Divide the leader of the operator into the

leader of the operand, ignoring the remainder.

Denote the quotient by q.

STEP 6: Subtract q times the operator from the operand,

recording the result as a new row and striking out

the operand.

Repeat **STEP 2**.

Alg: 6.8

Example (6.14): Find the g.c.d. of the set

$$p_1(s) = s^4+4s^3+4s^2+4s+3$$

$$p_2(s) = s^3+2s^2+s+2$$

Matrix B has the form $B = \left[\begin{array}{ccccc|ccccc|ccccc} 1 & 4 & 4 & 4 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$

B1 B2 B3

We apply shifting to the second row and

$$B = \left[\begin{array}{ccccc|ccccc|ccccc} 1 & 4 & 4 & 4 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \begin{array}{l} \longrightarrow \text{row 1} \\ \longrightarrow \text{row 2} \end{array}$$

Iteration 1: "operand": row 1

"operator": row 2

$q = 1/1 = 1$

$$B = \left[\begin{array}{ccccc|ccccc|ccccc} 0 & 2 & 3 & 2 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \xrightarrow[\text{row 1}]{\text{shifting to}}$$

$$B = \left[\begin{array}{ccccc|ccccc|ccccc} 2 & 3 & 2 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

Delete the zero columns of submatrices B1, B2, B3

$$B = \left[\begin{array}{cccc|cccc|cccc} 2 & 3 & 2 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \longrightarrow \text{row 1} \\ \longrightarrow \text{row 2} \end{array}$$

Iteration 2: "operand": row 1

"operator": row 2

$q = 2/1 = 2$

$$B = \left[\begin{array}{cccc|cccc|cccc} 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -2 \\ 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \text{shifting} \\ \longrightarrow \\ \text{to row 1} \end{array}$$

$$B = \left[\begin{array}{cccc|cccc|cccc} -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -2 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \longrightarrow \text{ row 1} \\ \longrightarrow \text{ row 2} \end{array}$$

Iteration 3: "operand": row 2

"operator": row 1

q = 1/-1 = -1

$$B = \left[\begin{array}{cccc|cccc|cccc} -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -2 & 0 \\ 0 & 2 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 & -2 & 1 \end{array}\right] \begin{array}{l} \text{shifting} \\ \longrightarrow \\ \text{to row 2} \end{array}$$

$$B = \left[\begin{array}{cccc|cccc|cccc} -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -2 & 0 \\ 2 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 & -2 & 1 & 0 \end{array}\right]$$

Delete the zero columns of submatrices B1, B2, B3

$$B = \left[\begin{array}{ccc|ccc|ccc} -1 & 0 & -1 & 0 & 0 & 1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & -4 & -3 \end{array}\right] \begin{array}{l} \longrightarrow \text{ row 1} \\ \longrightarrow \text{ row 2} \end{array}$$

The method terminates and row1 contains the coefficient of the g.c.d. and of the multipliers too. The g.c.d. is $-s^2-1$ or s^2+1 and the multipliers are $x_1(s)=-1$ and $x_2(s)=s+2$. It holds that $s^2+1=p_1(s)x_1(s)+p_2(s)x_2(s)$.

■

The computational complexity of algorithm **BLANKISHIP** can be found in [Pac. & Bar., 1].

6.5 REMARKS-DISCUSSION

The most important methods for the computation of the g.c.d. of polynomials were presented. The methods mentioned were divided into two main categories.

The first category was including methods related to Euclid's algorithm. In Euclid's algorithm and its generalizations (Appendix C), we constructed a sequence of polynomials of successively smaller degrees. Unfortunately, as polynomials decrease in degree, their coefficients tend to grow, so the successive steps tend to become harder as the calculation progresses. If the g.c.d. of these inflated coefficients are required, the problem is aggravated. If the coefficient domain is not a field, this same remark applies to any g.c.d.'s of numerators and denominators that are required to simplify the inflated coefficients. If coefficients in a field are not simplified, the division steps become harder faster (as illustrated in Example (C.6)), and the final result, though formally correct, may be practically useless.

The first major advance in controlling the phenomenon of coefficient growth was the discovery by Collins [Col., 1] of an algorithm that effectively controlled coefficient growth without any g.c.d. computations in the coefficient domain.

Routh's algorithm for computing the g.c.d. was another variation of Euclid's algorithm. This was due to the fact that the rows in Routh's array are successively remainders when the $(n-1)^{th}$ row is divided by $(n-2)^{th}$ row. The extension of Routh's algorithm to unique factorization domain highlighted the problem of rapid coefficient growth and a scheme for tackling this was given in section C.2 of Appendix C . An alternative array scheme can be also found in [Bar., 3].

The classical Routh's array has the drawback that it involves divisions. Hence if one starts with integer polynomial entries one ends up with rational numbers or rational functions. If Routh's array is modified so as to avoid divisions then the elements of the array grow very fast and storage problems might be encountered. A solution might be to divide each row by its g.c.d., but finding the g.c.d. is again time consuming. The optimal fraction free Routh array presented in section C.2 of Appendix C avoids rapid coefficient growth without computing the g.c.d.

The second category was including several matrix methods. Companion matrices were introduced and a solution to the g.c.d. problem was given in terms of these matrices.

Sylvester's matrix was also employed in the g.c.d. calculations and the well known Sylvester's Resultant played an important role in the derivation of the g.c.d.

Finally Blankinship's method for integers was generalized to deal with polynomials.

A detailed comparison concerning the computational complexity of some of the above methods (Routh array, Companion matrix method, Blankiship's method) can be found in [Pac. & Bar., 1]. This comparison yielded that the method of Blankiship is preferred over all others since it combines both the calculation of g.c.d. and multipliers for two or more polynomials. When only the g.c.d. is required, the method of Routh is somewhat faster.

Other methods for the calculation of the g.c.d. are developed in [Wein., 1], [Akr., 1]. The method of [Akr., 1] works with integer-preserving arithmetic.

In the following Chapter a completely new method for the evaluation of the g.c.d. of polynomials will be described.

6.6 CONCLUSIONS

The aim of this Chapter was to provide a comprehensive survey of the most important methods for calculating the g.c.d. of polynomials. Due to the extremely wide range of applications requiring the computation of the g.c.d. of polynomials, the present survey can help as index for finding the suitable numerical method wanted each time. Therefore, this Chapter serves the following purposes:

(i) It provides a survey of the most important numerical methods achieving the computation of g.c.d. of polynomials using mostly extensions and variations of the Euclid's algorithm. All these methods are suitable and convenient for the evaluation of g.c.d. of two polynomials.

(ii) It provides a survey of the most important numerical methods attaining the evaluation of g.c.d. of polynomials using matrices. Basically, these methods are applicable when the g.c.d. of several polynomials is required.

CHAPTER 7

A NEW NUMERICAL METHOD FOR THE COMPUTATION OF THE GREATEST COMMON DIVISOR OF POLYNOMIALS

7.1 INTRODUCTION

The problem of finding the greatest common divisor (g.c.d.) of a set of m polynomials of R[s], of maximal degree d, $P_{m,d}$,has attracted a lot of attention (Chapter 6) and has widespread applications in linear systems,network theory etc. [Pac. & Bar., 1]. Most of the procedures for finding the g.c.d. of a $P_{m,d}$ involve the use of some type of generalized resultant test [Bar., 2], [Bar., 5].

The aim of the present Chapter is to provide a new numerical method for the computation of the g.c.d. of a given $P_{m,d}$. This method, is based on a theoretical algorithm suggested in [Kar., 1] which establishes that the g.c.d. of a given $P_{m,d}$ is invariant under the combined action of Extended-R-Equivalent (E-R-E) and shifting operations. Briefly, this theoretical algorithm starts by selecting a base $P_{r,d}$ of $P_{m,d}$, $r \leq m$. By applying successively E-R-E and shifting transformations on the basis matrix $P_r \epsilon R^{rx(d+1)}$ of $P_{r,d}$, the rank is successively reduced and finally leads to a matrix $P_{r'}$ with rank one; any non zero row of the unit rank matrix defines the coefficients of the g.c.d. of the given set $P_{m,d}$.

When this theoretical algorithm is to be implemented as a numerical procedure, extra care must be given to avoid the numerical difficulties arising from the fact that the existence of a nontrivial g.c.d. of $P_{m,d}$ is a nongeneric property. Special numerical procedures are needed to avoid the introduction of additional numerical errors and help in the "catching up" of approximate solutions. It should be emphasised that if measures to define approximate solutions are not incorporated in the algorithm, then almost always, due to numerical errors and the nongeneric nature of the g.c.d. , the answer to this search will be that the g.c.d. is one. A number of concrete issues should be addressed in defining the g.c.d. algorithm, in order to avoid the problems raised above. More specifically:

(i) The selection of a base for the row space of P_m should be performed on the existing data by avoiding any transformations that may corrupt the original data. Defining an orthogonal basis by using standard procedures is avoided for the above reasons. A selection of the best uncorrupted base for the row space of P_m using the tools developed in Chapter 5 is applied . This procedure chooses amongst the existing elements of $P_{m,d}$ and without transforming them, a most orthogonal linearly independent subset $P_{r,d}$ which forms an uncorrupted base for $P_{m,d}$.

(ii) Instead of unstable E-R-E transformations that are exclusively used in the theoretical procedure, Gaussian transformations and shifting are applied

successively on the basis matrix $P_r \epsilon R^{rx(d+1)}$ of $P_{r,d}$, producing each time a new matrix $P_z \epsilon R^{zx(d+1)}, z \leq r$. The rank of the matrix P_z should be computed in a numerically sensible way; we use the notion of the numerical ε-rank defined in Chapter 5 . In order to check the unit rank property of P_z and thus terminate the algorithm, we use the notion of strongly ε-dependent sets of vectors and a singular value based test is deployed. This test is based on the properties of singular values of normalized matrices and states that almost unit rank is achieved, when the maximal singular value of the normalized P_z is equal to $\sqrt{z}$.

The latter test is essential in the termination of the algorithm and the derivation of the approximate solution; otherwise, we will always end up with a conclusion that the original set of polynomials is coprime.

We must notice the fact that this method evaluates always satisfactory approximate solutions according to specified accuracies and that it can be applied to any set $P_{m,d}$ without any restrictions to the sizes of m,d. (The more polynomials we have the quicker it converges.) Also by the selection of an uncorrupted base we achieve a serious reduction in the number of the original data required for the process.

7.2 A THEORETICAL ALGORITHM FOR COMPUTING THE G.C.D. OF POLYNOMIALS [Kar.,1]

7.2.1 Extended-R-Equivalence on sets of polynomials

Let $P_{m,d} \equiv \{p_i(s): p_i(s) \epsilon R[s], i \epsilon \underline{m}, d_i = \deg\{p_i(s)\}, d = \max\{d_i, i \epsilon \underline{m}\}\}$ be the set of polynomials of R[s]; m,d will be referred to as the dimension, degree of $P_{m,d}$ respectively. We define the sets:

$$\{P_d\} \equiv \{P_{m_i,d}, m_i \epsilon Z^+, d \epsilon Z^+ \text{ fixed }\}$$

and

$$\langle P_d \rangle \equiv \{P_{m_i,d'}, m_i \epsilon Z^+, d' \leq d, d \epsilon Z^+ \text{ fixed }\}.$$

$\{P_d\}, \langle P_d \rangle$ are the sets of all polynomial sets from R[s] of degree d, maximal degree d respectively.

For a $P_{m,d} \epsilon \{P_d\}$, the polynomial vector

$$\underline{P}_m(s) \equiv \begin{bmatrix} p_1(s) \\ \vdots \\ p_m(s) \end{bmatrix} = P_m \underline{e}_d(s), \underline{e}_d(s) = [1, s, \ldots, s^d]^t, P_m \epsilon R^{mx(d+1)} \tag{7.1}$$

will be referred to as a vector representative (v.r.) and the matrix P_m as a basis matrix (b.m.) of $P_{m,d}$. P_m, or $\underline{P}_m(s)$ uniquely

defines $P_{m,d}$ up to permutation of its elements.We may always express P_m as

$$P_m=[0_{m,c};\underline{P}_{c+1},\ldots,\underline{P}_{d+1}],\ \underline{P}_i\epsilon R^m,\ i\epsilon\underline{m},\ \underline{P}_{c+1}\neq 0 \tag{7.2}$$

where $c\epsilon\{0,1,2,\ldots\}$. The integer c will be referred to as the <u>order</u> of $P_{m,d}$ and clearly defines the degree of the elementary divisor (e.d.) at s=0 in a greatest common divisor (g.c.d.) of $P_{m,d}$ The set $P_{m,d}$ will be called <u>proper</u>, <u>nonproper</u>, if $c=0, c\epsilon Z^+$ respectively.

Let $P_{m,d}\epsilon\{P_d\}, m^*\epsilon Z^+, m^*\geq m$. An m^*-<u>description</u>, $P_{m^*,d}$, of $P_{m,d}$ is defined by trivially expanding $P_{m,d}$ by m^*-m zeros; the corresponding basis matrix P_{m^*} will be referred to as an m^*-<u>extension</u> of P_m and it has the form

$$P_{m^*} \equiv \left[\begin{array}{c} P_m \\ \hline 0_{m^*-m,d+1} \end{array}\right] \epsilon R^{m^*x(d+1)} \tag{7.3}$$

<u>Definition (7.1):</u> Let $P^1_{m_1,d}, P^2_{m_2,d}\ \epsilon\{P_d\}$, $m^*=\max\{m_1,m_2\}$ and let $P^i_{m^*}$, i=1,2 be the m^*-extensions of the $P^i_{m_i}$, i=1,2 basis matrices. The sets $P^1_{m_1,d}$, $P^2_{m_2,d}$ will be called <u>extended-R-equivalent</u> (E-R-E) and shall be denoted by $P^1_{m_1,d}EP^2_{m_2,d}$, if there is a $Q\epsilon R^{m^*xm^*}$, $|Q|\neq 0$, such that

$$P^2_{m^*} = QP^1_{m^*} \tag{7.4}$$

If Q is orthogonal, then we write $P^1_{m_1,d}EnP^2_{m_2,d}$ and $P^1_{m_1,d}, P^2_{m_2,d}$ will be called <u>normally extended-R-equivalent</u> (NE-R-E).

■

It is readily shown that $E(En)$ is an equivalence relation on $\{P_d\}$ and the corresponding equivalence class of $P_{m,d}\epsilon\{P_d\}$ will be denoted by $E(P_{m,d})$. The characterisation of $E(P_{m,d})$ by invariants is examined next.

Lemma (7.1) : Let $P_m \epsilon R^{mx(d+1)}$ and $\rho(P_m) = r \leq \min\{m,d+1\}$. There exists $Q \epsilon R^{mxm}, |Q| \neq 0$, such that

$$P_m^H \equiv QP_m = \begin{bmatrix} P_r^H \\ \hline 0_{m-r,d+1} \end{bmatrix}, \quad P_r^H \epsilon R^{rx(d+1)} \tag{7.5}$$

where $P_r^H = (h_{ij})$ is a matrix having the following properties:

(i) $\rho(P_r^H) = r$.

(ii) There is a sequence of integers $n_1,\ldots,n_r$, $1 \leq n_1 < n_2 < \ldots < n_r \leq d+1$ such that $h_{ij}=0, j=1,\ldots n_i-1, h_{in_i}=1, i=1,\ldots,r$ and $h_{tn_i}=0$, $t=1,\ldots,i-1,i+1,\ldots,r$.

(iii) The rest of $h_{ij} \epsilon R$ and they are uniquely defined.

■

P_m^H is known as the Left-Hermite-Form (LHF) and P_r^H will be referred to as the Left-Echelon-Form (LEF) of P_m. More explicitly,

$$P_r^H = \begin{array}{c} \quad\quad n_1 \quad\quad\quad n_2 \quad\quad\quad n_r \\ \quad\quad\quad\downarrow \quad\quad\quad\downarrow \quad\quad\quad\downarrow \\ \begin{bmatrix} 0\ldots0\ 1\ *\ldots*\ 0\ *\ldots*\ 0\ *\ldots* \\ 0\ldots0\ 0\ 0\ldots0\ 1\ *\ldots*\ 0\ *\ldots* \\ 0\ldots0\ 0\ 0\ldots0\ 0\ *\ldots*\ 0\ *\ldots* \\ \vdots \\ 0\ldots0\ 0\ 0\ldots0\ 0\ 0\ldots0\ 1\ *\ldots* \end{bmatrix} \end{array} \tag{7.6}$$

where * denotes unspecified entries. The uniqueness of P_r^H provides the means for the characterization of $E(P_{m,d})$ by invariants.

Theorem (7.1): Let $P_{m,d} \epsilon \{P_d\}$, P_m a b.m. and let P_r^H be the LEF of P_m. P_r^H is a complete invariant of $E(P_{m,d})$.

■

Remark (7.1): An equivalent, complete invariant for $E(P_{m,d})$, is defined by the space $R(P_m) \equiv$ row-span$\{P_m\}$.

■

Remark (7.2): The set of polynomials $P^H_{r,d} \epsilon \{P_d\}$ defined by the b.m. P^H_r (the LEF), is a canonical form for $E(P_{m,d})$.

■

Corollary (7.1): Let $P_{m,d}=\{p_i(s), i\epsilon \underline{m}\}\epsilon\{P_d\}$ and let $P'_{r,d}=\{p_j(s), j\epsilon \underline{r}\}$ be a subset of $P_{m,d}$. $P'_{r,d} E P_{m,d}$, if and only if

$$\text{row-span}\{P'_r\} = \text{row-span}\{P_m\} \tag{7.7}$$

■

An important function defined on any $P_{m,d}\epsilon\{P_d\}$ is the g.c.d. The effect of E-R-E transformations on the g.c.d. of $P_{m,d}$ is examined next.

Proposition (7.1): Let $P_{m,d}\epsilon\{P_d\}$ and let $\varphi(s)$ be a g.c.d. of $P_{m,d}$. Then, $\varphi(s)$ is an invariant of $E(P_{m,d})$.

■

The invariance of the g.c.d. under E-R-E transformations suggests that any set in $E(P_{m,d})$ may be used for its computation.

The canonical set $P^H_{r,d}$ has least dimension and the simplest structure. This set will be used for the computation of the g.c.d. of $P_{m,d}$. We first note:

Remark (7.3): The canonical set $P^H_{r,d}$ of $E(P_{m,d})$ may be computed from any subset $P'_{r,d}$ of $P_{r,d}$ which satisfies condition (7.8) of Corollary (7.1). Furthermore, any subset of $P_{m,d}$ which satisfies condition (7.8) has the same g.c.d. with $P_{m,d}$.

■

7.2.3 The g.c.d. of $P_{m,d}$ and the shifting operation

Let $P^H_{r,d}$ be the canonical form of $E(P_{m,d})$. Then the v.r. of $P^H_{r,d}$ is of the following form:

$$\underline{P}^{H}_{r}(s) \equiv \begin{bmatrix} & & n_1 & & & & n_2 & & & & n_r & & \\ & & \downarrow & & & & \downarrow & & & & \downarrow & & \\ & & 1 & & 1 & & 2 & & r-1 & & r & & r \\ 0\ldots0 & 1 & a_{11} & \ldots & a_{1k_1} & 0 & a_{11} & \ldots & a_{1k_{r-1}} & 0 & a_{11} & \ldots & a_{1k_r} \\ 0\ldots0 & 0 & 0 & \ldots & 0 & 1 & a_{21} & \ldots & a_{2k_{r-1}} & 0 & a_{21} & \ldots & a_{2k_r} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0\ldots0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 1 & a_{r1} & \ldots & a_{rk_r} \end{bmatrix} \underline{e}_d(s)$$

$$= P^{H}_{r}\cdot\underline{e}_d(s) = [0_{r,n_1-1}:\hat{P}^{H}_{r}]\cdot\underline{e}_d(s) = s^{n_1-1}\,\hat{P}^{H}_{r}\cdot e_{d'}(s) =$$

$$= s^{n_1-1}\,\hat{\underline{P}}^{H}_{r}(s),\ d'=d-n_1+1 \qquad (7.8)$$

Clearly, the order of $P_{m,d}$ is $c=n_1-1$ and the set $\hat{P}^{H}_{r,d}$, defined by $\hat{\underline{P}}^{H}_{r}(s)$, or $\hat{P}^{H}_{r}$ will be referred to as the <u>reduced canonical set</u> of $E(P_{m,d})$.

<u>Remark (7.4)</u>: Let $\varphi(s)$, $\hat{\varphi}(s)$ be the g.c.d. of $P_{m,d}$, $\hat{P}^{H}_{r,d}$, respectively. Then $\varphi(s) = cs^{n_1-1}\cdot\hat{\varphi}(s)$, $c\in R-\{0\}$ and $\hat{\varphi}(0)\neq 0$. ■

Let $\hat{P}^{H}_{r,d'}=\{t_i(s), i\in\underline{r}\}$ where $t_i(s)$ are the coordinates of $\hat{\underline{P}}^{H}_{r}(s)$. By eqn. (7.8) it is clear that $t_i(s)=s^{n_i-n_1}\,\bar{t}_i(s), i\in\underline{r}$ and the set of polynomials $\bar{P}_{r,d'} \equiv \{\bar{t}_i(s), i\in\underline{r}\}$ is also an invariant of $E(P_{m,d})$.

Let $\bar{t}(s)=1+a_1s+\ldots+a_\delta s^\delta$ be a minimal degree polynomial of $\bar{P}_{r,d'}$; the minimal degree δ will be referred to as the <u>characteristic</u> of $P_{m,d}$.

<u>Proposition (7.2)</u>: Let $P_{m,d}\in\{P_d\}$ be a proper set of characteristic δ.

(i) If $\varphi(s)$ is a g.c.d. of $P_{m,d}$, then $\deg\varphi(s) \leq \delta$.

(ii) If $\delta = 0$, then $P_{m,d}$ is coprime.

(iii) If P_m is a b.m. of $P_{m,d}$ and $\rho(P_m)=d+1$, then $P_{m,d}$ is coprime. ■

Remark (7.5): Necessary conditions for the g.c.d. of a proper $P_{m,d}$ set to be different than a unit of R[s] are: (i) $\delta\epsilon Z^+$, (ii) $\rho(P_m)\leq d$.

■

If $\rho(P_m)\leq d$, or the strongest condition $\delta\epsilon Z^+$ is satisfied the search for the g.c.d. should continue. We first define the following operation.

Definition (7.2): (i) Let $\underline{a}^t\epsilon R^{1xk}$ be a vector of the following type:

$$\underline{a}^t = [\underbrace{0,\ldots\ldots,0}_{\varepsilon-1},a_0,a_1,\ldots,a_k\ ]\ ,\ a_0\neq 0 \tag{7.9}$$

where $\varepsilon\epsilon Z^+$. ε will be called the index of $\underline{a}^t$ and $\underline{a}^t$ will be referred to as an ε-indexed vector. Let $A\epsilon R^{mxk}$, $\underline{a}_i{}^t$ be the i-th row of A and let ε_i be the corresponding index. Then A will be referred to as an $(\varepsilon_1,\ldots,\varepsilon_m)$-indexed matrix.

(ii) On an ε-indexed vector $\underline{a}^t\epsilon R^{1xk}$ we define the shifting operation shf: $\mathrm{shf}(\underline{a}^t)=\underline{a}'^t$, where

$$\underline{a}'^t = [a_0,a_1,\ldots,a_k,\underbrace{0,\ldots\ldots,0}_{\varepsilon-1}]\ \epsilon R^{1xk} \tag{7.10}$$

If $A\epsilon R^{mxk}$ is $(\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_m)$-indexed, we define the shifting operation shf: $\mathrm{shf}(A)=A'\epsilon R^{mxk}$, where the i-th-rows $\underline{a}_i{}^t,\underline{a}_i{}'^t$ of A,A' respectively are related by: $\underline{a}_i{}'^t=\mathrm{shf}(\underline{a}_i{}^t)$, $i\epsilon\underline{m}$.

(iii) Let $P_{m,d}\epsilon\{P_d\}$ and P_m a corresponding b.m. The shifting operation may be defined on $\{P_d\}$ by: $\mathrm{shf}(P_{m,d})=P'_{m,d}\epsilon<P_d>$, where the b.m. of $P'_{m,d}$ is defined by $P'_m=\mathrm{shf}(P_m)$. The set $P'_{m,d}$ will be referred to as the shifted set of $P_{m,d}$.

■

A set $P_{m,d}$ may always be assumed to be $(\varepsilon_1,\ldots,\varepsilon_m)$-indexed, since every matrix P_m is always indexed by some integers $(\varepsilon_1,\ldots,\varepsilon_m)$. Furthermore, since a b.m. P_m is not uniquely defined we may assume that $0<\varepsilon_1\leq\ldots\leq\varepsilon_m$. Such an ordering of the ε_i's will be

assumed in the following. Note that if $\varepsilon_1>1$ (the $\min\{\varepsilon_1\}$), then $\mathrm{shf}(P_{m,d})$ has degree less than d and thus $\mathrm{shf}(P_{m,d})\epsilon\langle P_d\rangle$.

Theorem (7.2): Let $P_{m,d}\epsilon\{P_d\}$ be $(\varepsilon_1,\ldots,\varepsilon_m)$-indexed, $P'_{m,d}=\mathrm{shf}(P_{m,d})$ and let $\varphi(s),\varphi'(s)$ be the g.c.d.s of $P_{m,d}$, $P'_{m,d}$ respectively. Then

$$\varphi(s) = c\cdot s^{\varepsilon_1-1}\cdot\varphi'(s),\ c\epsilon R-\{0\} \text{ and } \varphi'(0)\neq 0 \qquad (7.11)$$

■

Corollary (7.2): If $P_{m,d}\epsilon\{P_d\}$ is proper, then the g.c.d. of $P_{m,d}$ is invariant under the combined action E-R-E transformations and shifting operations.

■

Remark (7.6): Let $P_{m,d}\epsilon\{P_d\}$ be a nonproper set of order c, $P_{m',d'}\epsilon\langle P_d\rangle$ be any set obtained from $P_{m,d}$ under the combined action of E-R-E transformations and shifting operations and let $\varphi(s),\varphi'(s)$ be the g.c.d.s of $P_{m,d},P_{m',d'}$ respectively. Then,

$$\varphi(s) = a\cdot s^{c}\cdot\varphi'(s),\ a\epsilon R-\{0\} \text{ and } \varphi'(0)\neq 0 \qquad (7.12)$$

■

7.2.4 The computation of the g.c.d. of $P_{m,d}$

The invariance of the g.c.d. of a proper set $P_{m,d}$ under E-R-E and shifting transformations suggests that such transformations may be deployed for the computation of the g.c.d.

Remark (7.7): If $P_{m,d}\epsilon\{P_d\}$ is proper, then the combined action of E-R-E and shifting operations produces proper sets $P_{m',d'}\epsilon\langle P_d\rangle$ with $m'\leq m$ and $d'\leq d$. Thus, the g.c.d. of $P_{m,d}$ and $P_{m',d'}$ are equal (mod. $a\epsilon R-\{0\}$).

■

The result stated next provides an essential step in the computation of a g.c.d. of $P_{m,d}\epsilon\{P_d\}$.

Proposition (7.3): Let $P^{H}_{m,d}\epsilon\{P_d\}$ be a proper set, $P_{r,d}$ its canonical form, $P'^{H}_{r,d} = shf(P_{m,d})$ and let $t(s)=1+a_1s+\ldots+a_\delta s^\delta$ be a least degree polynomial of $P'^{H}_{r,d}$. By the combined action of E-R-E and shifting operations $P^{H}_{r,d}$ may be reduced to a set $P^{*}_{r,d}\epsilon\langle P_d\rangle$ with the following properties:

(i) $P^{*}_{r,d}$ is proper and its degree is δ.

(ii) The g.c.d.s of $P^{*}_{r,d}$ and $P_{m,d}$ are equal (mod. $a\epsilon R-\{0\}$).

(iii) A b.m. P^{*}_{r} of $P^{*}_{r,d}$ has the form

$$P^{*}_{r} \equiv \begin{bmatrix} 1 & a_1 \ldots a_\delta & 0 \ldots 0 \\ * & * \ldots * & 0 \ldots 0 \\ . & . \quad . & . \quad . \\ . & . \quad . & . \quad . \\ . & . \quad . & . \quad . \\ * & . \ldots * & 0 \ldots 0 \end{bmatrix} = [\hat{P}^{*}_{r} | 0_{r,d-\delta}] \epsilon R^{rx(d+1)} \qquad (7.13)$$

■

The set $P^{*}_{r,\delta}$ defined by the $\hat{P}^{*}_{r}\epsilon R^{rx(\delta+1)}$ b.m. will be called a δ-reduced set of $P^{H}_{r,d}$ and has degree δ. The simpler set $P^{*}_{r,\delta}$ may then be used for the computation of the g.c.d. of $P_{m,d}$.

Remark (7.8): Let $P^{*}_{r,\delta}$ be a δ-reduced set of $P^{H}_{r,d}$ and let $\hat{P}^{*}_{r}$ be a b.m. for $P^{*}_{r,\delta}$. If $\rho(\hat{P}^{*}_{r})=1$, then a g.c.d. of $P_{m,d}$ is $t(s)=1+a_1s+\ldots+a_\delta s^\delta$. If $\rho(\hat{P}^{*}_{r})=\delta+1$, then $P_{m,d}$ is coprime.

■

The derivation of $P^{*}_{r,\delta}$ from $P_{m,d}$ completes a cycle of the search for the g.c.d. of $P_{m,d}$. If $\rho(\hat{P}^{*}_{r})\neq 1,\delta+1$, then the search has to continue on $P^{*}_{r,\delta}$. A systematic theoretical procedure for the computation of the g.c.d of $P_{m,d}$ is described next.

Theoretical procedure for the computation of the g.c.d.

Let $P_{m,d}\epsilon\{P_d\}$, $\varphi(s)$ be a g.c.d. of $P_{m,d}$, $P_m\epsilon R^{mx(d+1)}$ be a basis matrix of $P_{m,d}$, $\rho(P_m)=r$ and let $c\geq 0$ be the order of $P_{m,d}$. We may compute $\varphi(s)$ by adopting the following procedure.

(I) Nonproper sets: If $c\geq 1$, then $P_m=[0_{m,c},\bar{P}_m]$, s^c is an e.d. of $\varphi(s)$ at $s=0$ and we may write $\varphi(s)=s^c\cdot\bar{\varphi}(s)$, where $\bar{\varphi}(s)$ is a g.c.d. of the proper set defined by the b.m. $\bar{P}_m$.

(II) Proper sets: If $c=0$, the first column of P_m is nonzero and $\varphi(0)\neq 0$. To compute $\varphi(s)$ for the case of proper sets we follow the next steps:

(a.1) If $r=d+1$, then the set $P_{m,d}$ is coprime.

(a.2) If $r<d+1$, we distinguish the following two cases:

(i) If $r=1$, then any nonzero polynomial in $P_{m,d}$ defines the g.c.d.

(ii) If $r>1$, we define a maximal, linearly independent set of r vectors amongst the rows of P_m and we denote by $P_r\epsilon R^{rx(d+1)}$ the corresponding submatrix of P_m. Clearly, $P_{r,d}, P_{m,d}$ have the same g.c.d (modulo $a\epsilon R-\{0\}$).

(b) Let $r>1$ and $P_{r,d}$ a normal subset of $P_{m,d}$ with b.m. P_r. Compute the LEF P_r^H of P_r and thus the canonical set $P_{r,d}^H$. Let $t(s)=1+a_1s+\ldots+a_\delta s^\delta$ be a minimal degree polynomial of $shf(P_{r,d}^H)$. Then,

(b.1) If $\delta=0$, then the set $P_{m,d}$ is coprime.

(b.2) If $\delta\geq 1$, then $\varphi(s)$ divides $t(s)$. To compute $\varphi(s)$ we distinguish the following two cases:

(i) If $\delta=1,2$, then compute the zeros of $t(s)$ and test whether or not are zeros of a v.r. of $P_{m,d}$.

(ii) If $\delta\geq 3$, then by the combined action of E-R-E transformations and shifting we obtain a δ-reduced set $P_{r,\delta}^*$ of $P_{r,d}^H$ having $t(s)$ as a first element. $P_{r,\delta}^*$ and $P_{m,d}$ have the same g.c.d. (mod. $a\epsilon R-\{0\}$).

(c) $P_{r,d}^*$ is proper and thus for the computation of $\varphi(s)$ repeat the repeat the steps described in (II). ■

The procedure described in (II) corresponds to a cycle of the searching process.Note that if the search does not terminate in a cycle, it produces a set of smaller degree. Because the dimension and the degree of the equivalent sets are reduced within a cycle the search eventually terminates. We may illustrate the above procedure by an example.

Example (7.1): Let $P_{3,4}=\{t_1(s)=s^4+s^3-s-1,\ t_2(s)=s^3+3s^2-s-3,\ t_3(s)=s^4-1\}$. In the following, by $P_1 \longrightarrow P_2$, $P_1^* \longrightarrow P_2^*$ we mean that P_2, P_2^* are obtained from P_1, P_1^* by elementary row operations, shifting respectively. A b.m. of $P_{3,4}$ is defined by

$$P_3 = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 \\ -3 & -1 & 3 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \rho(P_3)=r$$

Since, $r=3<4+1=5$ we compute the LEF P_3^H

$$P_3 \xrightarrow{E} \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 3 & 1 & -3 \end{bmatrix} \xrightarrow{E} \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} = P_3^H$$

The minimal degree polynomial in $shf(P_{3,4}^H)$ is $t(s)=s^2-1$.

Clearly ± 1 are roots of $t_i(s), i=1,2,3$ and thus $\varphi(s)=s^2-1$.

Alternatively, from $P_{3,4}$ we may compute a δ-reduced set as:

$$shf(P_3^H)=\begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{bmatrix} \xrightarrow{E} \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{shf}$$

$$\xrightarrow{shf} \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{E} \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = P_3'$$

P_3' is the b.m. of the 2-reduced set and $t(s)=1-s^2=\varphi(s)$.

■

Remark (7.9): Generally, the transformations on the b.m. stop when all the rows of the matrix become linear dependent with the row that contains the minimal degree polynomial.

■

7.3 NUMERICAL PROBLEMS OF THE THEORETICAL ALGORITHM AND THEIR SOLUTION

The nongeneric nature of a nontrivial g.c.d. of a set of polynomials necessitates the need for extra care in trying to formulate a numerical procedure for its computation. In order to implement the main steps of the algorithm in [Kar., 1], in an effective numerical manner, that catches the exact degree of the g.c.d. and obtains a satisfactory approximate solution, we have to resolve a number of issues related to the theoretical algorithm. Some important problems that arise are:

(P1) For a given set $P_{m,d}$ we must choose an uncorrupted base $P_{r,d}$. By the term "uncorrupted" we mean that we want to find a base for $P_{m,d}$ without transforming the original data and evidently introducing roundoff-error even before the method starts. Already known techniques for the evaluation of bases, based on Gram-Schmidt orthogonalization are excluded, because they transform the original data.

(P2) Instead of using E-R-E transformations which generally are numerically unstable we should apply for the same purpose an appropriate stable numerical method, which leaves invariant the row-space of the basis matrix. This numerical method must also retain the zero entries of the b.m. obtained at each iteration. The preservation of this zero structure helps us in reducing the degree of the polynomials after each iteration. Therefore, orthogonal techniques such as Householder transformations cannot be used because they spoil the zero structure of the matrix.

(P3) We must search for an effective criterion that corresponds to the theoretical condition $\rho(P_r)=1$. This criterion must allow the termination of the method only when the exact degree is found, thus permitting the method to provide a satisfactory approximate solution; otherwise, the algorithm may converge to the generic solution, which is that of coprime polynomials.

We consider next the solution of the previously stated numerical problems.

(P1) Selection of a best basis from an existing set

If $\{\underline{x}_i, i\underline{e}m\}$ is a set of vectors of R^n, $X = sp\{\underline{x}_i, i\underline{e}m\}$ and dim $X < m$, then the selection of a basis for X is a problem that may be handled by the Gram-Schmidt orthogonalization procedure, or use of the Singular Value Decomposition. Such procedures yield orthogonal bases, but transform the original data. The nongeneric existence of g.c.d., necessitates the minimization of all unnecessary round off errors and thus leads to the problem of selecting the "best" (according to some normality criterion) basis from the existing vectors of $\{\underline{x}_i, i\underline{e}m\}$. This problem may be referred to as selection of the best uncorrupted basis. The technique proposed in section 5.7.2 of Chapter 5 can be used. More specifically, algorithm **UNCBAS** developed in section 5.7.3 can be applied for the selection of the best uncorrupted basis $P_{r,d}$ of $P_{m,d}$.

(P2) Stable row operations preserving specified zero entries

We apply successively Gaussian elimination with partial pivoting which leaves invariant the row space of a matrix. Then, an L-U factorization of the b.m. P_r of $P_{r,d}$ is obtained and an upper triangular, or trapezoidal form is used instead of the LEF. Due to the fact that Gaussian elimination zeroes the linearly dependent rows of a matrix (except one of them), after each iteration zero rows might appear to b.m. P_r. In such a case all this zero rows must be dismissed. Gaussian elimination with partial pivoting has also the property that leaves invariant the first row of the matrix. Taking advantage of this property (and having in mind that the degree of the g.c.d. of $P_{r,d}$ will be less than or equal to the degree of the lowest degree polynomial), in the beginning of each iteration we reorder the polynomials of $P_{r,d}$ and set always in the first row of b.m. P_r the polynomial of lowest degree. By doing this we preserve all the zero entries of the b.m. obtained at each iteration. Consequently, a serious reduction to the degrees of all remaining polynomials is achieved, and the rank of b.m. P_r will be substantially decreased after a finite number of steps.

(P3) The termination criterion

In Sect. 7.2, it was noticed that the theoretical algorithm will always converge after a finite number of steps. Consequently, the corresponding numerical method will converge after a finite number of iterations. The

main issue now is that we have to introduce a specific criterion that will allow the termination of the method, when the exact degree of the g.c.d. is found. From Definition (5.3) and Remark (5.3) developed in Chapter 5, we conclude that the theoretical condition $\rho(P_r)=1$ which specifies the termination of the theoretical algorithm, numerically can be applied as $\rho_\varepsilon(P_r)=1$ for a given accuracy ε. Thus, we can formulate a criterion for the termination of the method as follows:

Criterion (I) : For a given $P_{m,d}$ we start the method with a set $A=\{\underline{P}_1{}^t,\underline{P}_2{}^t,\ldots,\underline{P}_r{}^t\}$ of r linearly independent vectors $\underline{P}_i{}^t\epsilon R^{1x(d+1)}$, $i=1,2,\ldots,r$. By applying successively Gaussian transformations and shifting we produce each time sets $A_G=\{\underline{Q}_1{}^t,\underline{Q}_2{}^t,\ldots,\underline{Q}_z{}^t\}$ with $z\leq r$. After a number of iteration cycles, the process terminates when we obtain a set A_G which is strongly ε-dependent for a given tolerance ε. i.e. it has $\rho_\varepsilon(A_G)=1$.

■

Criterion (I) has the feature that is closely connected with a given tolerance ε. It is evident that different sets $P_{m,d}$ may require different values of ε, depending on their coefficients. It is impossible to assess from the beginning the appropriate value of ε for each $P_{m,d}$. Therefore, we have to formulate an improved criterion which is convenient for all sets $P_{m,d}$ without having an explicit dependence on the specified tolerance ε.

Theorems (5.5) and (5.6) formulated in Sect. 5.4 of Chapter 5 form the basis for developing the following specific criterion which allows the termination of the method.

Criterion (II) : For a given $P_{m,d}$ we start the method with a set $A=\{\underline{P}_1{}^t,\underline{P}_2{}^t,\ldots,\underline{P}_r{}^t\}$ of r linearly independent vectors $\underline{P}_i{}^t\epsilon R^{1x(d+1)}$, $i=1,2,\ldots,r$. By applying successively Gaussian transformations and shifting we derive at the end of every cycle sets $A_G=\{\underline{Q}_1{}^t,\underline{Q}_2{}^t,\ldots,\underline{Q}_z{}^t,\underline{0}^t\}$ with $z\leq r$. This process terminates when the maximal singular value σ_1 of the normalization of A_G is approximately equal to $\sqrt{z}$ and all the other singular values vanish.

Remark (7.10): Actually when the set A_G satisfies the condition of Criterion (II) it is a strongly ε-dependent set set for some value of ε.

Remark (7.10): Actually when the set A_G satisfies the condition of Criterion (II) it is a strongly ε-dependent set set for some value of ε. Therefore, Criterion (II) is an improved, more specific and convenient, version of Criterion (I).

■

Remark (7.11): By applying normalization at each new set A_G, we keep the data in reasonable size and consequently our process will never end up with a fuzzy ε-dependent set.

■

Remark (7.12): For some tolerance ε, it is evident that the numerical ε-rank of a normalized set $A_N=\{\underline{v}_1^t,\underline{v}_2^t,\ldots,\underline{v}_z^t\}$ satisfying Criterion (II), is equal to one. Thus, all the vectors of such a set are linearly dependent. If we select any of them, this contains the coefficients of the g.c.d. (mod $a\epsilon R-\{0\}$).

■

A more accurate selection could be achieved if the approach concerning the selection of a "best" representative of a strongly ε-dependent set, developed in Sect. 5.5 of Chapter 5, is applied.

7.4 THE NUMERICAL ALGORITHM OF THE METHOD AND ITS ANALYSIS

7.4.1 The numerical algorithm

To describe our algorithm we assume that $P_{m,d}\ \epsilon\{P_d\}$, $\varphi(s)$ a g.c.d. of $P_{m,d}$, $P_m\ \epsilon R^{mxn}$, n=d+1 a basis matrix of $P_{m,d}$, $\rho(P_m)=r$, $c\geq 0$ the order of $P_{m,d}$, σ_i ,i=1,2,...min{m,n} the singular values of P_m, ε, ε_1 given tolerances.

Algorithm MAIN

STEP 1: **if** P_m is nonproper ($c\geq 1$) **then**

$P_m := [\ 0_{m,c}\ ,\ \bar{P}_m\ ]$

s^c is an e.d. of $\varphi(s)$ at s= 0

$\varphi(s) := s^c\cdot\bar{\varphi}(s)$, where $\bar{\varphi}(s)$ is a g.c.d

of the proper set defined by the b.m. $\bar{P}_m$

else

P_m is proper (c= 0)

The first column of P_m is nonzero

$\varphi(0) \neq 0$

STEP 2: **if** $\rho_\varepsilon(P_m)= n$ **then**

the polynomials are coprime

$\varphi(s) := 1$

quit

else if $\rho_\varepsilon(P_m)= 1$ **then**

any nonzero row of P_m gives the

the coefficients of $\varphi(s)$

quit

if $\rho_\varepsilon(P_m) \neq m$ **then**

find a best uncorrupted basis of $\rho_\varepsilon(P_m)$=r vector

amongst the rows of P_m and $P_r \in R^{rxn}$ $r \leq n$ the

corresponding submatrix of P_m.

$P_m := P_r$

STEP 3: 3.1 Reorder the rows of P_m in decreasing order

according to the number of zeros in them.

3.2 **if** {deg{$p_i(s)$} the same for all i=1,2,...,m} **then**

find $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m$ the singular values of Pm

if ($\sigma_1 \approx \sqrt{m}$ and $\sigma_i \leq \varepsilon$, i=1,2,...,m) **then**

any row of P_m gives the coefficients of $\varphi(s)$

quit

3.3 Apply Gaussian elimination with partial pivoting to

P_m and transform it to an upper trapezoidal or

triangular form P_m^G.

$P_m := P_m^G$

3.4 P_m := the normalization of P_m

3.5 Apply shifting to P_m

3.6 Repeat **STEP 3**

Alg: 7.1

7.4.2 Implementation of the algorithm

For the implementation of the previous algorithm, several subalgorithms are required. Most of them were constructed especially for the needs of the above algorithm. Two basic ones were taken from NAG Library. In the sequel, an analytical description of the most important of them is presented. Their computational complexity is measured by using the concept of flops [Gol., 1]. Whenever necessary, their error analysis is also developed. Useful hints concerning the implementation of the steps of the above algorithm are also given.

(i) **Algorithm PROPER**

Given $A \epsilon R^{mxn}$, the following algorithm checks if A is proper or not.

if $A = [\ O_{m,c}\ ,\ \bar{A}_m\ ]$ **then**

$A := \bar{A}_m$

s^c is an e.d. of $\varphi(s)$

Alg: 7.2

(ii) **Algorithm BSCALE**

It has been observed empirically, that if the elements of the coefficient matrix P_m vary greatly in size, then it is likely that loss of significance

errors will be introduced and that the rounding error propagation will be worse. To avoid this problem, in the beginning of the evaluations we scale matrix P_m, so that its elements vary less. A specific row-scaling, named B-scaling [Atk., 1] which uses the machine's arithmetic base β and does not cause any rounding errors is applied. The following algorithm scales the original matrix using the computer number base β. After the application of this scaling the elements of A satisfy:

$$\beta^{-1} < \max_{1\le j\le n} |a_{ij}| \le 1 \quad , \; i = 1,2,\ldots,m$$

Matrix A is overwritten by the scaled matrix.

```
for  i = 1,2,...,m
     s_i := max |a_ij|
            1≤j≤n
     Determine the smallest integer r_i ∈ Z so β^(r_i) ≥ s_i
     di := 1/β^(r_i)
     for j = 1,2,...,n
         a_ij := a_ij · d_i
```

Alg: 7.3

Remark (7.13): This scaling is extremely useful because no rounding error is involved in defining the new a_{ij}. It only appears a change in the exponent in the floating point form of a_{ij}. ■

(iii) For the evaluation of the singular values of matrix P_m we use subroutine F02WCF of NAG Library. (see section 5.7.3 of Chapter 5).

(iv) For the selection of an uncorrupted basis for the row space of the matrix P_m, algorithm **UNCBAS** of section 5.7.3 is used.

(v) **Algorithm REORD**

Given $A = [\underline{r}_1^t, \underline{r}_2^t, \ldots, \underline{r}_m^t] \epsilon R^{mxm}$ the following algorithm reorders the rows of the matrix, in ascending order according to the number of zeros in them and throws the zero rows (lzero) from it. To each row $\underline{r}_i^t$ we correspond the pair $(metr_i, indrow_i)$ where $metr_i$ counts the zeros of each row and $indrow_i$ contains the index of each row.

```
Order array metr_i in descending order
Each new metr_i retains invariant its previous
row index (indrow_i)
lzero := 0
for  i=1,2,...,m
     if  metr_i := n  then
         lzero := lzero+1
     else
         r_{i-lzero}^t := r_{indrow_i}^t
```

Alg: 7.4

Computational complexity of **REORD** [Kron., 1]

The computational complexity of the algorithm is measured by the number of comparisons between the elements of array $metr_i$ (key comparisons). If we use the simple selection sort this number is :

$$(m-1)+(m-2)+\ldots+2+1 = m\cdot(m-1)/2 \approx O(m^2)$$

Notation (7.1): For large arrays other faster methods can be used for sorting them. (such as Quicksort)

■

(vi) It is evident that only when all the polynomials of a set have the same degree it is likely for their b.m. to have numerical ε-rank one, for some

tolerance ε. In order to avoid unneccessary calculations we check whether Criterion (II) holds true, only when all polynomials in $P_{r,d}$ have the same degree. After testing a variety of examples, it had been observed that when $\sigma_1 \approx \sqrt{m}$ all the rest σ_i become less than some tolerance ε_1. Thus, instead of checking if $\{ \sigma_1 \approx \sqrt{m}$ and $\sigma_i \leq \varepsilon_1$, $i=2,3,\ldots,m \}$ for some ε_1 that we must specify from the beginning, we only check if $\{\sigma_1 \approx \sqrt{m} \}$. When this is achieved, σ_2 determines the value of ε_1. From the examples it seems that the value of ε_1 specifies the number of significant digits in the final result. The condition $\sigma_1 \approx \sqrt{m}$ is translated into $| \sigma_1 - \sqrt{m} | \leq n$, for some accuracy n.

(vii) **Algorithm APRSCA**

Let $A = [\underline{r}_1^t, \underline{r}_2^t, \ldots, \underline{r}_m^t] \in R^{mxn}$ a given matrix. Its first row after the application of algorithm **REORD** contains the maximum number of zeros. We want this first row to be retained in the same position after the application of partial pivoting. So, we scale appropriately each row of matrix A that has first coefficient greater than the first coefficient of the first row.

```
if  a11 < 0   then
    r1^t := -r1^t
    for  i = 2,3,...,m
         if  |ai1| > a11   then
             numbz :=  minimum power of 10 so the
                       integer part of a11·10^numbz > 0
             ri^t := 1/10^numbz · ri^t
```

Alg: 7.5

Error analysis of **APRSCA** [Wilk., 1]

The multiplication of the i-th row of matrix A by an appropriate scale factor (lets say λ) is equivalent by a left multiplication of A by a diagonal matrix D with 1 on diagonal and the scale factor λ in the i-th position. Then,

$$D\cdot A = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & \lambda & \\ & & & \ddots \\ & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots \dots & \vdots \\ \lambda a_{i1} & \lambda a_{i2} \dots & \lambda a_{in} \\ \vdots & \vdots \dots & \vdots \\ a_{m1} & a_{m2} \dots & \lambda a_{mn} \end{bmatrix}$$

$$fl(DA) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \dots & \vdots \\ \lambda a_{i1}(1+\varepsilon_{i1}) & \lambda a_{i2}(1+\varepsilon_{i2}) & \dots & \lambda a_{in}(1+\varepsilon_{in}) \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} =$$

$$D\cdot A + \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \lambda a_{i1}\varepsilon_{i1} & \lambda a_{i2}\varepsilon_{i2} & \dots & \lambda a_{in}\varepsilon_{in} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \quad |\varepsilon_{ij}| \le u \quad j=1,2,\dots,n$$

The relative error is given by the expression:

$$Rel = \frac{| fl(DA) - DA |}{|DA|}$$

We remark that:

$$\text{Rel} \leq u \frac{\begin{vmatrix} 0 & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ |\lambda a_{i1}| & |\lambda a_{i2}| & \cdots & |\lambda a_{in}| \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 0 \end{vmatrix}}{\begin{vmatrix} |a_{11}| & |a_{12}| & \cdots & |a_{1n}| \\ |a_{21}| & |a_{22}| & \cdots & |a_{2n}| \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ |\lambda a_{i1}| & |\lambda a_{i2}| & \cdots & |\lambda a_{in}| \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ |a_{m1}| & |a_{m2}| & \cdots & |a_{mn}| \end{vmatrix}}$$

Thus, the relative error is always very small.

(vii) **Algorithm GAUSS**

In Appendix D, an analytical description of this algorithm is presented. Furthermore, its computational comlpexity and its error analysis are also fully discussed.

(ix) The normalization of the matrix is achieved using algorithm **NORMAL** proposed in section 5.7.2 of Chapter 5. A slight variation can be added. If zero entries exist amongst the rows of the matrix P_m, subroutine **NORMAL** will not take them into account; and therefore the number of flops will be seriously decreased.

(x) The space complexity of the algorithm depends on the size of the given matrix P_m. Basically, it requires an mxn array for storing the original data and all the intermediate transformations are kept on the same array. Some auxiliary arrays of one or two dimensions are also used but they do not occupy much extra storage.

7.4.3 Error Analysis of the algorithm

According to the technique of backward error analysis, we will prove that after the end of the k-th iteration:

(i) $M \cdot P_m^{(k)} = P_m^{(k-1)} + E$, where E is a matrix of small elements which accounts for the rounding errors and M a matrix accounting for the normalization and the Gaussian transformations performed at each step.

(ii) The size of E is properly bounded.

In fact, it is observed that:

(i) During the k-th iteration, the application of Gaussian elimination with partial pivoting in the b.m. $P_m^{(k-1)}$ produces a b.m. $P_m^{G(k)}$ that satisfies [Wilk., 1]:

$$P_m^{G(k)} = L^{-1} \cdot (P^{(k-1)} + E_G), \qquad \| E_G \|_\infty \leq 2.01u \cdot (m-1)^2 \tag{7.14}$$

where $L \epsilon R^{mxm}$ a unit lower triangular matrix with nondiagonal elements less than unity, $E_G \epsilon R^{mxn}$ the error matrix, u the unit round off. (see Appendix D).

As it was proved in section 5.7.3 of Chapter 5, the normalization $P_m^{(k)}$ of $P_m^{G(k)}$ satisfies:

$$P_m^{(k)} = N \cdot P_m^{G(k)} + E_N, \qquad \| E_N \|_\infty \leq 3.003n \cdot u \tag{7.15}$$

where $N \epsilon R^{mxm} = \text{diag}(d_1, d_2, \ldots, d_m)$, $d_i = 1 \Big/ \left(\sum_{j=1}^{n} |P_{m\,ij}^{G(k)}|^2 \right)^{1/2}$, $i = 1,2,\ldots,m$,

$E_N \epsilon R^{mxn}$ the error matrix.

There is no loss of generality in assuming that $P_m^{G(k)} \leq 1$ since this can achieved by scaling (without rounding error), and therefore we can suppose that $\| N^{-1} \|_\infty \leq \sqrt{n}$.

(7.14), (7.15) yields:

$$P_m^{(k)} = N \cdot L^{-1} \cdot (P_m^{(k-1)} + E_G) + E_N \tag{7.16}$$

By setting $(N \cdot L^{-1})^{-1} = M \epsilon R^{mxm}$, (7.16) can be written as:

$$M\cdot P_m^{(k)} = P_m^{(k-1)} + E \tag{7.17}$$

where $E = E_G + M\cdot E_N \in R^{mxn}$ the error matrix.

(ii) From (7.17) we notice that

$$\| E \|_\infty \leq \| E_G \|_\infty + \| M\cdot E_N \|_\infty \tag{7.18}$$

Combining relations (7.14), (7.15) and from the fact that

$$\| L \|_\infty \leq n, \qquad \| M \|_\infty \leq \| L \|_\infty \cdot \| N^{-1} \|_\infty \leq n\cdot\sqrt{n}$$

we conclude that:

$$\| E \|_\infty \leq 2.01\cdot u\cdot(m-1)^2+3.003\cdot u\cdot n^{5/2} \tag{7.19}$$

The previous analysis leads us to the formulation of the following result , which establishes the stability of each iteration step.

Theorem (7.3): The matrix $P_m^{(k)}$, computed by the method in the k-th iteration, using floating-point arithmetic with unit round off u, satisfies the properties:

$$M\cdot P_m^{(k)} = P_m^{(k-1)} + E, \quad \| E \|_\infty \leq 2.01\cdot u\cdot(m-1)^2+3.003\cdot u\cdot n^{5/2} \tag{7.20}$$

■

7.5 NUMERICAL RESULTS - DISCUSSION

The previous numerical method was programmed [Mit. & Kar., 1] and a variety of examples were tested. Next we present some representative ones. Because the computer was operated on a time sharing principle, each example was run several times and the timings averaged. To clarify the algorithm, in the first example all the required intermediate steps are described analytically before the final result.

The following tolerances are required:

ε - specifies the initial numerical ε-rank of P_m

ε_G - specifies the accuracy of Gaussian elimination (values less than ε_G will be set equal to zero during Gaussian elimination)

n - specifies the accuracy of our final result.

Example (7.2) : Find the g.c.d. of the set:

$P_{6,4} = \{ P_1(s) = s^4-15s^3-8s+120,$
$P_2(s) = s^4-8s^3-101s^2-59s-15,$
$P_3(s) = s^4-14s^3-14s^2-14s-15,$
$P_4(s) = 2s^4-9s^3-302s^2-85s-1650,$
$P_5(s) = s^4-36s^3+322s^2-66s-585,$
$P_6(s) = 139s^4-1585s^3-7389s^2-648s-1525 \}$

Using 5 decimal digits and by setting $\varepsilon = \varepsilon_G = n = 10^{-11}$ the following results are obtained.

THE MATRIX IS PROPER

THE GIVEN MATRIX IS

120.00000	-8.00000	.00000	-15.00000	1.00000
-15.00000	-59.00000	-101.00000	-8.00000	1.00000
-15.00000	-14.00000	-14.00000	-14.00000	1.00000
-1650.00000	-85.00000	-302.00000	-9.00000	2.00000
-585.00000	-66.00000	322.00000	-36.00000	1.00000
-15255.00000	-648.00000	-7389.00000	-1585.00000	139.00000

SINGULAR VALUES
.17118E+05
.71733E+03
.11223E+03
.65364E+02
.14080E-11

THE NUMERICAL RANK OF THE MATRIX IS 4

THE ROW INDEPENDENT MATRIX IS

120.00000	-8.00000	.00000	-15.00000	1.00000
-15.00000	-59.00000	-101.00000	-8.00000	1.00000
-15.00000	-14.00000	-14.00000	-14.00000	1.00000
-1650.00000	-85.00000	-302.00000	-9.00000	2.00000

THE B-SCALED MATRIX IS

.23438	-.01563	.00000	-.02930	.00195
-.02930	-.11523	-.19727	-.01563	.00195
-.23438	-.21875	-.21875	-.21875	.01563
-.40283	-.02075	-.07373	-.00220	.00049

WE BEGIN GAUSSIAN TRANSFORMATIONS TO THE MATRIX

STEP 0

MATRIX A(I,J)

.23438	-.01563	.00000	-.02930	.00195
-.02930	-.11523	-.19727	-.01563	.00195
-.23438	-.21875	-.21875	-.21875	.01563
-.40283	-.02075	-.07373	-.00220	.00049

NUMBER OF NONZERO ROWS = 4

THE REORDERED MATRIX IS

-.40283	-.02075	-.07373	-.00220	.00049
.23438	-.01563	.00000	-.02930	.00195
-.02930	-.11523	-.19727	-.01563	.00195
-.23438	-.21875	-.21875	-.21875	.01563

SINGULAR VALUES

.59359E+00
.33044E+00
.12091E+00
.75649E-02

THE APPROPRIATE SCALED MATRIX IS

.40283	.02075	.07373	.00220	-.00049
.23438	-.01563	.00000	-.02930	.00195
-.02930	-.11523	-.19727	-.01563	.00195
-.23438	-.21875	-.21875	-.21875	.01563

THE NORMALIZED MATRIX IS

.98238	.05061	.17981	.00536	-.00119
.00000	-.59374	-.50519	-.62475	.04407
.00000	.00000	-.67355	.73770	-.04619
.00000	.00000	.00000	-.99779	.06652

STEP 1

MATRIX A(I,J)

.98238	.05061	.17981	.00536	-.00119
-.59374	-.50519	-.62475	.04407	.00000
-.67355	.73770	-.04619	.00000	.00000
-.99779	.06652	.00000	.00000	.00000

NUMBER OF NONZERO ROWS = 4

THE REORDERED MATRIX IS

-.99779	.06652	.00000	.00000	.00000
-.67355	.73770	-.04619	.00000	.00000
-.59374	-.50519	-.62475	.04407	.00000
.98238	.05061	.17981	.00536	-.00119

THE APPROPRIATE SCALED MATRIX IS

.99779	-.06652	.00000	.00000	.00000
-.67355	.73770	-.04619	.00000	.00000
-.59374	-.50519	-.62475	.04407	.00000
.98238	.05061	.17981	.00536	-.00119

THE NORMALIZED MATRIX IS

.99779	-.06652	.00000	.00000	.00000
.00000	.99779	-.06652	.00000	.00000
.00000	.00000	-.99779	.06652	.00000
.00000	.00000	.00000	.99779	-.06652

STEP 2

MATRIX A(I,J)

.99779	-.06652	.00000	.00000	.00000
.99779	-.06652	.00000	.00000	.00000
-.99779	.06652	.00000	.00000	.00000
.99779	-.06652	.00000	.00000	.00000

NUMBER OF NONZERO ROWS = 4

THE REORDERED MATRIX IS

.99779	-.06652	.00000	.00000	.00000
.99779	-.06652	.00000	.00000	.00000
-.99779	.06652	.00000	.00000	.00000
.99779	-.06652	.00000	.00000	.00000

SINGULAR VALUES
.20000E+01
.95934E-15
.00000E+00
.00000E+00

WE FIND THE GCD OF THE POLYNOMIALS

-15.00000	1.00000	.00000	.00000	.00000

Therefore the g.c.d. of $P_{6,4}$ is $\varphi(s) = s-15$. The required time was 0.574 sec.

Remark (7.14): A better approximation can be achieved if Remark (5.6) is applied. The S.V.D. of the final reordered matrix M is given by $M=V\Sigma W^T$, where $M=[\underline{m}_1{}^t,\underline{m}_2{}^t,\underline{m}_3{}^t,\underline{m}_4{}^t]$. The matrices are:

MATRIX M

.99779	-.06652	.00000	.00000	.00000
.99779	-.06652	.00000	.00000	.00000
-.99779	.06652	.00000	.00000	.00000
.99779	-.06652	.00000	.00000	.00000

MATRIX V

.50000	.00000	.86603	.00000
.50000	-.15430	-.28868	.80178
-.50000	.61721	.28868	.53452
.50000	.77152	-.28868	-.26726

MATRIX W^T

.99779	-.06652	.00000	.00000	.00000
.06652	.99779	.00000	.00000	.00000
.00000	.00000	-1.00000	.00000	.00000
.00000	.00000	.00000	-1.00000	.00000

Since it is required a "best" representative of the strongly ε-dependent set $\{\underline{m}_1{}^t, \underline{m}_2{}^t, \underline{m}_3{}^t, \underline{m}_4{}^t\}$, this is the singular vector $\underline{w}_1$ =(.99779 -.06652 .0000 .0000 .0000)^t of W.

■

Example (7.3) : Find the g.c.d. of the set:

$$P_{3,5} = \{\ P_1(s) = 2.9s^2+14.85s+15.75,$$
$$P_2(s) = 6.1s^2+11.65s^2+11.85s+12.15,$$
$$P_3(s) = 3.7s^3+17.05s^2+30.35s+19.65\ \}$$

After two iterations the method finds φ(s)=s+1.5. This result was achieved by setting $\varepsilon = \varepsilon_G = 10^{-11}$. The required time was 0.3sec.

■

Example (7.4) : [Pac. & Bar., 1] Find the g.c.d. of the set:

$$P_{2,16} = \{\ P_1(s) = s^{16}-30s^{15}+435s^{14}-4060s^{13}+27337s^{12}-140790s^{11}+$$
$$5731055s^{10}-1877980s^9+4997788s^8-10819380s^7+$$
$$18959460s^6-26570960s^5+29153864s^4-24178800s^3+$$
$$14280000s^2-5360000s+960000,$$
$$P_2(s) = s^{14}-140s^{12}+7462s^{10}-191620s^8+2475473s^6-$$
$$15291640s^2-5401600\ \}$$

After 21 iterations the method finds:
φ(s) ≈ s^4-9.9997s^3+34.99734s^2-49.9923s+23.99288 (the accurate g.c.d. is φ(s) = s^4-10s^3+35s^2-50s+24) This result was achieved by setting $\varepsilon = \varepsilon_G = 10^{-11}$, n = 10^{-5}. The required time was 3.656 sec.

■

Example (7.5) : Find the g.c.d. of the set:

$P_{11,21}$ = { $P_1(s) = s^{20}+4s^{19}+7s^{18}+21s^{17}+54s^{16}+82s^{15}+61s^{14}+29s^{13}+36s^{12}+47s^{11}+26s^{10}+7s^9+15s^8+20s^7+12s^6+6s^5+27s^4+131s^3+286s^2+318s+140$,

$P_2(s) = s^{20}+3s^{19}+4s^{10}+2s^{17}+3s^{14}+9s^{13}+12s^{12}+6s^{11}+5s^{10}+15s^9+22s^8+16s^7+9s^6+7s^5+4s^4+2s^3$,

$P_3(s) = s^{20}+3s^{19}+4s^{18}+2s^{17}+s^{13}+3s^{12}+4s^{11}+2s^{10}+s^6+3s^5+15s^4+35s^3+44s^2+22s$,

$P_4(s) = 5s^{20}+15s^{19}+20s^{18}+10s^{17}+4s^{13}+12s^{12}+16s^{11}+8s^{10}+2s^8+6s^7+8s^6+4s^5+10s^3+30s^2+40s+20$,

$P_5(s) = -s^{20}-3s^{19}-4s^{18}-2s^{17}-s^8-3s^7-4s^6-2s^5+30s^3+90s^2+120s+60$,

$P_6(s) = s^{20}+3s^{19}+4s^{18}+2s^{17}-2s^{16}-6s^{15}-8s^{14}-4s^{13}+s^{12}+3s^{11}+4s^{10}-s^9-9s^8-12s^7-6s^6+11s^3+33s^2+44s+22$,

$P_7(s) = s^{20}+3s^{19}+4s^{18}+2s^{17}+11s^{10}+33s^9+44s^8+22s^7+20s^3+60s^2+80s+40$,

$P_8(s) = s^{20}+3s^{19}+7s^{18}+11s^{17}+12s^{16}+8s^{15}+6s^{14}+8s^{13}+4s^{12}+5s^9+15s^8+20s^7+10s^6+9s^3+27s^2+36s+18$,

$P_9(s) = s^{20}+3s^{19}+4s^{18}+3s^{17}+3s^{16}+4s^{15}+5s^{14}+9s^{13}+13s^{12}+9s^{11}+9s^{10}+17s^9+20s^8+10s^7+s^6+3s^5+4s^4+5s^3+9s^2+12s+6$,

$P_{10}(s) = s^{20}+2s^{19}+s^{18}-2s^{17}-2s^{16}+s^{12}+3s^{11}+4s^{10}+2s^9-s^8-3s^7-4s^6+2s^5-4s^3-12s^2-16s-8$,

$P_{11}(s) = s^{20}+3s^{19}+15s^{18}+35s^{17}+44s^{16}+22s^{15}+3s^{14}+9s^{13}+13s^{12}+9s^{11}+4s^{10}+2s^9+30s^3+90s^2+120s+60$ }

After 5 iterations the method finds:
$\varphi(s) \approx s^3+2.999999999997s^2+3.999999999995s+1.999999999996$ (the accurate g.c.d. is $\varphi(s)=s^3+3s^2+4s+2$). This result was achieved by setting $\varepsilon = 10^{-11}$, $\varepsilon_G = n = 10^{-8}$. The required time was 12.169 sec.

■

The following example demonstrates the "catching up" of approximate solutions when a "best" uncorrupted base is used.

Example (7.6) : Find the g.c.d. of the set

$P_{3,1}$ = { $P_1(s) = s+3$, $P_2(s) = s+2.999$, $P_3(s) = 2s+5.999$ }

We select a most orthogonal uncorrupted base of the above set

$\mathcal{B}_{unc}$ = { $P_1(s)$ = s+3, $P_2(s)$ = s+2.999 }. Then, for n = 10^{-5} the method catches up an approximate g.c.d. equal to s+3.
On the contrary, if we select an orthonormal base
$\mathcal{B}_{orth}$ = { $P_1(s)$ = 0.3162752s+0.948667, $P_2(s)$ = 0.948667s-0.3162752 }
the method ends up with a coprime set of polynomials.

Thus, although in some cases (specially when m is large) the selection of a most orthogonal uncorrupted base compared with the selection of an orthonormal base might seem much more tedius, it provides an efficient technique able to catch up approximate solutions.

■

Remarks

Computationally, the method is attractive because you can easily evaluate the g.c.d. of any $P_{m,d}$; it requires the forming of the basis matrix $P_m \epsilon R^{mx(d+1)}$ directly from the coefficients of the original polynomials and then the application of Gaussian transformations and shifting on it. The dimensions of b.m. P_m are far lower than the dimensions of the matrices used in generalised resultant based methods. For a given $P_{m,d}$, the following table shows the precise dimensions of the initial matrix A used for the evaluation of the g.c.d. of $P_{m,d}$ in some of the existing methods.

The present method	$A \epsilon R^{mx(d+1)}$
Barnett's method	$A \epsilon R^{dx(m-1)d}$
Sylvester's method	$A \epsilon R^{[(m-1)d+p]x(d+p)}$, p is the maximum degree of the polynomials $P_i(s)$, $deg\{P_i(s)\} \leq d-1$
Blankiship's method	$A \epsilon [\ P_m(s) \mid I_m\ ]$

One of the advantages of the method is that of choosing a linearly independent subset of the original set of polynomials and initiates the process with this subset. Thus, we have a remarkable decrease in the number of polynomials used by this step. For example, if we are given a $P_{15,20}$ set, with $\rho(P_{15})=5$ the method will find a most orthogonal basis for the row space of P_{15} and starts the evaluations using only five polynomials. On the contrary, most of the

other methods will start their evaluations with the whole set of fifteen polynomials.

From the previous numerical examples it is clear, that the method deals successfully with polynomials of high degree and with large coefficients. Also, it works extremely satisfactorily with sets containing a large number of polynomials. The more polynomials we have the fewer iterations required for the convergence of the algorithm.

In case that we have polynomials with nearly equal factors, the method catches an approximate g.c.d. of the set. For example, for $\mathbf{P_1(s)} = (s-1)^2 \cdot (s-2) \cdot (s-2-\varepsilon)$, $\mathbf{P_2(s)=P_1'(s)}$, $\varepsilon \leq 10^{-3}$ [Pac. & Bar., 1], we get an approximate g.c.d. of the set which equals to $s^2-3.0005s+2.0005$ (using 4 decimal digits). A modification of the method that may compute a better estimate of the g.c.d. of such polynomials is under study.

The value of n used in **STEP 3** of the algorithm should not be fixed, but will vary according to the given data. In examples (7.4) and (7.5), where we have large coefficients and a large number of polynomials respectively, the right values of n that catches the correct degree of the g.c.d. are 10^{-5}, 10^{-8}. On the other hand, in example (7.2) it is 10^{-11}. It seems that perhaps a mathematical relation between the original data and n exists. It might be possible to express n as a function of $P_{m,d}$ and of the computer's accuracy. It must be also noticed that results corresponding to small values of n ($n \leq 10^{-8}$) are far more accurate than others corresponding to larger values of n.

From the examples, the following observations about the values of tolerances ε_G and ε are made. The value ε_G is influenced from the original data; this is why its value may vary. (But generally it will be less than 10^{-8}, the upper bound is achieved for sets such as $P_{11,20}$). The value of ε will be common for any set of data (e.g. 10^{-11}). Of course, both of them are closely connected with the computer's accuracy.

Unless we know exactly the degree of the g.c.d. of a $P_{m,d}$ we cannot specify from the beginning the number of iterations required by the algorithm. In case that the degree of the g.c.d. is known, let us say k, the maximum number of iterations needed for a given $P_{2,d}$ set is 2(d-k). For m>2 this number is seriously decreased. Amongst all the sets $P_{m,d}$ the one that requires the maximum number of iterations for the evaluation of its g.c.d. is a set $P_{2,d}$ of coprime polynomials (i.e. k=0). Such a set requires 2d iterations. Since the least degree d_{min} of the polynomials in $P_{m,d}$ defines an upper bound for the number of iterations needed for the convergence of the method.

7.6 CONCLUSIONS

The aim of this Chapter was to provide a new numerical method for the computation of the g.c.d. of polynomials. The main advantages of this new method were:

(a) The simplicity in constructing the basis matrix.

(b) The remarkable decrease in the polynomials used by the method.

(c) The success in handling a large number of polynomials having high degree and large coefficients.

(d) The ability to catch up approximate solutions.

Thus, this Chapter serves the following purpose:

- It provides an attractive and stable numerical algorithm evaluating the g.c.d. of any set of polynomials of R[s].

CHAPTER 8

COMPUTATION OF ALMOST ZEROS AND ZERO TRAPPING DISCS

8.1 INTRODUCTION

In dealing with engineering systems models, on the one hand the uncertainty about the true value of the parameters, and on the other hand, because of the finite number of significant digits used in a computer, a mathematically singular matrix is indistinguishable from a "nearly singular" matrix. Thus, it might seem that the precise concepts of zeros (multivariable zeros, decoupling zeros) that are defined as those frequencies where certain polynomial matrices lose rank, are of little relevance for engineering system models, since either parameter uncertainty or round off computational errors make the test of those exact concepts rather impossible. Therefore, the notion of almost zero was developed in [Kar., Gia. & Hub., 1].

The aim of this Chapter is to study several aspects arising from the almost zero definition. In the beginning, a brief description of the most important properties characterising the almost zero is presented. In the sequel, an efficient algorithm evaluating the prime almost zero of a given polynomial set is developed. The major issue of almost zero's sensitivity is also considered. The notions of B-scaled, normalized, $\| \ \|_\infty$ row-scaled almost zero are introduced and a variety of examples applying different scalings to the original polynomials are tested. Useful results concerning the relation between almost zero's position in the complex plane and the distribution of the roots of the original polynomials are derived.

Next, our study is focused on polynomial combinants of a set of polynomials. An efficient algorithm computing an upper bound for the zero radius is presented. The sensitivity of almost zero to scaling is used for the specification of improved bounds for the zero trapping region. Finally the definition and the most important properties of the fixed order dynamic combinants are introduced.

8.2 ALMOST ZEROS OF A SET OF POLYNOMIALS OF R[s]

8.2.1 Almost Zero Equivalence [Kar., Gia. & Hub., 1]

We recall, from Chapter 7, that if

$P=\{p_i(s): p_i(s)=a_0^i+a_1^i s+\ldots+a_{d_i}^i s^{d_i} \in R[s],\ a_{d_i}^i \neq 0,\ i=1,2,\ldots m\}$ is a set of polynomials and $d=\max\{d_i,\ i=1,2,\ldots,m\}$, then a polynomial vector $\underline{p}(s)=[p_1(s),\ldots,p_m(s)]^t = P_d \cdot \underline{e}_d(s) \in R^m[s]$ may be always associated with the set P, where $P_d \in R^{mx(d+1)}$ is the basis matrix of $\underline{p}(s)$ given by Eqn (7.1) and $\underline{e}_d(s)=[1,s,\ldots,s^d]^t$. When $s \in C$, $\underline{p}(s)$ defines a vector valued analytic function

with domain C and co-domain C^m; the norm of $\underline{p}(s)$ is defined as a positive definite real function with domain C as

$$\|\underline{p}(s)\| = \sqrt{\underline{p}^t(s^*)\underline{p}(s)} = \sqrt{\underline{e}_d{}^t(s^*)P_d{}^T P_d \underline{e}_d(s)} \tag{8.1}$$

where s^* is the complex conjugate of s. Note, that if $q(s)=s+a$ is a common factor of the polynomials $p_i(s)$, $i=1,\ldots,m$, then for all $i=1,\ldots,m$ $p_i(-a)=0$, $\underline{p}(-a)=\underline{0}$ and thus $\|\underline{p}(-a)\|=0$. This observation leads to the following definition.

Definition (8.1): Let P be a set of polynomials of R[s], $\underline{p}[s]$ be the associated polynomial vector and let $\varphi(\sigma,w)=\|\underline{p}(s)\|$, where $s=\sigma+wj\epsilon C$.
An ordered pair (z_k,ε_k), $z_k\epsilon C$, $\varepsilon_k\epsilon R$ and $\varepsilon_k\geq 0$, defines an <u>almost zero</u> (a.z.) of P at $s=z_k$ and of order ε_k, if $\varphi(\sigma,w)$ has a minimum at $s=z_k$ with value ε_k. From the set $Z=\{(z_k,\varepsilon_k),\ k=1,\ldots,r\}$ of almost zeros of P the element (z^*,ε^*) for which $\varepsilon^*=\min\{\varepsilon_k,k=1,\ldots,r\}$ is defined as the <u>prime almost zero</u> of P. ■

It is clear that if P has an exact zero, then the corresponding ε is zero. Clearly, Definition (8.1) is an extension of the concept of exact zero to that of the almost zero. The magnitude of ε at an almost zero $s=z$ provides an indication of how well z may be considered as an approximate zero of P; we should note, however, that ε depends on the scaling of the polynomials $p_i(s)$ in P by a constant c, $c\epsilon R-\{0\}$.

The set P may be standardized in various ways; we shall adopt the following standardization: Let $\underline{p}(s)$ be written as

$$\underline{p}(s) = \underline{p}_0+\underline{p}_1 s+\ldots+\underline{p}_d s^d \tag{8.2}$$

The polynomial vector $\underline{p}(s)$ may also be expressed around $s=a$, $a\epsilon C$, by a Taylor expansion as

$$\underline{p}(w)=\underline{b}_0+\underline{b}_1 w+\ldots+\underline{b}_d w^d=B_d\cdot\underline{e}_d(w),\ w=s-a,\ \underline{b}_i\epsilon C^m \tag{8.3}$$

where

$$\underline{b}_0=\underline{p}(a) \text{ and } \underline{b}_i=\frac{1}{i!}\left[\frac{d^i\{\underline{p}(s)\}}{ds^i}\right]_{s=a},\ i=1,\ldots,d \tag{8.4}$$

Definition (8.2): Let P', P'' be two sets of polynomials in R[s] and let $\underline{p}'(s)=P_d'\cdot\underline{e}_d(s)$, $\underline{p}''(s)=P_d''\cdot\underline{e}_d(s)$ be their respective polynomial vector representations, where $P_d'\epsilon R^{rx(d+1)}$, $P_d''\epsilon R^{kx(d+1)}$ are the corresponding basis

matrices. The sets P', P'' will be said to be <u>normally equivalent</u> (NE) and shall be denoted by $P' E P''$, if there exists an orthogonal matrix Q such that

$$P_d' = Q\left[\frac{P_d''}{O_{r-k}}\right], \quad Q\epsilon R^{rxr}, \quad \text{if } r\geq k \quad \text{or}$$

$$\left[\frac{P_d'}{Q_{k-r}}\right] = QP_d'' , \quad Q\epsilon R^{kxk}, \quad \text{if } k\geq r \tag{8.5}$$

■

Note that such an equivalence is defined between sets of polynomials of R[s] which have the same degree, but not necessarily the same number of polynomials.

<u>Proposition (8.1)</u>: The almost zero structure of a set of polynomials P, which is defined by $\|\underline{p}(s)\|$ is invariant under normal equivalence.

■

Note that the elementary row operations which reorder the rows of a matrix belong to the family of orthogonal transformations. Thus we have that the particular ordering of the polynomials $p_i(s)$ of P in the vector $\underline{p}(s)$ does not affect the almost zero structure of P.

An important consequence of Proposition (8.1) is that the almost zero structure of P may be studied on any set P of the normal equivalence class $E(P)$ of P.

If $s=z$ is an exact zero of P, then $s=z$ is also an exact zero of $P'=\{k_i \cdot p_i(s),\ k_i\epsilon R-\{0\},\ i=1,2,\ldots,m\}$. This property, however, does not extend to the case of almost zeros; in fact, the almost zero pattern of a set P is affected by the scaling of the polynomials by different nonzero constants.

<u>Remark (8.1)</u>: The sets of polynomials $P=\{p_i(s),\ i=1,\ldots,m\}$ and $P'=\{k_i \cdot p_i(s),\ k_i\epsilon R-\{0\},\ i=1,\ldots,m,\ k_i \neq k_j$ for at least two $i,j\}$ do not belong to the same normal equivalence class. If $k_i=k$ for all $i=1,2,\ldots,m$ then $\varphi'(\sigma,w)=|k|\varphi(\sigma,w)$ and P', P have the same almost zero distribution. If (z_i,ε_i), (z_i',ε_i') are almost zeros of P, P' respectively for which $z_i=z_i'$, then $\varepsilon_i'=|k|\varepsilon_i$.

■

8.2.2 **The Location of Prime Almost Zeros: The Prime Disc** [Kar., Gia.& Hub.,1]

The general properties of the distribution of the almost zeros of the set of polynomials P on the complex plane are briefly considered next.

Proposition (8.2): Let P be a set of polynomials of R[s], $P_d \epsilon R^{mx(d+1)}$ be the basis matrix of P and let $\bar{\gamma}$, $\underline{\gamma}$ be the maximum, minimum singular values of p_d. Then

$$\underline{\gamma}\|\underline{e}_d(s)\| \leq \varphi(\sigma,w) \leq \bar{\gamma}\|\underline{e}_d(s)\| \tag{8.6}$$

■

Lemma (8.1): If (z,ε^2) is a minimum of $\|\underline{p}(s)\|^2$, $s\epsilon C$, $z\epsilon C$, $\varepsilon>0$, then (z,ε) is a minimum of $\|\underline{p}(s)\|$ and vice versa.

■

Lemma (8.1) implies that $\varphi(\sigma,w)^2$ may be used for the computation of almost zeros instead of $\varphi(\sigma,w)=\|\underline{p}(s)\|$. The location of prime almost zeros is defined by the following result.

Theorem (8.1): The prime almost zero of P is always within the circle centered at the origin of the complex plane and with radius p^*, defined as the unique positive solution of the equation

$$1+r^2+\ldots+r^{2d} = \bar{\gamma}^2/\underline{\gamma}^2 = \theta^2 \tag{8.7}$$

■

The disc $[0,p^*]$ within which the prime almost zero lies, will be referred to as the prime disc of P. The radius p^* of the prime disc is defined by the degree d and the condition number θ of P. Clearly p^* is an invariant of $E(P)$. The following general results may be stated for the radius p^*.

Corollary (8.1): If d is the degree and θ is the condition number of P. then the radius $p^*=g(d,\theta)$ of the prime disc is a uniquely defined function of d and θ and it has the following properties:

(i) the radius p^* is invariant under the scaling of the polynomial of P by the same nonzero constant c;

(ii) the radius p^* is monotonically decreasing function of d and $1/\theta$;

(iii) the radius p^* is within the following intervals:

a) if $d+1>\theta^2$, then $0<p^*<1$

b) if $d+1<\theta^2$, then $2<p^*<\theta^{1/2}$

c) if $d+1=\theta^2$, then $p^*=1$

■

The conditioning of the polynomials plays an important role in determining the position of the prime almost zero. In fact, the prime almost zero is always in the vicinity of the origin of the complex plane. The uncertainty in its exact position is measured by the radius of the prime disc. Well conditioned sets of polynomials P (i.e. $\theta\approx 1$) have a very small radius prime disc even for very small values of the degree d. Badly conditioned sets of polynomials P (i.e. $\theta>>1$) have very large radius disc, even for large values of the degree d. Thus, for well conditioned sets of polynomials which are also characterized by a large d, the prime almost zero is very close to the origin. Necessary, but not sufficient condition, for the prime almost zero of a set P to be away from the origin of the complex plane, is that P is badly conditioned and its degree is relatively small.

8.2.3 The Computation of Almost Zeros

In this section, the conditions defining the exact location of almost zeros are considered, as well as a numerical technique for their calculation.

Proposition (8.3) [Kar., Gia. & Hub., 1]: Let P be a set of polynomials of $R[s]$, $\underline{p}(s)=P_d\cdot\underline{e}_d(s)$ be the polynomial vector associated with P and $P_d\epsilon R^{mx(d+1)}$ be the corresponding basis matrix. Necessary conditions for $z\epsilon C$ to be an almost zero of P are

$$\underline{e}_d{}^t(z^*)\Delta^T P_d{}^T P_d\cdot\underline{e}_d(z)=0 \quad \text{and} \quad \underline{e}_d{}^t(z^*)P_d{}^T P_d\Delta\underline{e}_d(z)=0 \tag{8.8}$$

where

$$\Delta = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 2 & 0 & \dots & 0 & 0 \\ 0 & 0 & 3 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & d & 0 \end{bmatrix}$$

■

Remark (8.2) [Kar., Gia. & Hub., 1]: If we set $\underline{e}_d(s)=\underline{r}+\underline{q}j$, where $\underline{r}=Re\{\underline{e}_d(s)\}$, $\underline{q}=Im\{\underline{e}_d(s)\}$, $s=\sigma+wj$, then conditions (8.8) are reduced to the following equivalent set

$$\underline{r}^t\Delta^T P_d{}^T P_d\underline{r}+\underline{q}^t\Delta^T P_d{}^T P_d\cdot\underline{q}=0 \quad \text{and} \quad \underline{r}^t\Delta^T P_d{}^T P_d\underline{q}-\underline{q}^t\Delta^T P_d{}^T P_d\cdot\underline{r}=0 \tag{8.9}$$

The components r_i, q_i of $\underline{r}$, $\underline{q}$ are given in terms of σ,w by

$$r_i = \sum_{k=odd}^{i} \binom{i-1}{k-1} \sigma^{i-k} (w_j)^{k-1},$$

$$q_i = \sum_{k=even}^{i} \binom{i-1}{k-1} \sigma^{i-k} (w_j)^{k-1} \qquad (8.10)$$

■

Conditions (8.8) or the equivalent set (8.9) are necessary conditions for determining the location of an almost zero, but not sufficient. A set of sufficient conditions is discussed next.

Proposition (8.4) [Kar., Gia. & Hub., 1]: Sufficient conditions for a solution $z=\sigma+w_j \epsilon C$ of conditions (8.8) or (8.9) to be an almost zero of P are

$$\underline{e}_d^t(z^*)\{(\Delta^T)^2 P_d^T P_d + 2\Delta^T P_d^T P_d \Delta + P_d^T P_d \Delta^2\}\underline{e}_d(z) > 0 \qquad (8.11)$$

and

$$2\{\underline{e}_d^t(z^*)\Delta^T P_d^T P_d \Delta \underline{e}_d(z)\}^2 > \{\underline{e}_d^t(z^*)(\Delta^T)^2 P_d^T P_d \underline{e}_d(z)\}^2 + \{\underline{e}_d^t(z^*) P_d^T P_d \Delta^2 \underline{e}_d(z)\}^2 \qquad (8.12)$$

■

The results of Proposition (8.3) and (8.4) may be used for the analytic computation of the almost zeros of a set of polynomials P. In practice, such analytic computations are tedious and therefore it is recommended to avoid their use in a convenient numerical method. Due to this restriction the following numerical algorithm is suggested for the computation of a prime almost zero.

Algorithm ALMZERO

Let P be a set of polynomials of R[s], $\underline{p}(s)=P_d \cdot \underline{e}_d(s)$ be the polynomial vector associated with P and $P_d \epsilon R^{mxn}$, n=d+1 be the corresponding basis matrix. The following algorithm computes a prime almost zero $z=\sigma+wj\epsilon C$ of the given set P.

STEP 1: Find the singular values σ_i, $i=1,2,..,r=\min\{m,n\}$ of matrix $P_d \epsilon R^{mxn}$.

Determine the maximum index $j\epsilon\{2,3,\ldots,r\}:\sigma_j \neq 0$

$$\theta := \frac{\sigma_j}{\sigma_1}$$

STEP 2: Evaluate the radius p^* of the prime disc

p^* := the unique positive solution of the equation

$1+x^2+\ldots+x^{2(n-1)}=\theta^2$

STEP 3: Determine an initial value for $s=\sigma+wj\epsilon C$

$$\sigma := \frac{p^*}{3}$$

$$w := -\frac{p^*}{3}$$

Evaluate the vector $\underline{e}_d(s)=\underline{r}+\underline{q}j$, $\underline{r},\underline{q}\epsilon R^n$

for $i = 2,\ldots,n$

  for $k = 1,\ldots,i$

    if k=even **then**

$$q_i := \sum_k \binom{i-1}{k-1} \sigma^{i-k} w^{k-1} (-1)^{\frac{k-2}{2}}$$

    else

$$r_i := \sum_k \binom{i-1}{k-1} \sigma^{i-k} w^{k-1} (-1)^{\frac{k-1}{2}}$$

Evaluate the function

$$\text{func}(s) := \|\underline{p}(s)\|^2 = \underline{e}_d{}^t(s^*) \cdot P_d{}^T P_d \cdot \underline{e}_d(s)$$

STEP 4: Find $\min_s\{\text{func}(s)\}$

z := the value of s that minimizes func(s)

ε := the value of the minimum at z

Alg: 8.1

Implementation of the algorithm

For the implementation of this algorithm several routines from NAG Library were used. The evaluation of the singular values was attained using the routine F02WCF which was described in Chapter 5. For the evaluation of the radius p^*, subroutine C02AEF was used. This subroutine finds all the roots of the (n-1)th order polynomial equation, using the method of Grant and Hitchins. Its computational complexity is approximately proportional to $O(n^2)$. The minimization of function func(s) is achieved by using subroutine E04KCF. This subroutine uses a modified Newton algorithm for finding a minimum of a function $f(x_1,\ldots,x_n)$. The number of iterations required depends on the number of variables, the behaviour of $f(\underline{x})$ and the distance of the starting point from the solution. The number of operations performed in an iteration of E04KCF is roughly proportional to $n^3+O(n^2)$. In addition, each iteration makes at least m+1 calls of FUNCT2, where m is the number of variables. FUNCT2 evaluates the value of $f(\underline{x})$ and its first derivatives. For the evaluation of function value $f(\underline{x})$ the computation of $\underline{e}_d(s)$ and $P_d{}^T \cdot P_d$ are required. This task requires $O(n^3)$ and n^2m flops respectively. In the sequel, the evaluation of the function func(s) needs n^2+n flops. For the computation of the first derivatives of function func(s) with respect to σ and w, $O(\frac{4n^3}{3})$ flops are required.

Example (8.1): Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$

$$\underline{p}(s)=\begin{bmatrix}p_1(s)\\p_2(s)\\p_3(s)\end{bmatrix}=\begin{bmatrix}s^3+5.5s^2+11s+7.5\\s^2-1\\s-2\end{bmatrix}=\begin{bmatrix}7.5 & 11 & 5.5 & 1\\-1 & 0 & 1 & 0\\-2 & 1 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}1\\s\\s^2\\s^3\end{bmatrix}=P_3\cdot\underline{e}_3(s),\ P_3\epsilon R^{3x4}$$

We want to evaluate the almost zero z of the above set of polynomials.

Applying algorithm **ALMZERO** we obtain the following results (they are written using an accuracy of five decimal digits).

The singular values of matrix P_3 are:

$\sigma_1 = 14.44295$, $\sigma_2=2.42926$, $\sigma_3=1.00000$

$$\theta = \frac{\sigma_1}{\sigma_3} = 14.44295$$

The radius p^* is the unique positive solution of the equation $1+r^2+r^4+r^6=(14.44295)^2$. We obtain that $p^*=r=2.35644$.
The almost zero is located at the point:

$$z=(-0.99999,\ 0.)$$

and the corresponding function norm has the value $\varepsilon=10.0$.

■

Example (8.2): Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$

$$\underline{p}(s)=\begin{bmatrix} p_1(s) \\ p_2(s) \end{bmatrix} = \begin{bmatrix} 2s^3+3s+2s+3 \\ 10s^3+15s^2+14s+21 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 3 & 2 \\ 21 & 14 & 15 & 10 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \end{bmatrix} = P_2 \cdot \underline{e}_3(s),\ P_2 \epsilon R^{2x4}.$$

It is known that the above two polynomials $p_1(s)$, $p_2(s)$ are not coprime and they have an exact zero at -1.5. Consequently their almost zero $z=\sigma+wj$ must be located somewhere in the area of -1.5. In order to achieve this desired result, the value of the initial point required for starting the minimization process must be taken into account.More analytically, the following results were obtained (They are given using an accuracy of five digits):
The singular values of P_2 and the condition number are:
$\sigma_1=31.42157$, $\sigma_2=0.82746$, $\theta=37.97367$
The radius of the prime disc is $p^*=3.30798$

initial point	**almost zero (σ,w)**	**norm**
$(\frac{p^*}{3}, -\frac{p^*}{3})$	(-0.00044, -1.17635)	2.236
(-1, -0.2)	(-1.5, 0)	$0.26177 \cdot 10^{-14}$
(-1.4, -0.0001)	(-1.5, 0)	$-0.93213 \cdot 10^{-11}$
(-2, 0.5)	(-1.5, 0)	$0.13644 \cdot 10^{-11}$
(1, 1)	(-0.00044, 1.17635)	2.236

We should make the remark that when the initial point is somewhere in the area of the exact zero, then the final value of the almost zero is very close to

the exact one. On the other hand, if we select an initial point that does not belong to the same halfplane with the exact zero, then the final computed value of the almost zero varies significantly from that of the exact zero. Therefore, when we have a set of noncoprime polynomials and the value of the initial point belongs in an area of a common root, the almost zero algorithm specifies this root.

■

For a given noncoprime set of polynomials, the following example illustrates a procedure for the specification of the roots of their greatest common divisors.

Example (8.3): Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$

$$\underline{p}(s)=\begin{bmatrix} s^4-5s^3+5s^2+5s-6 \\ s^4-10s^2+9 \end{bmatrix} = \begin{bmatrix} -6 & 5 & 5 & -5 & 1 \\ 9 & 0 & -10 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \\ s^4 \end{bmatrix} = P_2 \cdot e_4(s),\ P_2 \epsilon R^{2x5}.$$

The singular values of P_2 are $\sigma_1=15.993$, $\sigma_2=6.1819$. Thus, $\rho(P_2)=2<d+1=5$. From Remark (7.5) we conclude that the polynomials are not coprime and a greatest common divisor $\varphi(s)$ exist.

Applying Proposition (7.3) we compute the LEF P_2^H of P_2

$$P_2 \xrightarrow{E} \begin{bmatrix} 1 & 0 & -\frac{20}{18} & 0 & \frac{1}{18} \\ 0 & 1 & -\frac{5}{15} & -1 & 2 \end{bmatrix} = P_2^H$$

The minimal degree polynomial in the $shf(P_{2,4}^H)$ is $t(s)=2s^3-s^2-\frac{5}{15}s+1$.

Therefore $\deg\{\varphi(s)\}\leq 3$. Using algorithm **ALMZERO** the following results were obtained:

The condition number of P_2 is $\theta=2.5871$ and the radius p^* of the prime disc is equal to 1.07065

initial point	almost zero (σ,w)	norm
$(\frac{p^*}{3}, -\frac{p^*}{3})$	(1.00 , 0.00)	$0.28422 \cdot 10^{-13}$
$(-\frac{p^*}{3}, -\frac{p^*}{3})$	(-1.00 , 0.00)	$0.49364 \cdot 10^{-13}$
(0.5, -0.5)	(1.00 , 0.00)	$-0.19895 \cdot 10^{-12}$
(-0.5, -0.5)	(-1.00 , 0.00)	$0.11369 \cdot 10^{-12}$
(2.8, -0.001)	(2.999999094, 0.00)	$-0.10515 \cdot 10^{-9}$
(1 , 1)	(1.00 , 0.00)	$-0.11369 \cdot 10^{-12}$
(-1 , 2.5)	(-1.00 , 0.00)	$0.56851 \cdot 10^{-12}$

We remark that after the application of various initial points, the values of almost zeros corresponding to the lowest values of norm are approximately equal to 1, -1, 3. Taking into account the fact that $\varphi(s)$ can have three roots or less we conclude that the roots of $\varphi(s)$ are 1, -1, 3.

A major disadvantage in specifying the roots of the greatest common divisor of a polynomial set is due to the uncertainty in choosing the initial points. The whole process is considered to be successful only if the selected initial points are quite close to areas of the initial roots.

■

8.2.4 Sensitivity of Almost Zeros

The objective of this section is to investigate the sensitivity of almost zeros and define the parameters which affect it.

It has been shown that the scaling of the polynomials by the same nonzero constant does not affect the pattern of the almost zeros, however this does not hold true when the polynomials are scaled by different nonzero constants. In such cases, the position of the almost zero varies according to the scaling which is used. An obvious question arising is whether we can use this property of scaling the polynomials to shift the almost zero to a particular position in the complex plane.

Before discussing analytically the effect that scaling of the original polynomials causes in the position of the almost zero, we must first discuss the type of scalings which can be applied to the polynomials. In section 7.2 of Chapter 7 was shown that each polynomial set $P_{m,d}$ can be expressed using the basis matrix $P_m \epsilon R^{mx(d+1)}$ and the polynomial vector $\underline{e}_d(s)$. Thus, the

problem of scaling the original polynomials is equivalent to premultiplication of P_m by a diagonal matrix $D \epsilon R^{mxm}$. The diagonal elements of the matrix D can be chosen randomly or can be properly adjusted to provide a "heavy" or "light" scaling of specific polynomials.

In Numerical Analysis it is generally agreed that the best way to scale a matrix $A \epsilon R^{nxn}$ is to make the condition number of the transformed matrix as small as possible. (Recall that $\text{cond}(A)=\|A\| \cdot \|A^{-1}\|$, so the solution depends on the choice of norm). Given that the quality of numerical computations is generally improved if the condition number of the matrix concerned is reduced, the following four minimization problems are investigated in order to get optimal pre-scaling of a matrix.

(i) $\inf_{D_1,D_2} \text{cond}\,(D_1AD_2)$, (ii) $\inf_{D_2} \text{cond}\,(AD_2)$

(iii) $\inf_{D_1} \text{cond}\,(D_1A)$, (iv) $\inf_{D} \text{cond}\,(D^HAD)$ for Hermitian A.

In [Bau., 1] a serious study of the above problems is developed. In [Gol. & Var., 1] it is proved that a matrix A with singular value decomposition of the form $A=U\Sigma V^T$ is best scaled in the Euclidean norm if the first and last columns of U and V have components of equal magnitude. This is referred to as the EMC property. More things concerning optimal scaling of square matrices can be found in [For. & Str., 1].

In our case, the given matrices P_m are generally rectangular and since the above mentioned determination of optimal scaling has not been extended to rectangular matrices (only in [Gol., & Var., 1] this problem is introduced and a conjecture regarding a possible best scaling is made), the following scalings will be used:

(a) Normalization: This kind of scaling was developed and used in previous Chapters. (Chapter 5)

(b) B-scaling: This kind of scaling was also developed in section 7.4.1 of Chapter 7.

(c) $\|\ \|_\infty$ -row scaling: For a given matrix $A \epsilon R^{mxn}$ this scaling is defined in the following way:

We choose the diagonal factors d_i, i=1,2,...,m of matrix D according to the formula:

$$d_i = \frac{1}{\sum_{j=1}^{n} |a_{ij}|}, \quad i=1,2,\ldots,m$$

The absolute row sums of DA are then all equal. When $A \epsilon R^{nxn}$ the above defined matrix D minimizes $\|DA\|_\infty \cdot \|A^{-1}D^{-1}\|_\infty$ [Nob. & Dan., 1]

Remark (8.3): For a given matrix $A \epsilon R^{mxn}$, normalization and $\| \ \|_\infty$ row scaling provide a standardization for the equivalent class $E=\{A' \epsilon R^{mxn}: A'=D \cdot A, D=diag\{d_i\} \epsilon R^{mxm}\}$. In fact, the normalization and the $\| \ \|_\infty$ row scaled form of A are invariants of the class *E*.

■

According to which kind of scaling is applied to the original basis matrix, the following definitions can be formulated:

Definition (8.3): Let *P* be a set of polynomials of R[s] and $\underline{p}(s)=P_d \cdot \underline{e}_d(s)$ be the associated polynomial vector. The almost zeros of $D_B \cdot P_d \cdot \underline{e}_d(s)$, $D_N \cdot P_d \cdot \underline{e}_d(s)$, $D_r \cdot P_d \cdot \underline{e}_d(s)$ where D_B, D_N, D_r are the diagonal matrices that B-scales, normalizes, $\| \ \|_\infty$ row scales matrix P_d are defined as the B-scaled, normalized, $\| \ \|_\infty$ row-scaled almost-zero of *P* respectively.

■

Next, we present some characteristic examples illustrating the different positions of almost zero's appearance in the complex plane, attained after the application of specific scalings to the original basis matrix. Each example demonstrates a different situation in relevance with the distribution of the roots of the original polynomials. The numerical results were achieved by programming Algorithm **ALMZERO** [Mit. & Kar., 1].

Example (8.4): Polynomials with well-separated roots

Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$:

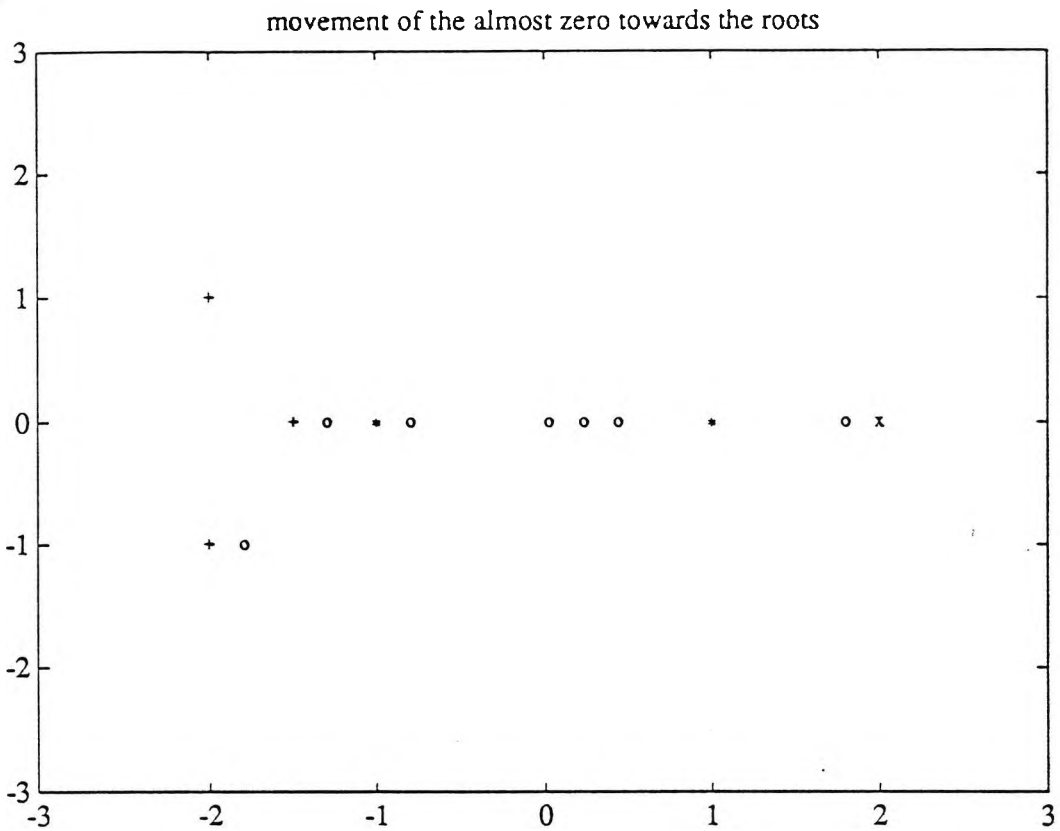
$$\underline{p}(s)=\begin{bmatrix} s^3+5.5s^2+11s+7.5 \\ s^2-1 \\ s-2 \end{bmatrix} = \begin{bmatrix} 7.5 & 11 & 5.5 & 1 \\ -1 & 0 & 1 & 0 \\ -2 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \end{bmatrix} = P_3 \cdot \underline{e}_3(s), \quad P_3 \epsilon R^{3x4}$$

We apply scaling to matrix P_3, each time with a different matrix D and for each occasion we evaluate the radius of the prime disc, the almost zero and the corresponding function norm. In the following matrix we demonstrate the obtained results. (All the values that are less than 10^{-6} are set equal to zero)

scaling	radius p	initial guess	almost zero z=(σ,w)	norm
diag(1,1, 1)	2.35644	ρ/3,-ρ/3	(-0.9999, 0)	10
		-ρ/3,-ρ/3		
normaliza- tion	1.042	ρ/3,-ρ/3	(0.02179, 0)	1.5697
		-ρ/3,-ρ/3		
B-scaling	1.13623	ρ/3,-ρ/3	(0.2289, 0)	0.089
		-ρ/3,-ρ/3		
$\|\|\|_\infty$ row scaling	1.03	ρ/3,-ρ/3	(0.434, -0.4)	0.7241
		-ρ/3,-ρ/3		
diag(20, 20,20)	2.35644	ρ/3,-ρ/3	(-0.9999, 0)	$0.4 \cdot 10^4$
		-ρ/3,-ρ/3		
diag(0.5, 0.5,0.5)	2.35644	ρ/3,-ρ/3	(-0.9999, 0)	2.5
		-ρ/3,-ρ/3		
diag(2,8, 5)	1.47	ρ/3,-ρ/3	(-0.78, 0)	215.864
		-ρ/3,-ρ/3		
diag(1,1, 100)	5.85	ρ/3,-ρ/3	(1.794308, 0)	3000.71
		-ρ/3,-ρ/3		
diag(1,1, 300)	8.47301	ρ/3,-ρ/3	(1.971939, 0)	3471.7897
		-ρ/3,-ρ/3		
diag(1,1, 900)	12.23511	ρ/3,-ρ/3	(1.9966935, 0)	3540.3573
		-ρ/3,-ρ/3		
diag(100, 1,1)	11.28792	ρ/3,-ρ/3	(-1.4995, 0)	13.8091
		-ρ/3,-ρ/3	(-1.999598,-0.9997)	36.989
diag(500, 1,1)	19.3188	ρ/3,-ρ/3	(-1.999, -0.999)	36.9995
		-ρ/3,-ρ/3	(-1.99998,-0.999989)	37.0
diag(800, 1,1)	22.59813	ρ/3,-ρ/3	(-1.99999,-0.99999)	$0.37 \cdot 10^2$
		-ρ/3,-ρ/3		
diag(1, 200,1)	5.44	ρ/3,-ρ/3	(0.999, 0)	623.5729
		-ρ/3,-ρ/3	(-1, 0)	10
diag(1, 0.0001, 0.0001)	52.45957	ρ/3,-ρ/3	(-1.999, -1)	$0.37 \cdot 10^{-6}$
		-ρ/3,-ρ/3	(-1.9999,-0.99999997)	
diag(1, 300,300)	3.66827	ρ/3,-ρ/3	(1.1641, 0)	$0.75 \cdot 10^5$
		-ρ/3,-ρ/3		

Matrix 8.1

Next, a figure showing the movement of the a.z. towards the roots of the given polynomials is presented.



+ : denotes the roots of $p_1(s)$
* : denotes the roots of $p_2(s)$
x : denotes the roots of $p_3(s)$
° : denotes the almost zero

Example (8.5): Polynomials with overlapping roots

Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$:

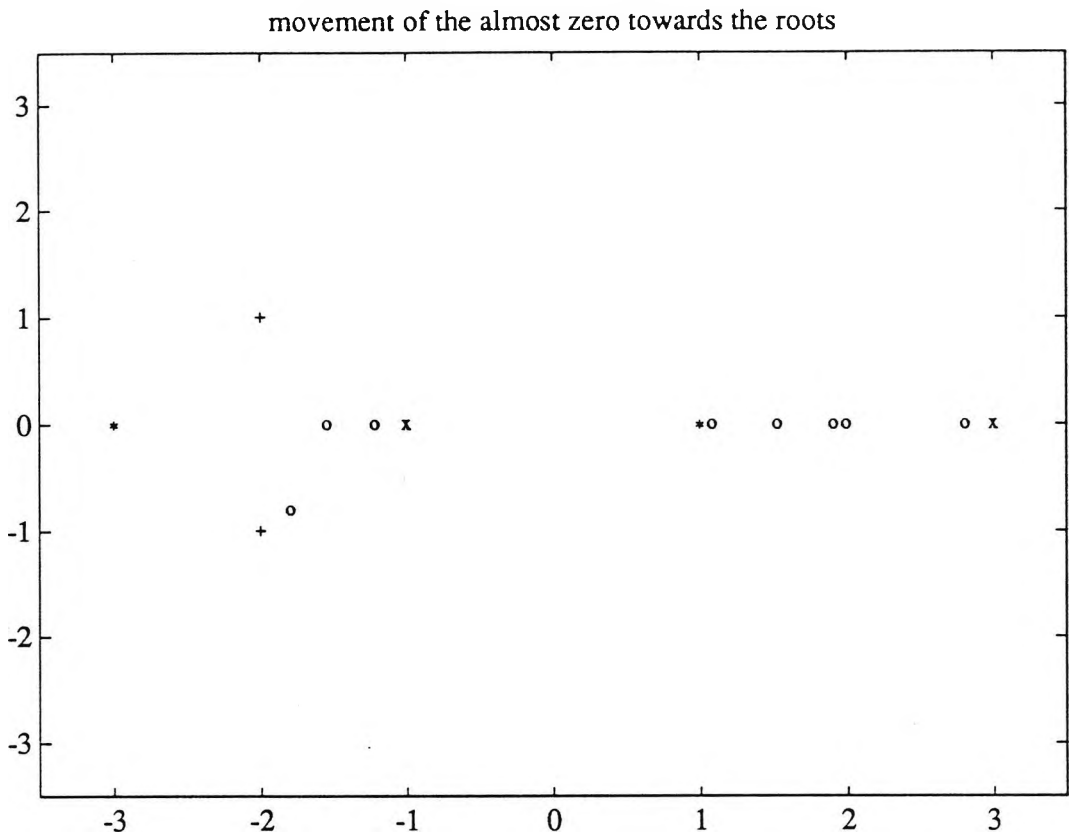
$$\underline{p}(s)=\begin{bmatrix} s^2+2s-3 \\ s^2-2s-3 \\ s^3+2s^2-3s-10 \end{bmatrix}=\begin{bmatrix} -3 & 2 & 1 & 0 \\ -3 & -2 & 1 & 0 \\ -10 & -3 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \end{bmatrix}=P_3\cdot\underline{e}_3(s),\ P_3\epsilon R^{3x4}$$

Following the same process as in the previous example, the subsequent matrix is obtained.

scaling	radius p	initial quess	almost zero z=(σ,w)	norm
diag{1,1,1}	2.61851	ρ/3,-ρ/3	(1.91991, 0)	32.14
		-ρ/3,-ρ/3	(-1.546306,0)	38.16
normaliza-tion	2.30633	ρ/3,-ρ/3	(1.517454, 0)	1.75
		-ρ/3,-ρ/3	(-1.207723,0)	1.412
B-scaling	2.82778	ρ/3,-ρ/3	(1.080865, 0)	0.274
		-ρ/3,-ρ/3	(-1.0418956,0)	0.259
$\|\|\|_\infty$ row scaling	2.2998	ρ/3,-ρ/3	(1.559064, 0)	0.7
		-ρ/3,-ρ/3	(-1.225756,0)	0.562
diag{30,30, 30}	2.61851	ρ/3,-ρ/3	(1.91991, 0)	$0.3\cdot10^5$
		-ρ/3,-ρ/3	(-1.546306,0)	$0.3\cdot10^5$
diag{1000, 1000,1000}	2.61851	ρ/3,-ρ/3	(1.91991, 0)	$0.3.10^8$
		-ρ/3,-ρ/3	(-1.546306, 0)	$0.4.10^8$
diag{100,1, 1}	8.42128	ρ/3,-ρ/3	(1.00025, 0)	115.99
		-ρ/3,-ρ/3	(-2.99865, 0)	243.7
diag{800,1, 1}	17.54788	ρ/3,-ρ/3	(1.000003, 0)	116
		-ρ/3,-ρ/3	(-2.9999833, 0)	244
diag{1,100, 1}	6.76958	ρ/3,-ρ/3	(2.993597, 0)	81.34
		-ρ/3,-ρ/3	(-1.000149, 0)	51.98
diag{1, 1000,1}	14.62443	ρ/3,-ρ/3	(2.999935, 0)	819.93
		-ρ/3,-ρ/3	(-1.0000015, 0)	52
diag{1,1, 100}	11.86578	ρ/3,-ρ/3	(1.9999917, 0)	34
		-ρ/ε,-ρ/3	(-1.999953,-0.999941)	71.996
diag{1,1, 2000}	32.24164	ρ/3,-ρ/3	(1.99999998, 0)	34
		-ρ/3,-ρ/3	(-1.99999988,-0.9999998)	72
diag{0.01, 1,0.01}	6.76958	ρ/3,-ρ/3	(2.993597, 0)	0.08
		-ρ/3,-ρ/3	(-1.00014996, 0)	0.0052

Matrix 8.2

Next, a figure illustrating the movement of the a.z. towards the roots of the given polynomials and according to the results achieved in the above matrix is presented.



+ : denotes the roots of $p_1(s)$
* : denotes the roots of $p_2(s)$
x : denotes the roots of $p_3(s)$
o : denotes the almost zero ■

Example (8.6): Polynomials with neighbouring roots

Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$:

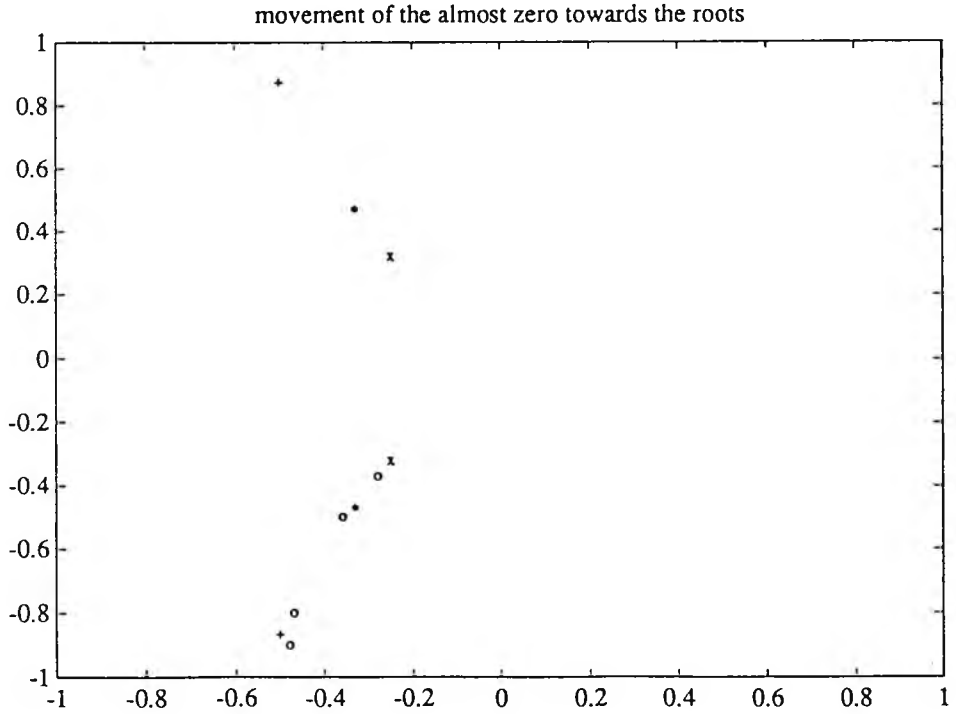
$$\underline{p}(s)=\begin{bmatrix} s^2 & +s & +1 \\ 3s^2 & +2s & +1 \\ 6s^2 & +3s & +1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \end{bmatrix} = P_2 \cdot \underline{e}_2(s),\ P_2 \epsilon R^{3x3}$$

Applying the process demonstrated in the above examples, the next matrix is achieved (The column of initial guess is omitted because although different initial guesses were used, the same results were always obtained)

scaling	radius p	almost zero z=(σ,w)	norm
diag (1,1,1)	7.84078	(-0.2888, -0.37)	0.6
normalization	6.99178	(-0.3675, -0.509	0.1206
B-scaling	7.84078	(-0.2888, -0.372	0.094
$\|\ \|_\infty$ row scaling	6.94	(-0.3563, -0.485	0.0432
diag(15,1,1)	13.32897	(-0.4757, -0.7989)	18.1393
diag(100,1,1)	33.88821	(-0.49935,-0.86427	21.89
diag(800,1,1)	95.83202	(-0.499999,-0.8659	21.998
diag(1,800,1)	121.62907	(-0.333333,-0.4714)	1
diag(1,1,800)	201.66459	(-0.25,-0.3227	0.6944
diag(0.00001,1,1)	1835.05	(-0.273,-0.3558)	0.13218
diag(1,0.001,1)	206.886	(-0.27555,-0.3458)	0.5066
diag(1,1,0.001)	99.95688	(-0.36586,-0.5218)	0.2973
diag(0.01,0.01,0.01)	7.84078	(-0.2888,-0.3725)	$0.6 \cdot 10^{-4}$
diag(0.01,0.02,0.03)	10.75882	(-0.26485,-0.34192)	$0.11 \cdot 10^{-3}$
diag(9,1,2)	11.26995	(-0.3929,-0.5617)	26.355
diag(2,1,2)	9.73554	(-0.279,-0.353)	2.1356

Matrix 8.3

Next, a figure showing the movement of the a.z. is presented.



+ : denotes the roots of $p_1(s)$
* : denotes the roots of $p_2(s)$
x : denotes the roots of $p_3(s)$
° : denotes the almost zero

■

General Remarks

It can be seen from the previous examples that the effect of scaling on a given set of polynomials is to move their almost zero (a.z.) within an area of the complex plane which is surrounded by the particular zeros of the original polynomials.

From the results obtained by the application of various scalings to many different sets of polynomials, the following observations were derived:

1) The a.z. always appears to the halfplane where most of the polynomials' roots are gathered. In some cases (Example (8.5)), it was noticed that for a different initial value of the a.z., a completely different solution was obtained meaning that the a.z. is not uniquely determined. From that fact, a general question arises: Can we specify some convenient criteria that will help us to determine from the beginning and without knowing the location of the original polynomials's roots, an area into which the a.z. always lies?

2) When we apply normalization, we remark that in cases where the polynomials' roots are scattered in both halfplanes then the a.z. is moved closer to the origin of the axis (Example (8.5)). In the specific case when the original zeros of the polynomials are well separated and far from each other, the a.z. is located very close to the origin (Example (8.4)). Also, when all the roots of the given polynomials belong to the same halfplane, then the a.z. is approximately moved towards the centre of the area enclosing the roots. Besides, when the basis matrix P_d is square and a $\| \|_\infty$ row scaling is applied, the obtained position of the a.z. is almost the centre of the area enclosed from the original polynomials' roots. This position is the best amongst all the positions of the a.z. achieved after the application of different scalings. This is due to the fact that, $\| \|_\infty$ row scaling is best in the sense of minimizing $\|DA\|_\infty \cdot \|A^{-1}D^{-1}\|_\infty$, where D is the scaling matrix.

Similar remarks are stated about the application of B-scaling and $\| \|_\infty$ row scaling. Particularly, when the roots of all the given polynomials were gathered in the same halfplane, the application of B-scaling did not actually alter the initial position of the a.z. (Example (8.6)).

3) By scaling heavily a concrete polynomial p_j (with this term we mean that we scale only the particular polynomial using as scaling factor a rather high number) the a.z. is pushed towards a particular zero of that polynomial.
As the scaling factor increases, i.e., the number by which the polynomial is multiplied increases, the polynomial becomes stronger in the sense that its zeros attract the almost zero closer to their location. In this respect we

can see from the examples that a set of three polynomials can be reduced to a set of two polynomials with the same almost zero. For instance, in Example (8.4), the scaling (1,300,300) gives an almost zero located at z=1.1641 and for a set of two polynomials consisting of only the second and third polynomials, the a.z. is located at z=1.1654. It is important to emphasize the fact that this property holds only when the scale factor is rather high. If for example, the scaling (1,20,20) is used then the a.z. of Example (8.4) is located at z=0.91985 which is virtually different from the one achieved using only the second and third polynomials.

An obvious question arising when heavy scaling is applied on a given polynomial concerns the determination of the particular zero of the polynomial which will attract the a.z. After testing a variety of examples, it has been observed that initially the a.z. is attracted from the most close to the origin polynomial root. The following four cases are distinguished:

a) If this root is real and somewhere further from the location of a complex root , then as the scale factor increases the a.z. moves towards the complex root (Example (8.4)).

b) If this root is complex and somewhere further from the location of a real root , then the a.z. always remains inside the area surrounding the complex root no matter how large scaling has been applied. For example, let

$$P_d=\begin{bmatrix} 4 & 6 & 4 & 1 \\ 8 & 2 & -5 & 1 \\ 1.5 & 1 & -3.5 & 1 \\ 1.5 & -2.75 & -1 & 1 \end{bmatrix}$$

is a given basis matrix and $-1\pm i$, -2 are the roots of the first polynomial $p_1(s)=s^3+4s^2+6s+4$. After the application of the scalings (500,1,1,1), (1000,1,1,1), (2000,1,1,1) the following values of a.z. were respectively found: (-0.99987, -0.99977), (-0.99997, -0.99994), (-0.99999, -0.99998).

c) If this root is real and somewhere further away from another real root, then the a.z. always remains inside the area surrounding the closer to the origin real root no matter how large scaling has been applied. For example,

$P_d = \begin{bmatrix} 2 & -2 & 1 \\ -2 & 1 & 0 \\ 1.5 & -3.5 & 1 \end{bmatrix}$ is a given basis matrix and 0.5, 3 are the roots of the third polynomial $p_3(s)=s^2-3.5s+1.5$. After the application of the scalings (1,1,100), (1,1,800), (1,1,2000) the following values of a.z. were respectively found: (0.50004, 0), (0.50000, 0), (0.50000, 0)

d) If this root is complex and somewhere further another complex root exists too, then the a.z. reaches first the closer to the origin root and then as the scale factor goes higher it approaches the furthest complex root. For example, let

$P_d = \begin{bmatrix} 10 & 18 & 15 & 6 & 1 \\ 2 & -3 & 1 & 0 & 0 \end{bmatrix}$ is a given basis matrix and $-1\pm i$, $-2\pm i$ are the roots of the first polynomial $p_1(s)=s^4+6s^3+15s^2+18s+10$. After the application of the scalings (200,1,1), (900,1,1), (2000,1,1) the following values of a.z. were respectively found: (-1.9999, -0.99997), (-2.00000, -1.00000), (-2.00000, -1.00000). The initial position of a.z. was located at (-0.60977, -0.41446).

From all the above cases it is always evident that the application of any heavy scaling did not force the a.z. outside the region surrounded by the particular zeros of the polynomials. Thus the sensitivity depends on the area of this region and the farther these zeros are from each other, the more sensitive the a.z. is.

If instead of applying such heavy scalings we apply light scaling, the achieved results will be exactly the opposite to those obtained after the application of heavy scaling. More specifically, in this case the a.z. is attracted from the roots of the least-affected polynomial. Analogous remarks about which particular zero of this polynomial will attract the a.z. can be derived. A variety of such cases was demonstrated in Examples (8.4), (8.5) and (8.6). Generally, this type of scaling must be used with caution because if the scaling factors are very low the corresponding entries of the basis matrix and therefore its singular values will become very small. Consequently the radius of the prime disc will be quite large. Thus, the scaling of the original polynomial set with low scaling factors ($\leq 10^{-3}$) is not recommended.

4)According to Remark (8.1) if (z_1,ε_1^2) is the a.z. of the original polynomial

set and $(z_1', \varepsilon_1'^2)$ is the one achieved after the application of a uniform scaling (having equal scaling factors $d_1=d_2=\ldots=d_m=d$) then the following relations hold: $z_1'=z_1$, $\varepsilon_1'^2=|d|^2\varepsilon_1^2$.

5) When we scale the polynomials randomly we can not say anything specific about the position of the a.z. but only that it depends on the concrete values of the scaling factors.

6) Generally, after the application of various forms of scalings the value of the norm and the radius of the prime disc increases. The question arising is whether there exists some optimal scaling in the sense of decreasing the above values. After testing a variety of examples, it was concluded that in most cases B-scaling provided the lowest value of norm $\| \|_\infty$ row scaling and normalization has resulted in low values too, but not as low as the ones achieved using B-scaling. The minimum value of the radius of the prime disc was achieved using $\| \|_\infty$ row scaling. Normalization gave low values too. On the contrary, the application of B-scaling was not effective in finding a low value for the above radius.

From the above discussion it is evident that the almost zero's position is related to the distribution of the roots of the original polynomials. Thus, an issue arising is whether it could be possible to formulate a geometric almost zero definition based on the roots of the given polynomials. For example, for a given set $P_{m,d}$ if R_i are the sets containing the roots of the i-th polynomial, i=1,2,...,m the a.z. might be defined as a point in C having the lowest distance from the sets R_i. For such a formulation the notions of distance between sets and between set and a point must be defined too. The whole problem is under consideration.

8.3 POLYNOMIAL COMBINANTS OF A SET OF POLYNOMIALS OF R[s]

8.3.1 Properties of Polynomial Combinants [Kar., Gia. & Hub., 1]

Let $P=\{p_i(s):p_i(s)=a_0^i+a_1^i s+\ldots+a_{d_i}^i s^{d_i}\in R[s],\ a_{d_i}^i\neq 0,\ i=1,\ldots,m\}$

be a set of polynomials and let $d=\max\{d_i,\ i=1,\ldots,m\}$. Let $\underline{p}(s)\in R^m[s]$ be the associated to P polynomial vector. $\underline{p}(s)$ can be expressed as $\underline{p}(s)=P_d\cdot\underline{e}_d(s)$. If we consider a $\underline{k}\in R^m$, then the $\underline{k}$-polynomial combinant of P may be written as

$$f(\underline{k},\underline{p}(s)) = \sum_{i=1}^{m} k_i p_i(s) = \underline{k}^t P_d \cdot \underline{e}_d(s) \tag{8.13}$$

For a set of polynomials P represented by a basis matrix P_d, the zero assignment problem for polynomial combinants may be defined as follows:
Find a $\underline{k}\epsilon R^m$, such that

$$f(\underline{k},\underline{p}(s)) = \underline{k}^t P_d \cdot \underline{e}_d(s) = a(s) \tag{8.14}$$

where $a(s)\epsilon R[s]$ is an arbitrary polynomial. It is clear that the maximum degree of $a(s)$ has to be equal to the degree d of $\underline{p}(s)$. Let us write now $a(s) = a_0+a_1s+\ldots+a_ds^d = [a_0,a_1,\ldots,a_d]\cdot\underline{e}_d(s) = \underline{a}^t\underline{e}_d(s)$, $\underline{a}\epsilon R^{d+1}$, then the zero assignment problem is reduced to the following linear problem:

$$\underline{k}^t \cdot P_d = \underline{a}^t \text{ or equivalently } P_d{}^t \cdot \underline{k} = \underline{a},\ \underline{a}\epsilon R^{d+1} \tag{8.15}$$

A set P for which the zeros of its combinants may be assigned arbitrarily will be called C-assignable or in short assignable; otherwise, P will be referred to as non-C-assignable. If for some $\underline{k}$, the zeros of $f(\underline{k},\underline{p}(s))$ may be assigned at $s=\infty$, then P will be called ∞-assignable; if there is no $\underline{k}$ for which $f(\underline{k},\underline{p}(s))=c$, $c\epsilon R$, i.e. the zeros of some combinants of P cannot be assigned at $s=\infty$, the P will be referred to as nonassignable. Note that assignability implies ∞-assignability, but the opposite is not always true; furthermore, nonassignability implies non C-assignability, but the opposite is not always true.

The following Proposition gives the necessary and sufficient conditions under which a set of polynomials P is assignable or nonassignable.

Proposition (8.5): Let P be a set of polynomials and let P_d be the basis matrix of the polynomial vector associated with P, where

$$P_d=[\underline{p}_0,\underline{p}_1,\ldots,\underline{p}_d]=[\underline{p}_0,\bar{P}_d],\ P_d\epsilon R^{m\times(d+1)} \tag{8.16}$$

(i) The set P is assignable if and only if $\rho(P_d)=d+1$;

(ii) The set P is ∞-assignable if and only if $N_1(\bar{P}_d)\neq\{\underline{0}\}$,

(iii) The set P is non-assignable if and only if $N_1(\bar{P}_d)=\{\underline{0}\}$ or in other words $\rho(\bar{P}_d)=m$.

■

8.3.2 The "Pinning" of Zeros of the Combinants of *P* by the Almost Zeros [Kar., Gia. & Hub.,1]

For a given set of polynomials *P*, let $f(\underline{k},\underline{p}(s))$ be the corresponding polynomial combinant of *P*. The zeros of all combinants of $f(c\cdot\underline{k},\underline{p}(s))$, $c\epsilon R-\{0\}$, are the same and thus, whenever we are interested in the properties of zeros of combinants we may assume that $\|\underline{k}\|=1$. If s=z is an exact zero of *P* (root of a common divisor of all $p_i(s)$), then s=z is also an exact zero of $f(\underline{k},\underline{p}(s))$ for all vectors $\underline{k}$; therefore, the exact zeros of *P* are fixed zeros of $f(\underline{k},\underline{p}(s))$ for all parameter vectors $\underline{k}$. The following Lemma gives a useful property of the exact zeros of *P*.

Lemma (8.2): Let $P=\{p_i(s)\epsilon R[s],\ i=1,2,\ldots,m\}$ be a set of polynomials and let

$$f(\underline{k},\underline{p}(s)) = \sum_{i=1}^{m} k_i p_i(s) = \underline{k}^t P_d\cdot\underline{e}_d(s)$$

be the corresponding $\underline{k}$-polynomial combinant of *P*. Necessary and sufficient condition for $f(\underline{k},\underline{p}(s))$ to have a fixed zero z for all parameter vectors $\underline{k}$ is that z is an exact zero of *P*, or in other words $p_i(s)$ have to be not coprime.

■

The property of the exact zeros of *P* to be fixed zeros of $f(\underline{k},\underline{p}(s))$ for all parameter vectors $\underline{k}$, motivates the investigation of the links between almost zeros of *P* and the exact zeros of various combinants of *P*. The following Theorem shows that for a given $\underline{k}$ the combinants $f(\underline{k},\underline{p}(s))$ have always at least one zero in a disk $D[s_0,R]$, where $s_0\epsilon C$ and R finite; almost zeros will emerge as those points of the C-plane where R becomes minimal.

Theorem (8.2): Let $P=\{p_i(s)\epsilon R[s].\ i=1,2,\ldots,m\}$ be a non-assignable set, $P_d\epsilon R^{mx(d+1)}$ be a basis matrix of *P*, $s_0\epsilon C$ and let

$$\underline{p}(w)=\underline{b}_0+\underline{b}_1 w+\ldots+\underline{b}_d w^d,\ w=s-s_0,\ \underline{b}_i\epsilon c^m \qquad (8.17)$$

be the Taylor expansion of the polynomial vector representative of *P* around $s=s_0$. For every $\underline{k}\epsilon R^m$, the combinant $f(\underline{k},\underline{p}(s))$ of *P* has at least one zero in the finite, minimal radius disk

$$D[s_0,\underline{k}]=\{s:|s-s_0|\le R(s_0,\underline{k}),\ R(s_0,\underline{k})=\min\{[\binom{d}{i}\frac{|\underline{k}^t\underline{b}_0|}{|\underline{k}^t\underline{b}_i|}]^{1/i},\ i=1,\ldots,d\} \qquad (8.18)$$

■

Theorem (8.2) establishes the important fact that for every $\underline{k}\epsilon R^m$ and for every $s_0\epsilon C$ there exists a minimal radius disk $D[s_0,R]$ which includes at least one zero of the $\underline{k}$-combinant of P; we must emphasize the fact that Theorem (8.2) makes no distinction between a general point $s_0\epsilon C$ and the almost zeros of P. The importance of almost zeros in the investigation of the zeros of the combinants of P is demonstrated by the following results.

Proposition (8.6): Let $\underline{p}(s)=\underline{p}_0+\underline{p}_1 s+\ldots+\underline{p}_d s^d$ be a polynomial vector representative of P and let $\underline{p}(w)=\underline{b}_0+\underline{b}_1 w+\ldots+\underline{b}_d w^d$, $w=s-s_0$, be the Taylor expansion of $\underline{p}(s)$ at $s-s_0$, $s_0\epsilon C$. For the various choices of $\underline{k}\epsilon R^m$, upper bounds for $R(s_0,\underline{k})$, are defined by the numbers $\bar{R}_i(s_0,\underline{k})$ as follows:

(i) if $\underline{k}^t\underline{p}_d\neq 0$ then

$$\bar{R}_d(s_0,\underline{k})=\left\{\frac{\|\underline{b}_0\|}{|\underline{k}^t\underline{p}_d|}\right\}^{1/d} \tag{8.19}$$

(ii) if $\underline{k}^t\underline{p}_d=0$ and $\underline{k}^t\underline{p}_{d-1}\neq 0$, then

$$\bar{R}_{d-1}(s_0,\underline{k})=\left\{\binom{d}{d-1}\frac{\|\underline{b}_0\|}{|\underline{k}^t\underline{p}_{d-1}|}\right\}^{1/d-1} \tag{8.20}$$

(iii) if $\underline{k}^t\underline{p}_d=\underline{k}^t\underline{p}_{d-1}=\ldots=\underline{k}^t\underline{p}_{i+1}=0$ and $\underline{k}^t\underline{p}_i\neq 0$, then

$$\bar{R}_i(s_0,\underline{k})=\left\{\binom{d}{i}\frac{\|\underline{b}_0\|}{|\underline{k}^t\underline{p}_i|}\right\}^{1/i} \tag{8.21}$$

■

Theorem (8.3): Let P be a nonassignable set of polynomials, z be an almost zero and z′ be the prime almost zero of P. Then

(i) for all $\underline{k}\epsilon R^m$ such that $\underline{k}^t\underline{p}_d\neq 0$

(a) $\bar{R}_d(z,\underline{k})<\bar{R}_d(s_0',\underline{k})$ for all $s_0'\epsilon C: |s_0'-z|<\varepsilon$

(b) $\bar{R}_d(z',\underline{k})<\bar{R}_d(s_0,\underline{k})$ for all $s_0\epsilon C$

(ii) for all $\underline{k}\epsilon R^m$ such that $\underline{k}^t\underline{p}_d=\ldots=\underline{k}^t\underline{p}_{i+1}=0$, $\underline{k}^t\underline{p}_i\neq 0$

(a) $\bar{R}_i(z,k)<\bar{R}_i(s_0',\underline{k})$ for all $s_0'\epsilon C: |s_0'-z|<\varepsilon$

(b) $\bar{R}_i(z',\underline{k})<\bar{R}_i(s_0,\underline{k})$ for all $s_0\epsilon C$.

■

The above result shows that for a given $\underline{k}$ the radius $\bar{R}_i(s_0,\underline{k})$ is locally minimized when s_0 is an almost zero, and it is globally minimized when s_0 is the prime almost zero. Thus, the almost zeros of P act as poles of attraction for the exact zeros of the k-combinants of P. The radius $R(s_0,\underline{k})$ of the disk $D(s_0,\underline{k})$ within which an exact zero of $f(\underline{k},\underline{p}(s))$ may always be found will be referred to as the zero radius at $s=s_0$.

8.3.3 Computation of an upper bound for the zero radius

Since the zero radius is clearly a function of $\underline{k}$, the following problem is considered;

Find an upper bound for $R(z,\underline{k})$, where z is an almost zero, independent of the parameter vector $\underline{k}$.

The finding of such an upper bound is the subject of the next Theorem.

Theorem (8.4) [Kar., Gia. & Hub., 1]]: Let $p(\underline{w})=\underline{b}_0+\underline{b}_1w+\ldots+\underline{b}_dw^d$, $w=s-s_0$ be the Taylor expansion of the polynomial vector representative $\underline{p}(s)$ of P around $s=s_0$, $s_0\epsilon C$ and let $x\epsilon R^+$. A sufficient condition for

$$R(s_0,\underline{k}) \leq \frac{x}{2^{1/d}-1} \tag{8.22}$$

for all $\underline{k}\epsilon R^m$, is that the matrix $B(s_0,x)$ is positive semidefinite, where

$$B(s_0,x)=\underline{b}_d\underline{b}_d^{*t}x^{2d}+\ldots+\underline{b}_1\underline{b}_1^{*t}x^2-\underline{b}_0\underline{b}_0^{*t} \tag{8.23}$$

The above result can be used in two different ways:

(i) If $x\epsilon R^+$ is fixed, then the positive semidefiniteness of $B(s_0,x)$ implies the existence of an upper bound $R(s_0,x)$ of $R(s_0,\underline{k})$ which is independent of $\underline{k}$; $R(s_0,x)$ is then given by

$$R(s_0,x) = \frac{x}{2^{1/d}-1} \tag{8.24}$$

(ii) Find the minimum positive number x for which $B(s_0,x)$ is positive semidefinite. In this case the smallest of the (8.22) type upper bounds for $R(s_0,\underline{k})$, which are independent of $\underline{k}$, is defined.

In the sequel, we present an algorithm for the computation of such type upper bounds for $R(s_0,\underline{k})$.

Algorithm TRAPDISK

For a given polynomial vector representative $\underline{p}(s)=P_d\cdot\underline{e}_d(s)$, $P_d\epsilon R^{mx(d+1)}$, s_0 an almost zero of this set, the following algorithm evaluates the minimum positive number $x_{min}\epsilon R^+$ for which $\frac{x_{min}}{2^{1/d}-1}$ provides an upper bound for the zero radius $R(s_0,x_{min})$. In the algorithm, x denotes an initial value of x_{min}, maxstep a maximum number of iterations and step a number determining the increase in x.

Construct the coefficients $\underline{b}_i\epsilon R^m$ of the Taylor expansion of $\underline{p}(s)$ around s_0.

$\underline{b}_0 := \underline{p}(s_0)$

$$\underline{b}_i := \frac{1}{i!}\left[\frac{d^i\{\underline{p}(s_0)\}}{ds^i}\right]_{s=s_0}, \quad i=1,2,\ldots,d$$

$x_{min} := x$

for i = 1,...,maxstep

Construct matrix $B(s_0,x)\epsilon R^{mxm}$

$$B(s_0,x) := \sum_{i=d}^{1} b_i\underline{b}_i^{*t}x^{2i}-\underline{b}_0\underline{b}_0^{*t}$$

evaluate the eigenvalues r_i, i=1,2,...,m of the Hermitian matrix $B(s_0,x)$

if $\{r_i>0$ for all i=1,2,..,m$\}$ **then**

if $x<x_{min}$ **then**

$x_{min} := x$

x: = x/2

else

x := x+step

$$R(s_0,x_{min}) := \frac{x_{min}}{2^{1/d}-1}$$

Alg: 8.2

Implementation of the algorithm

For the implementation of algorithm **TRAPDISC**, several numerical routines are required. For the evaluation of the coefficients $\underline{b}_i$, the algorithm of Horner for finding the derivatives of given polynomials is used [Atk., 1]. The computational complexity is analogous to $m(d+1)^2$ flops. For the evaluation of $B(s_0,x)$, $m^2(d+1)$ flops are required. The computation of the eigenvalues of a Hermitian matrix is achieved using subroutine F02AWF of NAG Library. The computational complexity of F02AWF is proportional to m^3. The initial value of x_{min} should be a quite large number which will guarantee the initial positive definiteness of matrix $B(s_0,x)$. In the sequel, assigning to maxstep a rather large integer, a better approximation of x_{min} will be calculated. Since we are dealing with complex numbers the storage requirements of the algorithm are rather important. Roughly, we can say that the algorithm uses seven two dimensional arrays for storing the original matrix, some intermediate results and finally the computed matrix $B(s_0,x)$.

Example (8.7): Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$:

$$\underline{p}(s)=\begin{bmatrix} s^3+5.5s^2+11s+7.5 \\ s^2-1 \end{bmatrix}=\begin{bmatrix} 7.5 & 11 & 5.5 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \end{bmatrix}=P_2\cdot\underline{e}_3(s),\ P_2\epsilon R^{2x4}$$

Applying algorithm **ALMZERO** we located an almost zero at s=-1.199. The Taylor expansion of $\underline{p}(s)$ at w=s+1.199 is $p(\underline{w})=\underline{b}_0+\underline{b}_1w+\underline{b}_2w^2+\underline{b}_3w^3=$

$$\begin{bmatrix} 0.494 \\ 0.438 \end{bmatrix}+\begin{bmatrix} 2.124 \\ -2.398 \end{bmatrix}w+\begin{bmatrix} 1.903 \\ 1 \end{bmatrix}w^2+\begin{bmatrix} 1 \\ 0 \end{bmatrix}w^3$$

In the sequel $B(-1.199,x)$ is computed.

$$B(-1.199,x)=\underline{b}_3\underline{b}_3^{*t}x^6+\underline{b}_2\underline{b}_2^{*t}x^4+\underline{b}_1\underline{b}_1^{*t}x^2-\underline{b}_0\underline{b}_0^{*t}=$$

$$\begin{bmatrix} x^6+3.61x^4+4.51x^2-0.244 & 1.9x^4-5.09x^2-0.216 \\ 1.9x^4-5.09x^2-0.216 & x^4+5.75x^2-0.19 \end{bmatrix}\epsilon R^{2x2},\ x\epsilon R^+$$

Assume that we wish to test whether $R(-1.199,\ \underline{k})\leq 1$ for all $\underline{k}$.

Then, the equation $\frac{x}{2^{1/3}-1} = 1$ gives x=0.2599. For that x we must check whether B(-1.199, 0.2599) is positive definite or not. The answer is negative, therefore the upper bound for R(-1.199, $\underline{k}$) will generally be larger than 1. The equation $\frac{x}{2^{1/3}-1} = 3$ gives x=0.77976 and for that x, B(-1.199, 0.77976) is positive semidefinite. Since R(-1.199, $\underline{k}$)$\leq$3 we may search for smaller bounds for R(-1.199, $\underline{k}$). Using algorithm **TRAPDISK** we locate the minimum positive x=0.55984 for which B(-1.199, 0.55984) is positive semidefinite. This value of x gives a minimal independent of $\underline{k}$ upper bound for R(-1.199, $\underline{k}$) which is found to be equal to 2.15388.

■

8.3.4 Use of sensitivity to scaling for improved bounds of the zero trapping region

It is already known that each disk $D(s_0,\underline{k})$ having as its centre the prime almost zero s_0 of a polynomial set *P*, contains at least one zero of the combinant $f(s,\underline{k})$. An upper bound for the zero radius of this disk was given above. Due to the fact that most of the times the area of the disk is quite large and evidently the uncertainty in the location of the zeros is important, we try to invent a technique which will help us to determine more precisely the position of the zeros inside $D(s_0,\underline{k})$. Since almost zero is sensitive to scaling we apply different scalings to the original set and for each one we locate the corresponding almost zero i.e. the position of the centre of the disk and its radius. Therefore, we get a number of overlapping circles and the region common to all the circles defines the most likely place where a zero can be found.

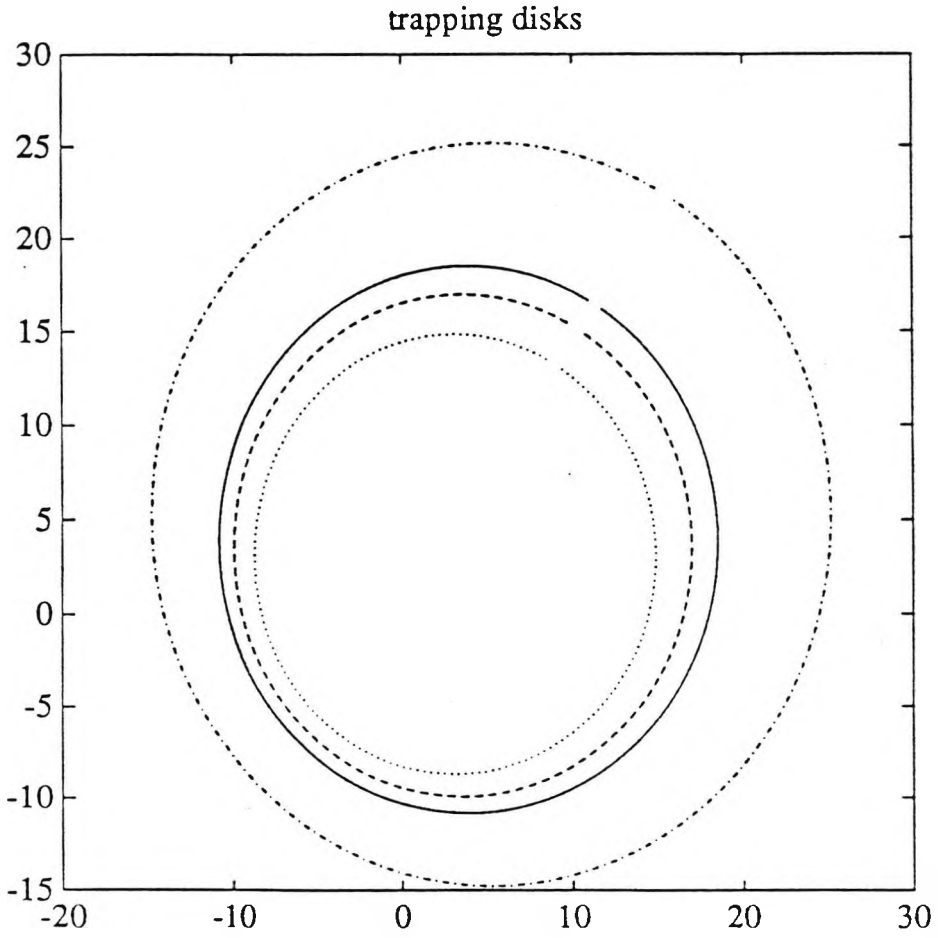
In the sequel, some examples illustrating the above technique are presented.

Example (8.8): Let us consider again Example (8.4). After programming algorithms **ALMZERO** and **TRAPDISK** [Mit. & Kar., 1], the following results are achieved:

scaling	almost zero=σ,w)	zero radius
diag{1,1,1}	(-0.9999, 0)	14.658 (x_{min}=3.81)
normalization	(0.02179, 0)	13.466 (x_{min}=3.5)
B-scaling	(0.2289, 0)	13.312 (x_{min}=3.46)
$\|\ \|_\infty$ row scaling	(0.434, 0)	13.158 (x_{min}=3.42)
diag{1,1,100}	(1.7943, 0)	11.773 (x_{min}=3.06)
diag{500,1,1}	(-2.0, -0.99999)	19.9676(x_{min}=5.19)

Matrix 8.4

Next, a figure showing the above defined trapping disks is presented.



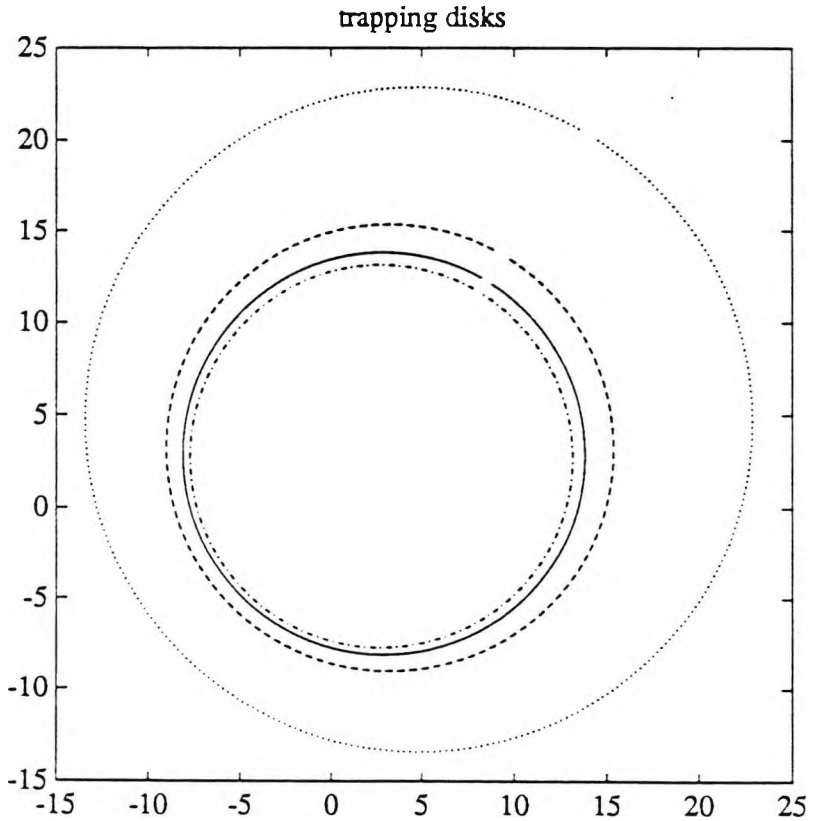
■

Example (8.9): Let us consider again Example (8.5). Applying consequently the algorithms **ALMZERO** having initial guess of the almost zero equal to $(-\frac{p}{3}, -\frac{p}{3})$ and **TRAPDISK**, the following results are achieved:

scaling	almost zero z=(σ,w)	zero radius
diag{1,1,1}	(-1.5463, 0)	12.196(x_{min}=3.17)
normalization	(-1.2077, 0)	10.965(x_{min}=2.85)
B-scaling	(-1.04189, 0)	10.426(x_{min}=2.71)
$\|\ \|_\infty$ row scaling	(-1.2258, 0)	11.042(x_{min}=2.87)
diag{100,1,1}	(-2.99865, 0)	18.159(x_{min}=4.72)

Matrix 8.5

Next, the above defined trapping regions are presented in a figure.



■

Example (8.10): Let the set of polynomials be defined by the polynomial vector $\underline{p}(s)$:

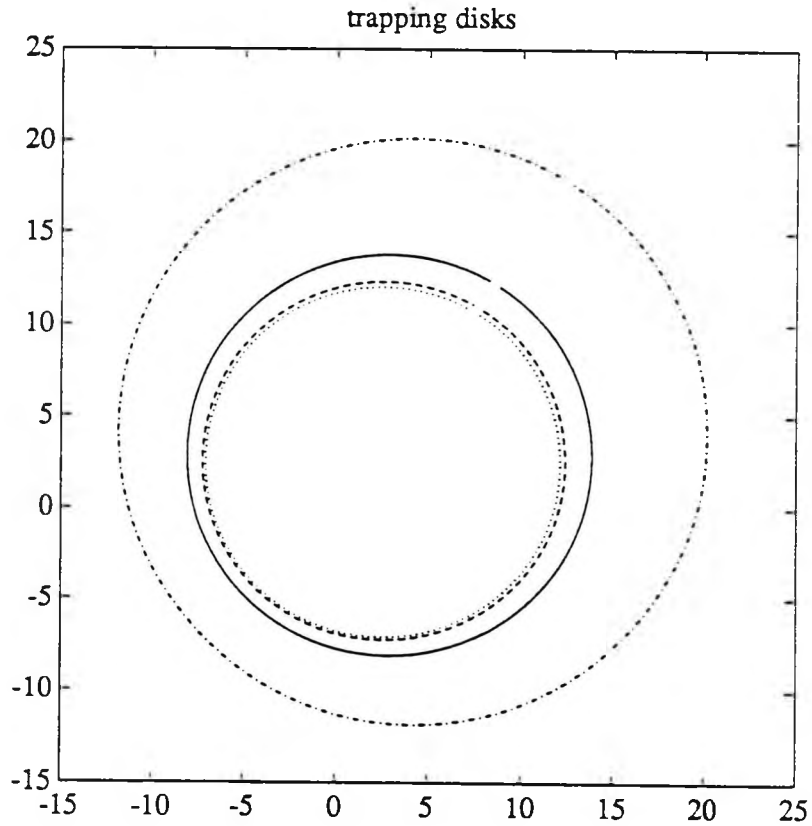
$$\underline{p}(s)=\begin{bmatrix} s-2 \\ s^3+5s^2+9s+5 \end{bmatrix}=\begin{bmatrix} -2 & 1 & 0 & 0 \\ 5 & 9 & 5 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \end{bmatrix}=P_2\cdot\underline{e}_3(s),\ P_2\epsilon R^{2x4}$$

After the application of algorithms **ALMZERO** with initial guess of almost zero $(-\frac{p}{3},-\frac{p}{3})$ and **TRAPDISK** the following matrix is obtained:

scaling	almost zero s=(σ,w)	zero radius
diag{1,1}	(-0.7057, 0)	10.965 (x_{min}=2.85)
normalization	(0.0473, 0)	9.811 (x_{min}=2.55)
B-scaling	(0.3267, 0)	9.58 (x_{min}=2.49)
$\|\ \|_\infty$ row scaling	(0.2083, 0)	9.6953(x_{min}=2.52)
diag{0.001,1}	(-1.999999,-0.9999)	16.005 (x_{min}=4.16)

Matrix 8.6

The following figure, illustrates the position of the above defined trapping disks.



■

General Remarks

From the above examples we remark that amongst the various applied scalings those giving lower values for the zero radius are normalization, B-scaling and $\| \|_\infty$ row scaling. Of course some scalings chosen randomly gave low values too, but since we can not define precisely such kind of scalings we do not consider them as important in our effort to specify a small area enclosing a zero of the given polynomial combinant. Therefore, we claim that the intersection of the circles having as their centres and radii the almost zeros and the zero radii corresponding to the normalized, B-scaled and $\| \|_\infty$ row scaled original set of polynomials, is a region containing a zero of the combinant $f(\underline{k},\underline{p}(s))$. We must point out that due to the method used for the determination of the zero radius, this area is not very small and thus the determination of the zero is not immediate. However, this area is considered to be quite satisfactory because any other disk defined by random scaling overlays this area.

8.4 ALMOST ZEROS AND DYNAMIC COMBINANTS

So far, we have examined fixed polynomial combinants of the form :

$$f(\underline{k},\underline{p}(s))=\sum_{i=1}^{m} k_i p_i(s) \ , \quad \underline{k}\epsilon R^m.$$

If instead of real vectors $\underline{k}\epsilon R^m$ we use polynomial vectors $\underline{k}(s)\epsilon R^m[s]$, then the notion of a dynamic polynomial combinant $f(\underline{k}(s),\underline{p}(s))$ is defined, which is discussed next.

Let $P=\{p_i(s)\epsilon R[s],\ i\epsilon\underline{m},\ d=\max\{\deg\{p_i(s)\}\}$ be a set of polynomials. Let $\underline{p}(s)\epsilon R^m[s]$ be the associated to p polynomial vector. It is known that $\underline{p}(s)$ can be expressed as:

$$\underline{p}(s)=\begin{bmatrix} p_1(s) \\ \vdots \\ p_m(s) \end{bmatrix}=[\underline{p}_0,\underline{p}_1,\ldots,\underline{p}_d]\begin{bmatrix} 1 \\ s \\ \vdots \\ s^d \end{bmatrix}=P_{m,d}^{0}\cdot\underline{e}_d(s) \qquad (8.25)$$

Let also $K=\{k_i(s)\epsilon R[s],\ i\epsilon m,\ v=\max\{\deg\{k_i(s)\}\}\}$ be a polynomial set. The polynomial vector associated to K, $\underline{k}(s)$, can be expressed as:

$$\underline{k}^t(s)=[k_1(s),\ldots,k_m(s)]=s^v\underline{k}_v^t+s^{v-1}\underline{k}_{v-1}^t+\ldots+s\underline{k}_1^t+\underline{k}_0^t \qquad (8.26)$$

The v-th order combinant $f_v(\underline{k}(s),\underline{p}(s))$ of a polynomial set P is defined in the following way:

$$f_v(\underline{k}(s),\underline{p}(s))=\underline{k}^t(s)\underline{p}(s) = \sum_{i=1}^{m} k_i(s)p_i(s)=\underline{k}_0^t\underline{p}(s)+s\underline{k}_1^t\underline{p}(s)+\ldots+s^v\underline{k}_v^t\underline{p}(s) \qquad (8.27)$$

$f_v(\underline{k}(s),\underline{p}(s))$ can be also expressed as follows:

$$f_v(\underline{k}(s),\underline{p}(s))=[\underline{k}_0^t,\underline{k}_1^t,\ldots,\underline{k}_v^t]\begin{bmatrix}\underline{p}(s)\\ s\underline{p}(s)\\ \vdots\\ s^v\underline{p}(s)\end{bmatrix}=\underline{k}_{m,v}^t\cdot\underline{p}^v(s) \qquad (8.28)$$

Definition (8.4): The polynomial set P^v defined by the vector

$$\underline{p}^v(s)=\begin{bmatrix}\underline{p}(s)\\ s\underline{p}(s)\\ \vdots\\ s^v\underline{p}(s)\end{bmatrix}=\begin{bmatrix}\underline{p}_0 & \underline{p}_1 & \cdots & \underline{p}_d & \underline{0} & \cdots & \underline{0}\\ \underline{0} & \underline{p}_0 & \underline{p}_1 & \cdots & \underline{p}_d & \cdots & \underline{0}\\ \vdots & \ddots & \ddots & & \ddots & & \vdots\\ \underline{0} & \cdots & \underline{p}_0 & \underline{p}_1 & & \underline{p}_d & \end{bmatrix}\cdot\begin{bmatrix}1\\ s\\ \vdots\\ s^{v+d}\end{bmatrix} \equiv P_{m,d}^v\cdot\underline{e}_{v+d}(s) \qquad (8.29)$$

is called the <u>v-th order set of P</u> and $P_{m,d}^v$ is the v-th Toeplitz of $P_{m,d}$. ■

Remark (8.4): The zero distribution properties of the v-th order combinant may be equivalently studied as a zero distribution of a constant combinant defined on the v-th order of P, P^v. Thus, the matrix $P_{m,d}^v$ will be mostly used in our study concerning the properties and applications of a given v-th order combinant $f_v(\underline{k}(s), \underline{p}(s))$ of a polynomial set P. ■

For a given polynomial set P, a first question arising when we are dealing with the v-th order sets of P, concerns their zero assignability or zero nonassignability.

More specifically, when we are given a strongly zero nonassignable set P it is extremely interesting to examine under what circumstancies the derived v-order sets of P will remain nonassignable or they will become zero assignable. In

the sequel, conditions providing the minimal order v_{min} that guarantee the zero assignability or the zero nonassignability of them are examined.

8.4.1 Conditions for zero assignability of fixed order dynamic combinants

Assume P to be strongly zero nonassignable. Then, the basis matrix $P_{m,d}$ can be expressed in the following form:

$$P_{m,d} = [\underline{p}_0, \underline{p}_1, \ldots, \underline{p}_d] \epsilon R^{mx(d+1)}$$

Matrix $P_{m,d}$ can also be written as:

$$P_{m,d} = [\underline{P}_0, \bar{P}], \text{ where } \bar{P} = [\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_d] \epsilon R^{mxd}$$

According to Proposition (8.5) the set P is strongly zero nonassignable if and only if $N_1\{\bar{P}\}=\{0\}$ or equivalently $\rho(\bar{P})=m$, $m \leq d$. Let

$$P^{v}_{m,d} = \begin{bmatrix} \underline{P}_0 & \underline{P}_1 & \ldots & \underline{P}_d & \underline{0} & \ldots & \underline{0} \\ \underline{0} & \underline{P}_0 & \underline{P}_1 & \ldots & \underline{P}_d & \ldots & \underline{0} \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \underline{0} & \ldots & \ldots & \ldots & \underline{P}_0 & \underline{P}_1 \ldots & \underline{P}_d \end{bmatrix} \epsilon R^{m(v+1)x(d+v+1)} \qquad (8.30)$$

be the basis matrix of the v-order set of P. The next Proposition readily follows from Proposition (8.5).

Proposition (8.7): Let P be a given strongly zero nonassignable polynomial set. The v-order set of P is completely zero assignable if and only if

$$\rho(P^{v}_{m,d}) = d+v+1 \qquad (8.31)$$

Thus, a necessary condition providing the zero assignability of a v-order polynomial set is

$$d+v+1 \leq m(v+1)$$

Proposition (8.8): If P is a coprime polynomial set, then there exists v such that P^v is zero assignable.

■

The proof of the above Proposition follows from the solvability of the Diophantine equation [Kuc., 1]

$$k_1(s)p_1(s)+\ldots+k_m(s)p_m(s)=\varphi(s) \tag{8.32}$$

where $\varphi(s)$ is arbitrary polynomial.

An immediate consequence of Proposition (8.8) is the subsequent Corollary.

Corollary (8.2): The set P is coprime, if and only if there exist indices v such that

$$\rho(P^{v}_{m,d}) = d+v+1$$

■

Coprimeness thus guarantees the existence of indices v for which we have assignement.

Corollary (8.3): Let v be an integer for which P^v is zero assignable. Then,

$$v \geq \frac{d}{m-1} - 1 \tag{8.33}$$

■

The above Corollary defines the range of values v for which dynamic assignability may hold. An obvious question arising is whether we can define the minimal value of v, v_{min} for which assignability is guaranteed. By testing ranks of $P^{i}_{m,d}$ this may be determined algorithmically. In the sequel, the following example demonstrates the above technique.

Example (8.11): Let

$$P_{3,3} = \begin{bmatrix} 3 & 1 & 3 & 1 \\ 1 & 2 & 1 & 2 \\ 3 & -3 & -2 & 2 \end{bmatrix} \text{ be a basis matrix } \epsilon R^{3x4}$$

$$\rho\left(\begin{bmatrix} 1 & 3 & 1 \\ 2 & 1 & 2 \\ -3 & -2 & 2 \end{bmatrix}\right) = 3 \leq d = 3, \text{ thus we have a set of strongly zero}$$

nonassignable polynomials.

For an integer $v \geq \frac{d}{m-1} - 1 = 1$ the set may be zero assignable. Actually, for v=1

$$P^{1}_{3,3} = \begin{bmatrix} 3 & 1 & 3 & 1 & 0 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 \\ 3 & -3 & -2 & 2 & 0 & 0 \\ 0 & 3 & 1 & 3 & 1 & 0 \\ 0 & 1 & 2 & 1 & 2 & 0 \\ 0 & 3 & -3 & -2 & 2 & 0 \end{bmatrix} \quad \text{has} \quad \rho(P^{1}_{3,3}) = 5$$

and therefore $v_{min}=1$.

■

An obvious issue arising is whether some of the properties of $P_{m,d}$ may be adequate to predict v_{min}, without going to the algorithmic procedure of the rank tests of $P^{v}_{m,d}$. A given $P^{v}_{m,d}$ of the form (8.30) will be zero assignable if its d+v+1 columns are linear independent that is if it does not exist vector $\underline{k} \epsilon R^{d+v+1}$, $\underline{k} \neq \underline{0}$ such as $P^{v}_{m,d} \cdot \underline{k} = \underline{0}$. Thus, our problem is transfered to the following equivalent one. For a specific $P_{m,d} = [\underline{P}_0, \underline{P}_1, \ldots, \underline{P}_d]$ can we specify the minimum integer $v_{min} \geq \frac{d}{m-1} - 1$, which provides the existence of a vector $\underline{k} \epsilon R^{d+v_{min}+1}$ satisfying:

$$\begin{aligned} &\sum_{i=0}^{d} k_i \, \underline{p}_i = 0 \\ &\sum_{l=j}^{j+d} k_l \, \underline{p}_{l-j} = 0, \quad j=1,\ldots,v_{min}. \end{aligned} \tag{8.34}$$

The above case is still under consideration.

8.4.2 Conditions for strong nonassignability of fixed order dynamic combinants

According to 8.4.1 if we assume that P is a strongly zero nonassignable set, then $P_{m,d}=[\underline{P}_0, \bar{P}]$ has $\rho(\bar{P})=m \leq d$. If we now consider

$$P^{v}_{m,d} = \begin{bmatrix} \underline{P}_0 & \underline{P}_1 & \underline{P}_2 & \cdots & \underline{P}_d & \underline{0} & \cdots & \underline{0} \\ \underline{0} & \underline{P}_0 & \underline{P}_1 & \cdots & \cdots & \underline{P}_d & \cdots & \underline{0} \\ \cdot & & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & \cdot & \cdot & \cdot \\ \underline{0} & \cdots & \cdots & \underline{0} & \underline{P}_0 & \underline{P}_1 & \cdots & \underline{P}_d \end{bmatrix} = [\underline{P}^{v}_0 \ \bar{\underline{P}}^{v}] \epsilon R^{m(v+1)x(d+v+1)} \qquad (8.35)$$

Then for all v such that

$$\rho(\bar{P}^{v}) = m(v+1) \leq (d+v) \qquad (8.36)$$

the set P^{v} is strongly nonassignable and thus the almost zeros act as centre of discs of dynamic combinants up to the order v.

From (8.36) we have:

$mv+m \leq d+v$ or $(m-1)v \leq d-m$ from which we derive that $v \leq \frac{d-m}{m-1} = \frac{d-1}{m-1} - 1$

Corollary (8.4): Necessary condition for dynamic strong nonassignability is that

$$v \leq \frac{d-m}{m-1} = \frac{d-1}{m-1} - 1 \qquad (8.37)$$

■

The above condition is not sufficient since $m(v+1) \leq (d+v)$ does not imply $\rho(\bar{P}^{v}) = m(v+1)$. However, it provides a range of values of v for which the rank of $P^{v}_{m,d}$ should be tested.

Example (8.12):

Let $P_{2,3} = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & 1 & 4 \end{bmatrix}$ be a basis matrix ϵR^{2x4}

(i) The possible values of indices for which v-dynamic nonassignability holds are:

$$0 \leq v \leq \frac{d-1}{m-1} - 1 = 1$$

For such values, $\rho(\bar{P}^{v})$ has to be tested for the $\rho(\bar{P}^{v}) = m(v+1)$ property. Indeed,

for v=0

$\rho\left(\begin{bmatrix} 0 & 2 & 3 \\ 1 & 1 & 4 \end{bmatrix}\right)=2 \leq d=3$ and thus $P^{0}_{2,3}$ is nonassignable.

For v=1 $P_{2,3}=\begin{bmatrix} 1 & 0 & 2 & 3 & 0 \\ 0 & 1 & 1 & 4 & 0 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$, $\rho(\bar{P}^{1}) = \rho\left(\begin{bmatrix} 0 & 2 & 3 & 0 \\ 1 & 1 & 4 & 0 \\ 1 & 0 & 2 & 3 \\ 0 & 1 & 1 & 4 \end{bmatrix}\right) = 4$

and therefore P^{1} is nonassignable.

The following values of almost zeros and zero radii were achieved

v	almost zero z=(σ,w)	zero radius
0	(0.09069, -0.498092)	2.732 (x_{min}=0.71)
1	(0.09010, -0.490844)	148.93 (x_{min}=28.179)

Matrix 8.7

(ii) The range of integer values v for which dynamic assignability may hold is defined by:

$$v \geq \frac{d}{m-1} - 1 = 2$$

For v=2 $P^{2}_{2,3} = \begin{bmatrix} 1 & 0 & 2 & 3 & 0 & 0 \\ 0 & 1 & 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 2 & 3 & 0 \\ 0 & 0 & 1 & 1 & 4 & 0 \\ 0 & 0 & 1 & 0 & 2 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 \end{bmatrix}$ has $\rho(P_{2,3}) = 6$ and thus $v_{min}=2$.

■

Example (8.13): Let

$P_{3,7} = \begin{bmatrix} 1 & 0 & 2 & 0 & 0 & 6 & 4 & 0 \\ 0 & 2 & 3 & 1 & 2 & 0 & 3 & 0 \\ 0 & 1 & 5 & 1 & 0 & 2 & 0 & 1 \end{bmatrix}$ be a basis matrix ϵR^{3x8}

(i) The possible values of indices for which v-dynamic nonassignability holds are:

$$0 \leq v \leq \frac{d-1}{m-1} - 1 = 2$$

For such values, $\rho(\bar{P}^v)$ has to be tested for the $\rho(\bar{P}^v)=m(v+1)$ property. Indeed, for v=0

$$\rho\left(\begin{bmatrix} 0 & 2 & 0 & 0 & 6 & 4 & 0 \\ 2 & 3 & 1 & 2 & 0 & 3 & 0 \\ 1 & 5 & 1 & 0 & 2 & 0 & 1 \end{bmatrix}\right) = 3 \leq d = 7$$ and thus $P^0_{3,7}$ is nonassignable.

For v=1, $\rho(\bar{P}^{-1}_{3,7})=6$ and therefore $P^1_{3,7}$ is nonassignable.

For v=2, $\rho(\bar{P}^{-2}_{3,7})=9$ and evidently $P^2_{3,7}$ is nonassignable
The following values of almost zeros and zero radii were achieved.

v	almost zero z=(σ,w)	zero radius
0	$(0.159\cdot10^{-12}, -0.50578\cdot10^{-12})$	0.6725 (x_{min}=0.7)
1	$(0.244\cdot10^{-11}, -0.60694\cdot10^{-11})$	7.9827 (x_{min}=0.7225)
2	$(0.452\cdot10^{-12}, -0.10228\cdot10^{-11})$	65.982 (x_{min}=5.2825)

Matrix 8.8

(ii) The range of integer values v for which dynamic assignability may hold is defined by:

$$v \geq \frac{d}{m-1} - 1 = 3$$

Actually for v=3, $\rho(P^3_{3,7})=11$ and therefore $v_{min}=3$.

■

Assuming that P^i, i=0,1,2,...,v are strongly nonassignable sets, the following questions arise:

(i) What is the relationship between their almost zeros, as we go from smaller to larger values of i.

(ii) What is the behaviour of the radii of zero trapping discs as we go from smaller to higher i.

From Examples (8.12) and (8.13) we remark that the almost zeros of P^i, i=0,1,2,...,v are subjected to very slight changes. On the contrary, the zero radius of the trapping discs grows tremendously as the i increases, which indicates increase in our ability to assign the zeros, when we increase the dynamic order of combinants.

8.4.3 The almost zero generating function

In the above discussion about the v-th order sets of P, the vector

$$\underline{p}^v(s) = \begin{bmatrix} \underline{p}(s) \\ s\underline{p}(s) \\ \vdots \\ s^v\underline{p}(s) \end{bmatrix}$$

was mainly used. The same vector leads us to formulate the following definition.

Definition (8.5): For a given integer v and for some seC, the v-th order almost zero's generating function is defined by

$$\varphi^v(s) = \underline{p}^{vt}(s^*)\underline{p}(s) = \varphi^0(s)(1+s^*s+\ldots+s^{*v}s^v), \qquad (8.38)$$

where $\varphi^0(s)=\underline{p}^t(s^*)\underline{p}(s)$

■

From (8.38) it is evident that for each integer value of v the generating function is modified by a mere multiplication by the function:

$$(1+s^*s+\ldots+s^{*v}s^v)$$

Based on Definition (8.5), a separate study of the almost zero properties of an v-th order polynomial set P^v can be developed.

8.5 CONCLUSIONS

The aim of this Chapter was to develop numerical techniques arising from the almost zero definition. Several properties of the almost zero were summarized and the new notions of B-scaled, normalized, $\|\ \|_\infty$ row-scaled almost zero were analytically introduced. Therefore, the Chapter serves the following purposes:

(i) It provides an efficient algorithm for computing the prime almost zero.

(ii) It provides a detailed study concerning the sensitivity of almost zero to scaling.

(iii) It provides an algorithm for computing an upper bound for the zero radius. Hints for improving this bound by using the sensitivity of almost zero to scaling are also given.

(iv) It provides a formulation for the Dynamic Combinants.

CHAPTER 9

THE COMPUTATIONAL FRAMEWORK OF THE DETERMINANTAL ASSIGNMENT PROBLEM

9.1 INTRODUCTION

The Determinantal Assignment Problem (DAP) has emerged as the common formulation of a variety of Control Theory problems such as the pole, zero assignment problems [Kar. & Gia., 1,2], [Gia. & Kar., 1] under different types of compensation (feedback). This problem (DAP) is of a multilinear nature and it is naturally reduced to a linear problem of zero assignment of polynomial combinants and a standard multilinear problem, that of decomposability of multivectors [Marc., 1]. The solvability of DAP is thus reduced to the solvability of a set of linear equations (characterising the linear subproblem) together with a set of quadratics (known as Quadratic Plucker Relations (QPR) which characterize the decomposability of multivectors [Marc., 1], [Kar. & Gia., 2]. Classical algebraic geometry [Hod. & Ped., 1] (in a projective, rather than affine space) is used to determine the existence of solutions [Gia. & Kar., 1], [Kar. & Gia., 2]. The approach heavily relies on exterior algebra and this has implications on the computability of solutions (reconstruction of solutions, whenever they exist) as well as on the introduction of new sets of invariants (of a projective character), which characterize the solvability of DAP.

The main advantages of the DAP approach are that it provides the means for computing solutions, it handles both generic and exact solvability investigations and it finally introduces new criteria for the characterization of solvability of different problems.

The aim of this Chapter is to develop the numerical framework, as well as a numerical algorithm for the computation of solutions of DAP. This algorithm may be used as a basis of a design technique centered around the frequency assignment problems.

Some of the important numerical techniques developed in Chapter 4 are applied in the present Chapter. More specifically, the algorithms concerning the evaluation of compounds of real matrices and the computation of Plucker matrices are mostly used for the specification of a computational framework for the formulation of a unifying algorithm solving DAP . The computation of solutions of DAP is reduced to an optimization problem of a function with quadratic equality constraints. A convenient algorithm appropriate for the computation of solutions of DAP is presented. Finally an analytical example demonstrating the application of the above technique is given.

9.2 THE DETERMINANTAL ASSIGNMENT PROBLEM FOR LINEAR SYSTEMS

Linear Control Theory problems such as those of pole assignment by state, output feedback (centralised or decentralised) , and zero assignment by input, or output squaring down, may be reduced to a standard common problem known as the Determinantal Assignment Problem (DAP) [Kar. & Gia., 1], which is defined below:

The Determinantal Assignment Problem

Let $M(s)\epsilon R^{p\times q}_{[s]}$, $q\leq p$, and let $\rho_{R(s)}(M(s)) = q$ and let $H=\{H:H\epsilon R^{q\times p},\ \rho(H)=q\ \}$. Finding an $H\epsilon H$ such that the polynomial:

$$f_M(s,H) = \det(HM(s)) \tag{9.1}$$

has a given set of zeros, has been defined as the Determinantal Assignment Problem (DAP). If $\underline{h}_i{}^t$, $\underline{m}_i(s)$, $i\epsilon\underline{q}$, denote the rows of H, columns of M(s) respectively, then

$$C_q(H) = \underline{h}_i{}^t\wedge\ldots\wedge\underline{h}_q{}^t = \underline{h}^t{}_\wedge\ \epsilon R^\sigma$$

and $C_q(M(s)) = \underline{m}_i(s)\wedge\ldots\wedge\underline{m}_q(s) = \underline{m}(s)_\wedge\ \epsilon R^\sigma[s]$, $\sigma = \binom{p}{q}$

and by the Binet-Cauchy theorem [Marc. & Minc., 1] we have that

$$f_M(s,H)= C_q(H)C_q(M(s)) = <\underline{h}_\wedge,\ \underline{m}(s)_\wedge> = \sum_{\omega\epsilon Q_{q,p}} h_\omega\ m_{\underline{\omega}}(s) \tag{9.2}$$

where $<.,.>$ denotes the inner product, $\omega = (i_1,\ldots,i_q)\epsilon Q_{q,p}$, and h_ω, $m_\omega(s)$ are the coordinates of $\underline{h}_\wedge$, $\underline{m}(s)_\wedge$ respectively. Note that h_ω is the qxq minor of H which corresponds to the ω set of columns of H and thus $\underline{h}_\omega$ is a multilinear alternating function of the entries h_{ij} of H. The initial problem may be reduced to a linear subproblem and a multilinear subproblem as it is shown below.

(i) Linear subproblem of DAP : Set $\underline{m}(s)_\wedge = p(s)\ \epsilon R^\sigma[s]$. Determine whether there exists a $\underline{k}\epsilon R^\sigma$, $\underline{k}\neq\underline{0}$, such that:

$$f_M(s,\underline{k})=\underline{k}^t\underline{p}(s) = \Sigma\ k_ip_i(s) = a(s),\ i\epsilon\ \underline{\sigma},\ a(s)\epsilon\ R[s] \tag{9.3}$$

(ii) Multilinear subproblem of DAP : Assume that K is the family of solution vectors $\underline{k}$ of (9.3). Determine whether there exists

$H^t = [\ \underline{h}_1,\ldots,\underline{h}_q]$ where $H\epsilon R^{p\times q}$, such that:

$$\underline{h}_1\wedge\ldots\wedge\underline{h}_q = \underline{k},\quad \underline{k}\epsilon K \tag{9.4}$$

Polynomials defined by Eqn. (9.3) are linear combinations of polynomials of R[s] and they are called polynomial combinants [Kar. & Gia., 1]; the zero assignability of them provides necessary conditions for the solution of DAP. The solution of the multilinear problem is strongly related to the well known notion of decomposability of multivectors [Marc., 1] of exterior algebra. Note that notions and tools from exterior algebra play also an important role in the linear subproblem, since $f_M(s,\underline{k})$ is generated by the decomposable multivector $\underline{m}(s)\wedge$.

9.3 THE COMPUTATIONAL FRAMEWORK OF DAP

The approach adopted for the solution of DAP uses the notion of decomposability of multivectors [Marc., 1] and the geometry of Grassmann variety [Hod. & Ped., 1] in an essential way. The general reduction of DAP introduced before, leads to a numerical procedure for the computation of solutions, whenever such solutions exist. The computational framework and the general algorithm are discussed here.

Let $M(s)= [\underline{m}_1(s),\ldots,\underline{m}_q(s)]\epsilon R^{pxq}[s]$, $p\geq q$, $\sigma = \binom{p}{q}$,

$\rho_{R(s)}(M(s)) = q$, and assume that M(s) has no finite zeros. If we denote V_M = col-$sp_{R(s)}\{M(s)\}$, then $\underline{m}_i(s)\wedge\ldots\wedge\underline{m}_q(s) = \underline{m}(s)\wedge = \underline{g}(V_M)$ is Known as a Grassmann Representative (GR) of V_M [Kar. & Gia., 1].

$\underline{g}(V_M)$ uniquely characterises V_M (mod R(s)) and if $\delta = \deg\{\underline{g}(V_M)\}$, then we may write

$$\underline{g}(V_M) = P_\delta\cdot e_\delta(s),\ P_\delta\epsilon R^{\sigma x(\delta+1)},\ \underline{e}_\delta(s)=[1,s,\ldots,s^\delta]^t \qquad (9.5)$$

The basis matrix P_δ of $\underline{g}(V_M)$ is referred to as the Plucker matrix of V_M [Kar. & Gia., 1].

Let $a(s)\epsilon R[s]$ be an arbitrary polynomial with $\deg\{a(s)\} \leq \delta$. The DAP defined in Section 9.2 can be reduced to the following two problems.

(i) Linear subproblem of DAP: Set $\sigma = \binom{p}{q}$, $\underline{m}(s)^{\wedge} = C_q(M(s)) = \underline{p}(s) \epsilon R^{\sigma x 1}[s]$.

Find the conditions under which vector $\underline{k}$ exists such that:

$$\underline{k}^t \cdot \underline{p}(s) = a(s) \quad \text{or equivalently} \quad P_\delta{}^t \cdot \underline{k} = \underline{a},\ \underline{a} \epsilon R^{\delta+1} \tag{9.6}$$

(ii) Multilinear subproblem of DAP : Assume that the linear subproblem is solvable and that K is the family of solution vectors $\underline{k} \epsilon K$ which are decomposable i.e. satisfies the set of Quadratic Plucker Relations (QPR) [Marc., 1]. If such a vector $\underline{k}$ exists, determine an $H \epsilon R^{qxp}$, $\rho(H) = q$ such that $C_q(H) = c \cdot \underline{k}^t$, $c \epsilon R - \{0\}$.

From the above analysis it is clear that the solvability of DAP, as well as the computation of its solutions, is reduced to the problem of solving a system of $(\delta+1)$ linear equations (defined by (9.6)) together with the set of quadratics known as QPRs. The set of QPRs defines the Grassmann variety of the corresponding projective space [Hod. & Ped., 1], but the quadratics in this set are not algebraically independent. A minimal algebraically independent set has been defined in [Gia., Kal. & Kar., 1] and it is Known as Reduced QPRs (RQPRs); this set has $\sigma - q(p-q) - 1$ quadratics with σ parameters. The set of RQPRs may thus be used for the study of the computational aspects of DAP. The solution of linear and quadratic equations may be formulated as an optimization problem where the linear equations define a perfomance index to be minimized and the RQPRs, define equality constraints. The following theoretical algorithm illustrates the numerical issues involved in the computation of solutions of DAP.

An Optimization Algorithm for the study of DAP

For a given $M(s) \epsilon R^{pxq}[s]$ and $a(s) \epsilon R[s]$ and with the assumptions stated before, a numerical procedure for the computation of approximate solutions of DAP, (whenever such solutions exist [Kar. & Gia., 3]) ,may be formulated, the basic steps of which are:

Algorithm DAP

STEP 1: Derive the Plucker matrix $P_\delta \epsilon R^{\sigma x(\delta+1)}$ of M(s).

STEP 2: Produce a set of RQPRs

and express them in the form:

$g_i(\underline{k}) = \underline{k}^t Q_i \underline{k} = \underline{0}$, i=1,...,t, $t=\sigma-q(p-q)-1$

where $Q_i \epsilon R^{\sigma x \sigma}$ are appropriate matrices.

STEP 3: Minimize $f(\underline{k}) = \| P_\delta^t \cdot \underline{k} - \underline{a} \|^2$

subject to the constraints $g_i(\underline{k}) = \underline{k}^t Q_i \underline{k} = \underline{0}$, i=1,...,t

STEP 4: If $\underline{k}^*$ is a solution of **STEP 3**, then find $H \epsilon R^{qxp}$,

$\rho(H) = q$ such that

$C_q(H) = c \cdot \underline{k}^{*t}$, $c \epsilon R-\{0\}$

Alg: 9.1

In order to implement the main steps of algorithm **DAP** in an effective numerical manner, the following important computational problems must be solved first:

(P1) For a given matrix $H \epsilon R^{qxp}$, determine a convenient algorithm for the evaluation of the compound matrix

$$C_l(H) \epsilon R^{\binom{q}{l} x \binom{p}{l}}, \quad 1 \leq l \leq q$$

For the case when l=q this algorithm will evaluate the Grassmann product $C_q(H) \epsilon R^{1x\sigma}$ of H.

(P2) For a given polynomial matrix $M(s)=[\underline{m}_1(s),...,\underline{m}_q(s)] \epsilon R^{pxq}[s]$, $\rho_{R(s)}(M(s)) = q$, define an algorithm for the computation of its Grassmann vector and thus Plucker matrix i.e.

$$g(M(s))=C_q(M(s))=\underline{m}_1(s) \wedge ... \wedge \underline{m}_q(s) \epsilon R^{\sigma x 1}[s].$$

(P3) Specify the RQPRs for a vector

(P4) Produce an efficient algorithm for solving the minimization problem defined in **STEP 3**.

(P5) Determine a matrix $H \epsilon R^{qxp}$, $\rho(H) = q$ such that $C_q(H) = c\underline{k}^{*t}$, $c \epsilon R-\{0\}$, where $\underline{k}^*$ is a solution of the minimization problem defined in **STEP 3.**

Of course, the above problems are considered when DAP has at least a complex solution in a generic sense. The solvability conditions of DAP have been investigated in [Kar. & Gia., 3] and are summarized below:
(a) If q=1, or q=p-1 then the solution of DAP is defined by the solvability of the linear system (9.6), that is it is solvable for all a(s), deg{a(s)}=δ if and only if $\rho(P_\delta) = \delta+1$
(b) If p>q and $q \neq 1, p-1$, then DAP is solvable for any a(s) with deg{a(s)}=δ by a complex H, when $q(p-q) \geq \delta+1$ and $\rho(P_\delta) = \delta+1$

A variety of conditions for the existence of real solutions are given in [Kar. & Gia., 3]. The numerical problems associated with DAP are considered, when either (a) or (b) conditions are satisfied. In the case (a), the conditions guarantee the existence of a complex solution. The optimization formulation of the problem, defines in the latter case an approximate real solution. In fact, for the $\underline{k} \epsilon R^{\sigma}$ solution of the optimization problem there exists a number ε>0 and a vector $\underline{\varepsilon} \epsilon R^{\delta+1}$ for which

$$\min f(\underline{k}) = \varepsilon \quad \text{subject to the constraints} \quad g_i(\underline{k}) = \underline{0}, \; i=1,2,\ldots,t$$

$$\text{and} \qquad P_\delta^t \cdot \underline{k} = \underline{a} + \underline{\varepsilon}, \quad \underline{k}^t Q_1 \underline{k} = \underline{0}, \quad i=1,2,\ldots,t$$

Thus in this case, a matrix $H \epsilon R^{qxp}$, $\rho(H) = q$ may be found which is such that $\det(HM(s)) = (\underline{a}^t+\underline{\varepsilon}^t)\cdot\underline{e}_\delta(s)$ where the vector $\underline{a}+\underline{\varepsilon}$ is as close as possible (in the Euclidean sense) to the vector $\underline{a}$. It can also be proved [Mard., 1] that the zeros of the polynomial $(\underline{a}^t+\underline{\varepsilon}^t)\cdot\underline{e}_\delta(s)$ are in the neighbourhoods of the zeros of the polynomial $\underline{a}^t\cdot\underline{e}_\delta(s)$, if ε is small. Thus if ε is very small, the matrix $H \epsilon R^{qxp}$ found by the algorithm solves approximately the problem and places the zeros in small discs centered at the roots of the given a(s).

The basic problems (P1-P5) listed above are considered in detail next.

Computational problems (P1), (P2) were encountered in Chapter 4. Algorithms **COMREL** and **PLUCKER** developed analytically in sections 4.4 and 4.5 can be applied when the solution of (P1) or (P2) is required. Techniques dealing with the problems (P3),(P5) are fully developed in [Gia. ,1], [Kar. & Gia., 2] respectively. In [Kar. & Gia., 2] an alternative method for reconstructing the compensators, from their exterior products, using the Singular Value

Decomposition (computation of a basis for the right null space of a matrix), is given using the notion of Grassmann; the latter method leads to an algorithm.

Computational problem (P4) attracts a lot of attention. Since the minimization problem defined in **STEP 3** of algorithm **DAP** may not have a unique solution, the specification of an efficient algorithm able to solve it, is not an easy task. Already known optimization techniques concerning problems with quadratic constraints are not appropriate since they are very general. The facts that we do not know exactly an approximation of an initial solution and of the number of the required steps indicate that any hill-climbing algorithm may not be suitable for (P4).

An algorithm developed in [Mar., 1] has been used. In the sequel, we briefly present this algorithm.

9.4 A FIRST ORDER METHOD FOR SOLVING AN EQUALITY CONSTRAINED PROBLEM [Mar., 1]

9.4.1 Introduction

In this section a first order method is presented for solving the finite dimensional equality constrained problem

$$\min\{\, g^0(x) : g(x)=0 \,\} \qquad (P)$$
$$g^0(\cdot) : R^n \longrightarrow R, \quad g(\cdot) : R^n \longrightarrow R^m, \quad m<n$$

This method is iterative in nature, i.e. starting from a given point $x_0 \in R^n$ it constructs a sequence $\{x_i\}$ according to the rule: $x_{i+1} = x_i + a_i h_i$, where $h_i \in R^n$ is the search direction and $a_i \in R$ is the step length at iteration $i=0,1,\ldots$ It can be proved that, under reasonable assumptions, any accumulation point of the sequence x_i satisfies first order necessary optimality conditions for problem (P).

The proposed method has the following salient features:

(i) It constructs a single sequence of points $\{x_i\}$ converging directly to a solution of (P). No feasibility requirements are made on the x_i's.

(ii) The search direction h_i is made up of two components. The "horizontal" component is determined from the projection of the cost gradient on the hyperplane H approximating the feasible manifold $\Gamma=\{x \in R^n: g(x)=0\}$ to first order at x_i. The "vertical" component of the search direction is the vector of minimum length from x_i to the hyperplane H. Thus the "horizontal" component is a descent direction for the cost and the "vertical" component is a displacement from x_i towards the feasible manifold Γ.

(iii) A convenient stepsize is determined by a line search (along h_i) on the exact penalty function

$$\gamma(x,c) = g^0(x) + c \sum_{j=1}^{m} |g^j(x)| \qquad (9.7)$$

where $g^j(\cdot)$ is the jth constraint function (i.e. $g(x) = [g^1(x)...g^j(x)...g^m(x)]^t$) and c a real positive parameter.

(iv) A scheme for automatically increasing the parameter c in order to ensure exactness of the penalty function γ is also incorporated in the method.

This first order method attempts to exploit the stabilizing effect that the exact penalty function $\gamma(x,c)$ of (9.7), can produce, when used together with a simple iterative scheme on a linearization of the original problem at each point x_i. This linearization yields the search direction described in (ii).

9.4.2 The numerical algorithm

The following notation is required for the development of the algorithm.

Notation (9.1) : Given a function $f: R^n \longrightarrow R$ we denote by

(i) $\frac{\partial f(x)}{\partial x}$ the partial derivative of function f(x) at x.

(ii) $\nabla f(x)$ the gradient of f(x) at x. $\nabla f(x)$ is always treated as a column vector, hence: $\nabla f(x) = [\frac{\partial f(x)}{\partial x^1},...,\frac{\partial f(x)}{\partial x^n}]^t$ ■

Notation (9.2) : (i) Given a function $g: R^n \longrightarrow R^m$, we denote by $\frac{\partial g(x)}{\partial x}$ its Jacobian matrix at x. This is an mxn matrix whose i,j-th element is $\frac{\partial g^i(x)}{\partial x^j}$, hence:

$$\frac{\partial g(x)}{\partial x} = \begin{bmatrix} \nabla g^1(x)^t \\ \vdots \\ \nabla g^m(x)^t \end{bmatrix}$$

(ii) The Langrangian $L(.,.): R^n x R^m \longrightarrow R$ is defined by: $L(x,\lambda) = g^0(x) + g(x)^t \cdot \lambda$ ■

The numerical algorithm requires also as input data an initial approximation x_0 to the solution $x \epsilon R^n$ of the minimization problem and the constants $\beta>0$, $c_{-1}>0$, $\delta>0$, $\alpha\epsilon(0,1)$ and $n\epsilon(0,1)$. Moreover, Subprocedure B requires as input data three more constants, η, L and ε satisfying $0<\eta<<1$, $L>1$ and $0<\varepsilon<<1$.

Algorithm OPTIM

STEP 0: $i := 0$

STEP 1: $\lambda(x_i) := -\left[\frac{\partial g(x)}{\partial x} \frac{\partial g(x)}{\partial x}^t \right]^{-1} \frac{\partial g(x)}{\partial x} \nabla g^0(x)$

$$p^0(x_i) := \left[I - \frac{\partial g(x)}{\partial x}^t \left[\frac{\partial g(x)}{\partial x} \frac{\partial g(x)}{\partial x}^t \right]^{-1} \frac{\partial g(x)}{\partial x}\right] \nabla g^0(x)$$

$$\hat{h}(x_i) := -\frac{\partial g(x)}{\partial g(x)}^t \left[\frac{\partial g(x)}{\partial x} \frac{\partial g(x)}{\partial x}^t \right]^{-1} g(x)$$

STEP 2: Normalize the projected gradient $p^0(x_i)$ using either either Subalgorithm A or B (take $f(.)$ to be either $g^0(.)$ or $L(.,\lambda(x_i))$), to obtain the normalizing parameter $\omega(x_i)$

$h(x_i) := \hat{h}(x_i) - \omega(x_i) p^0(x_i)$

STEP 3: **if** $(c_{i-1}-\beta)\sum_{j=1}^{m} |g^j(x_i)| - \lambda(x_i)^t g(x_i) \geq 0$ **then**

$c_i := c_{i-1}$

else

$$c_i := \max\left\{ \frac{\beta + \lambda(x_i)^t g(x_i)}{\sum_{j=1}^{m} |g^j(x_i)|} , c_{i-1}+\delta \right\}$$

STEP 4: $\theta(x_i,c_i) := \nabla g^0(x_i)^t h(x_i) - c_i \sum_{j=1}^{m} |g^j(x_i)|$

if $\theta(x_i,c_i) = 0$ **then**

quit

STEP 5: Compute the smallest nonnegative integer $\kappa(x_i)$ such that:

$\gamma(x_i + n^{\kappa(xi)} h(x_i), c_i) - \gamma(x_i,c_i) \leq \alpha \cdot n^{\kappa(xi)} \theta(x_i,c_i)$

STEP 6: $x_{i+1} := x_i + n^{\kappa(xi)} h(x_i)$

$i := i+1$

Repeat **STEP 1**

Alg: 9.2

Subprocedure A (Armigo's Rule)

Compute the smallest nonnegative integer $\tau(x_i)$ such that:

$$f(x_i - n^{\tau(xi)} p^0(x_i)) - f(x_i) \leq -\alpha \cdot n^{\tau(xi)} \cdot \|p^0(x_i)\|^2$$

$$\omega(x_i) := n^{\tau(xi)}$$

Alg: 9.3

Subprocedure B (Quadratic Interpolation)

$v := f(x_i - p^0(x_i)) - f(x_i) + \|p^0(x_i)\|^2$

if $v < \varepsilon$ **then**

  $\omega(x_i) := L$

  quit

else

  $u := \dfrac{\|p^0(x_i)\|^2}{2v}$

if $u < l$ **then**

  $\omega(x_i) := l$

  quit

if $u > L$ **then**

  $\omega(x_i) := L$

  quit

if $l \leq u \leq L$ **then**

  $\omega(x_i) := u$

  quit

Alg: 9.4

The development of a better minimization algorithm suitable for the needs of algorithm **DAP** is still under consideration.

In the sequel, a numerical example illustrating the application of algorithm **DAP** is presented.

9.5 NUMERICAL EXAMPLE

Let $M(s)=\begin{bmatrix} -s & 1 \\ 1 & -s(s+1) \\ 0 & -(s+1) \\ -1 & 0 \end{bmatrix} \in R^{4x2}[s]$, $p=4$, $q=2$, $\sigma=\binom{p}{q}=6$, $\delta=3$ and let

$a(s)=s^3+3s^2+4s+2$ be a given polynomial. Applying algorithms **DAP** and **PLUCKER** we have:

STEP 1: The Plucker matrix P_3 of $M(s)$ is

$$P_3 = \begin{bmatrix} -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} \in R^{6x4}$$

The conditions $2(4-2)\geq 4$, $\rho(P_3)=4$ guarantee the solvability of DAP for any $a(s)$ with $\deg\{a(s)\}=3$ by a complex H.

STEP 2: For a vector $\underline{k}=[k_0,k_1,k_2,k_3,k_4,k_5]\in R^6$ the number of the RQPRs are:

$$n_{RQPRs} = \binom{4}{2}-2(4-2)-1 = 1$$

and specifically this relation is

$$k_1k_4-k_2k_3 = k_0k_5$$

This relation can be also be expressed as:

$\underline{k}^t \cdot Q \cdot \underline{k} = \underline{0}$ where

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in R^{6x6}$$

STEP 3: Using algorithm **OPTIM** we solve the following optimization problem

minimize $f(\underline{k}) = \| P_3^t \cdot \underline{k} - \underline{a} \|^2$

subject to $\underline{k}^t \cdot Q \cdot \underline{k} = 0$

A solution to the above problem is

$\underline{k}^* = (1.001,\ 1.998,\ 0.997,\ -501.006,\ 0.00459,\ 498.998)$

STEP 4: For the above k^* we specify the following matrix $H \in R^{2x4}$

$$H = \begin{bmatrix} 1.001 & 0 & 501.006 & -0.00459 \\ 0 & 1.001 & 1.001 & 0.997 \end{bmatrix}$$

In this case

$$P_3^t \underline{k} = \underline{a} + \underline{\varepsilon}\ ,\quad \underline{\varepsilon} = [0, -0.00259, -0.00559, 0.001]^t$$

Thus, algorithm **DAP** assigns as closed loop pole polynomial the polynomial

$$(\underline{a}^t + \underline{\varepsilon}^t)\underline{e}_3(s) = 1.001s^3 + 2.994s^2 + 3.997s + 2$$

whose roots

$$\{-0.996, -0.9975 + 1.0055i, -0.9975 - 1.0055i\}$$

are close to the desired poles $\{-1, -1+i, -1-i\}$.

■

9.6 CONCLUSIONS

The aim of this Chapter was to provide the means for the formulation of a computational framework suitable for the solution of the Determinantal

Assignment Problem (DAP). In this formulation the exterior algebra algorithms developed in Chapter 4 and concerning the computation of compound and Grassmann product of real matrices and the evaluation of Plucker matrices are principally used. This formulation requires also an efficient numerical algorithm solving an equality constrained optimization problem.

Thus, this Chapter serves the following purpose:

- It provides the determination of an appropriate algorithm for the computation of solutions of DAP, which as it was shown, may be formulated as a constrained optimization problem.

CHAPTER 10

CONCLUSIONS

The main aim of this thesis was to provide several numerical techniques for handling computational problems arising from Algebraic Control Theory. Basically, it contributes in developing numerical methods dealing with Exterior Algebra computations, nongeneric computations and computations involving the almost zero notion. More specifically, the present thesis has aimed in achieving a number of goals such as:

(i) Derivation of Exterior Algebra algorithms

The developed algorithms are classified in two categories. The first category deals with computations concerning real vectors and matrices and produces techniques for the evaluation of the exterior product of vectors and of the compounds of matrices. The computation of the Decentralization characteristics [Kar., Lai. & Gia.,1] using the notion of compounds is being examined for the moment.

The second category deals with computations concerning polynomial vectors and matrices and produces techniques for the evaluation of polynomial compounds and Plucker matrices. The evaluation of Plucker matrices can be applied for the formulation of a computational framework for the Determinantal Assignment Problem (DAP). [Kar. & Gia, 1]. The whole issue is under consideration. The improvement of the proposed algorithm for the solution of DAP is still under consideration. In this area, research should be focussed in the development of more efficient hill climbing optimization methods, as well as investigation of significance of approximate solutions of DAP. Extension of the present framework of DAP to the case of Decentralized and Dynamic DAP is also feasible.

The material developed in this area of work provides the essential tools for the development of a Computer Aided Design environment for the solution of Frequency Assignment Problems such as pole assignment by output feedback or precompensation and zero assignment by squaring down. For the development of such techniques the optimization algorithm part of the approach should be modified to handle also inequality constraints.

(ii) Derivation of efficient techniques for tackling nongeneric computations

Most of the proposed techniques are based on the new notions of ε-independent, numerically ε-dependent, strongly ε-dependent, fuzzy ε-dependent sets of vectors. The most crucial technique, which in concrete cases will specify the termination of an algorithm , concerns the relationship between the numerical ε-rank of a strongly ε-dependent set and the singular values of its normalized matrix representation. Based on the above notions the

selection of a "best" representative of a strongly ε-dependent set is obtained.

An efficient technique for choosing a "best uncorrupted" base for the row space of a matrix is also developed. This technique in several nongeneric problems will allow the "catching up" of approximate solutions.

Furthermore, based on the definitions of numerical ε-rank and numerical ε-nullity, the notion of approximate null space can be introduced. A study concerning definitions , properties and general description of approximate null space is under consideration. The connection between approximate null space and numerical range is also investigated.

(iii) Derivation of a new numerical algorithm for the evaluation of the greatest common divisor of polynomials

This algorithm is really very useful because it provides the means for a convenient and efficient computation of the g.c.d. of any polynomial set $P_{m,d}$. The generalization of this algorithm so as to compute the right or left g.c.d. of matrices as well as polynomials solving the Diophantine equations are still under development.

Extension of the results to the case of matrix divisors is by no means trivial and requires additional effort.

An alternative algorithm for computing the g.c.d. and which is based on matrix pencil theory [Kar., 5] allows the possibility for defining almost zeros using the notion of numerical range and almost null space. This may allow the definition of almost zeros in a manner independent from the scaling of the polynomials.

(iv) Derivation of almost zero's algorithms

These algorithms are based on the existing almost zero definition [Kar., Gia. & Hub.,1]. From the comprehensive study of the almost zero's sensitivity to scaling, is derived that the almost zero's position is influenced from the position of the roots of the original polynomials. A new definition of a norm independent almost zero, based on the roots of the given polynomials is under consideration. Further research can also be done for a theoretical specification of the minimal value v_{min} of v for which a v-th order strongly zero nonassignable polynomial set P becomes zero assignable.

The topics examined in this thesis, by no means cover all the numerical issues faced in Algebraic Control Theory problems. As far as development of numerical methods for algebraic computations, the main obstacle in the matrix case is the reduction of the algebraic problem to a standard Linear Algebra

problem. In this area, important problems to be considered are efficient computations of common matrix divisors, common matrix multiples, Matrix Fraction Descriptions (MFDs), as well as factorization of rational matrices and construction of ordered minimal bases. Each of the above issues requires an appropriate real matrix formulation and subsequent definition of an efficient numerical algorithm. It should be pointed, that alternative means using tools from symbolic computations have to be examined.

APPENDIX A

Some numerical results achieved after the application of algorithm **COMREL** ,developed in section 4.4 of Chapter 4, are presented next.

Example (A.1):

THE GIVEN MATRIX A IS

2.00000	4.00000	3.00000	5.00000	7.00000
1.00000	7.00000	2.00000	1.00000	9.00000
1.50000	2.00000	3.00000	12.50000	1.00000
2.00000	3.00000	8.00000	6.00000	2.30000

EVALUATION OF THE P = 3 COMPOUND MATRIX OF A

ROWS OF A	COLUMNS OF A	ELEMENTS OF COMPOUND 1 ROW
1 2 3	1 2 3	8.50000
	1 2 4	84.50000
	1 2 5	-31.50000
	1 3 4	11.00000
	1 3 5	-12.50000
	1 4 5	-83.50000
	2 3 4	-83.50000
	2 3 5	52.00000
	2 4 5	207.50000
	3 4 5	-55.50000
ROWS OF A	COLUMNS OF A	2 ROW
1 2 4	1 2 3	51.00000
	1 2 4	7.00000
	1 2 5	-36.00000
	1 3 4	16.00000

1	3	5	-59.70000
1	4	5	3.10000
2	3	4	149.00000
2	3	5	113.10000
2	4	5	120.70000
3	4	5	209.90000

ROWS OF A

1	3	4

COLUMNS OF A			3 ROW
1	2	3	-8.50000
1	2	4	15.50000
1	2	5	.90000
1	3	4	-86.00000
1	3	5	35.45000
1	4	5	-73.75000
2	3	4	-216.50000
2	3	5	39.80000
2	4	5	-95.50000
3	4	5	-500.25000

ROWS OF A

2	3	4

COLUMNS OF A			4 ROW
1	2	3	-34.00000
1	2	4	87.00000
1	2	5	-4.05000
1	3	4	-44.00000
1	3	5	50.00000
1	4	5	-122.70000
2	3	4	-516.00000

2	3	5	52.10000
2	4	5	-71.85000
3	4	5	-691.40000

Example (A.2):

THE GIVEN MATRIX A IS

2.00000	4.10000	5.60000
4.50000	8.00000	10.50000
6.20000	2.60000	11.00000

EVALUATION OF THE P = 3 COMPOUND MATRIX OF A

THIS COMPOUND IS EQUAL WITH THE DETERMINANT

ROWS OF A	COLUMNS OF A	ELEMENTS OF COMPOUND 1 ROW
1 2 3	1 2 3	-26.88000

Example (A.3):

THE GIVEN MATRIX A IS

3.00000	4.00000	2.00000	1.00000	5.00000
6.00000	1.00000	2.00000	3.00000	4.00000
1.00000	2.00000	1.00000	2.00000	1.00000
2.00000	3.00000	4.00000	1.00000	3.00000
5.00000	6.00000	1.00000	3.00000	4.00000
1.00000	1.00000	1.00000	1.00000	1.00000

EVALUATION OF THE P = 5 COMPOUND MATRIX OF A

THIS COMPOUND IS A COLUMN VECTOR CALLED

THE GRASSMAN PRODUCT OF THE COLUMNS

	ELEMENTS OF COMPOUND
ROWS OF A 1 2 3 4 5 COLUMNS OF A 1 2 3 4 5	1 ROW -301.00000
ROWS OF A 1 2 3 4 6 COLUMNS OF A 1 2 3 4 5	2 ROW 10.00000
ROWS OF A 1 2 3 5 6 COLUMNS OF A 1 2 3 4 5	3 ROW 36.00000
ROWS OF A 1 2 4 5 6 COLUMNS OF A 1 2 3 4 5	4 ROW -105.00000
ROWS OF A 1 3 4 5 6 COLUMNS OF A 1 2 3 4 5	5 ROW 27.00000
ROWS OF A 2 3 4 5 6 COLUMNS OF A 1 2 3 4 5	6 ROW -4.00000

Example (A.4):

THE GIVEN MATRIX A IS

4.00000	2.00000	3.00000	1.00000
2.00000	1.00000	3.00000	2.00000
1.00000	1.00000	1.00000	1.00000

EVALUATION OF THE P = 3 COMPOUND MATRIX OF A

THIS COMPOUND IS A ROW VECTOR CALLED

THE GRASSMAN PRODUCT OF THE ROWS

ROWS OF A			ELEMENTS OF COMPOUND
1	2	3	
COLUMNS OF A			1 ROW
1	2	3	-3.00000
1	2	4	-3.00000
1	3	4	3.00000
2	3	4	3.00000

■

In the sequel, some examples attained after the application of algorithm **COMPOL1** ,developed in section 4.5 of Chapter 4, are presented.

Example (A.5):

$$M(s) = \begin{bmatrix} 3s+1 & 2s^2 \\ 5s+2 & 2s+3 \\ 3s+2 & 5s^2+1 \end{bmatrix}$$

THE COEFFICIENT OF POLYNOMIALS AUGMENTED MATRIX IS

.0000	2.0000	3.0000	.0000	1.0000	.0000
.0000	.0000	5.0000	2.0000	2.0000	3.0000
.0000	5.0000	3.0000	.0000	2.0000	1.0000

EVALUATION OF THE P= 2 COMPOUND MATRIX WHICH IS A COLUMN MATRIX

ROWS OF A		ELEMENTS OF COMPOUND
1	2	
PRIME COLUMNS OF A		1 ELEMENT
		(ADD THE NEXT TERMS)
1	2	.000*S** 4

1	4	.000*S** 3
1	6	.000*S** 2
2	3	-10.000*S** 3
2	5	-4.000*S** 2
3	4	6.000*S** 2
3	6	9.000*S** 1
4	5	2.000*S** 1
5	6	3.000*S** 0

FINAL ELEMENT

3.000*S** 0

11.000*S** 1

2.000*S** 2

-10.000*S** 3

ROWS OF A

---

1 3

PRIME COLUMNS OF A

---

2 ELEMENT

(ADD THE NEXT TERMS)

1	2	.000*S** 4
1	4	.000*S** 3
1	6	.000*S** 2
2	3	9.000*S** 3
2	5	1.000*S** 2
3	4	.000*S** 2
3	6	3.000*S** 1
4	5	.000*S** 1
5	6	1.000*S** 0

FINAL ELEMENT

1.000*S** 0

3.000*S** 1

1.000*S** 2

```
                                                          9.000*S** 3
-------------------------------------------------------------------
 ROWS OF A
 ---------
   2   3

 PRIME COLUMNS OF A                                        3 ELEMENT
 ------------------                                        --------
                                                  (ADD THE NEXT TERMS )
   1   2                                               .000*S** 4

   1   4                                               .000*S** 3

   1   6                                               .000*S** 2

   2   3                                             25.000*S** 3

   2   5                                             10.000*S** 2

   3   4                                             -6.000*S** 2

   3   6                                             -4.000*S** 1

   4   5                                             -4.000*S** 1

   5   6                                             -4.000*S** 0

                                                     FINAL ELEMENT

                                                     -4.000*S** 0

                                                     -8.000*S** 1

                                                      4.000*S** 2

                                                     25.000*S** 3
```

Example (A.6):

$$M(s) = \begin{bmatrix} 2s^2+3 & 8s+1 & 5s^2+9 \\ 2s+4 & 7 & 3s^2+1 \\ 4s+5 & 6s^2+2 & 4 \\ 6s+11 & 2s^2 & 15s+8 \end{bmatrix}$$

THE COEFFICIENT OF POLYNOMIALS AUGMENTED MATRIX IS

```
2.00    .00   5.00    .00   8.00    .00   3.00   1.00   9.00
 .00    .00   3.00   2.00    .00    .00   4.00   7.00   1.00
 .00   6.00    .00   4.00    .00    .00   5.00   2.00   4.00
 .00   2.00    .00   6.00    .00  15.00  11.00    .00   8.00
```

EVALUATION OF THE P= 3 COMPOUND MATRIX WHICH IS A COLUMN MATRIX

ROWS OF A			ELEMENTS OF COMPOUND
1	2	3	
PRIME COLUMNS OF A			1 ELEMENT
			(ADD THE NEXT TERMS)
1	2	3	-36.000*S** 6
1	2	6	.000*S** 5
1	2	9	-12.000*S** 4
1	3	5	.000*S** 5
1	3	8	-12.000*S** 4
1	5	6	.000*S** 4
1	5	9	.000*S** 3
1	6	8	.000*S** 3
1	8	9	52.000*S** 2
2	3	4	60.000*S** 5
2	3	7	66.000*S** 4
2	4	6	.000*S** 4
2	4	9	108.000*S** 3
2	6	7	.000*S** 3
2	7	9	198.000*S** 2
3	4	5	96.000*S** 4
3	4	8	-108.000*S** 3
3	5	7	120.000*S** 3
3	7	8	-138.000*S** 2
4	5	6	.000*S** 3
4	5	9	-32.000*S** 2
4	6	8	.000*S** 2
4	8	9	-220.000*S** 1
5	6	7	.000*S** 2

5	7	9	-88.000*S** 1
6	7	8	.000*S** 1
7	8	9	-176.000*S** 0
			FINAL ELEMENT
			-176.000*S** 0
			-308.000*S** 1
			80.000*S** 2
			120.000*S** 3
			138.000*S** 4
			60.000*S** 5
			-36.000*S** 6

ROWS OF A

---

1 2 4

PRIME COLUMNS OF A — 2 ELEMENT

(ADD THE NEXT TERMS)

1	2	3	-12.000*S** 6
1	2	6	.000*S** 5
1	2	9	-4.000*S** 4
1	3	5	.000*S** 5
1	3	8	.000*S** 4
1	5	6	.000*S** 4
1	5	9	.000*S** 3
1	6	8	210.000*S** 3
1	8	9	112.000*S** 2
2	3	4	20.000*S** 5
2	3	7	22.000*S** 4
2	4	6	.000*S** 4
2	4	9	36.000*S** 3
2	6	7	.000*S** 3
2	7	9	66.000*S** 2
3	4	5	144.000*S** 4

3	4	8	-192.000*S** 3
3	5	7	264.000*S** 3
3	7	8	-352.000*S** 2
4	5	6	-240.000*S** 3
4	5	9	-80.000*S** 2
4	6	8	-30.000*S** 2
4	8	9	-388.000*S** 1
5	6	7	-480.000*S** 2
5	7	9	-168.000*S** 1
6	7	8	255.000*S** 1
7	8	9	-546.000*S** 0

FINAL ELEMENT

-546.000*S** 0

-301.000*S** 1

-764.000*S** 2

78.000*S** 3

162.000*S** 4

20.000*S** 5

-12.000*S** 6

ROWS OF A

---

1 3 4

PRIME COLUMNS OF A

---

3 ELEMENT

(ADD THE NEXT TERMS)

1	2	3	.000*S** 6
1	2	6	180.000*S** 5
1	2	9	80.000*S** 4
1	3	5	.000*S** 5
1	3	8	.000*S** 4
1	5	6	.000*S** 4
1	5	9	.000*S** 3

1	6	8	60.000*S** 3
1	8	9	32.000*S** 2
2	3	4	-140.000*S** 5
2	3	7	-280.000*S** 4
2	4	6	.000*S** 4
2	4	9	-252.000*S** 3
2	6	7	270.000*S** 3
2	7	9	-384.000*S** 2
3	4	5	.000*S** 4
3	4	8	-60.000*S** 3
3	5	7	.000*S** 3
3	7	8	-110.000*S** 2
4	5	6	-480.000*S** 3
4	5	9	-64.000*S** 2
4	6	8	-60.000*S** 2
4	8	9	-116.000*S** 1
5	6	7	-600.000*S** 2
5	7	9	32.000*S** 1
6	7	8	15.000*S** 1
7	8	9	-146.000*S** 0
			FINAL ELEMENT
			-146.000*S** 0
			-69*S** 1
			-1186.000*S** 2
			-462.000*S** 3
			-200.000*S** 4
			40.000*S** 5

ROWS OF A

---

2 3 4

PRIME COLUMNS OF A	4 ELEMENT
	(ADD THE NEXT TERMS)
1 2 3	.000*S** 6
1 2 6	.000*S** 5
1 2 9	.000*S** 4
1 3 5	.000*S** 5
1 3 8	.000*S** 4
1 5 6	.000*S** 4
1 5 9	.000*S** 3
1 6 8	.000*S** 3
1 8 9	.000*S** 2
2 3 4	-84.000*S** 5
2 3 7	-168.000*S** 4
2 4 6	180.000*S** 4
2 4 9	52.000*S** 3
2 6 7	360.000*S** 3
2 7 9	104.000*S** 2
3 4 5	.000*S** 4
3 4 8	-36.000*S** 3
3 5 7	.000*S** 3
3 7 8	-66.000*S** 2
4 5 6	.000*S** 3
4 5 9	.000*S** 2
4 6 8	-360.000*S** 2
4 8 9	-36.000*S** 1
5 6 7	.000*S** 2
5 7 9	.000*S** 1
6 7 8	-405.000*S** 1
7 8 9	70.000*S** 0
	FINAL ELEMENT

-70.000*S** 0

-441.000*S** 1

-322.000*S** 2

376.000*S** 3

12.000*S** 4

-84.000*S** 5

■

Example (A.7):

$$M(s)=\begin{bmatrix} 2s+1 & 5s & 3s+4 & 8s \\ 5s+3 & 4s+3 & 2s & 3 \\ 4s & 8s & 3s+4 & 2s+1 \\ 2s+2 & 2s+4 & 4s+2 & 4s \\ 5 & 3s+2 & 8s & 5s+2 \end{bmatrix}$$

THE COEFFICIENT OF POLYNOMIALS AUGMENTED MATRIX IS

2.00000	5.00000	3.00000	8.00000	1.00000	.00000	4.00000	.00000
5.00000	4.00000	2.00000	.00000	3.00000	3.00000	.00000	3.00000
4.00000	8.00000	3.00000	2.00000	.00000	.00000	4.00000	1.00000
2.00000	2.00000	4.00000	4.00000	2.00000	4.00000	2.00000	.00000
.00000	3.00000	8.00000	5.00000	5.00000	2.00000	.00000	2.00000

EVALUATION OF THE P= 4 COMPOUND MATRIX WHICH IS A COLUMN MATRIX

ROWS OF A	PRIME COLUMNS OF A	ELEMENTS OF COMPOUND
1 2 3 4	1 2 3 4	1 ELEMENT (ADD THE NEXT TERMS) -376.000*S** 4
	1 2 3 8	-16.000*S** 3
	1 2 4 7	-156.000*S** 3
	1 2 7 8	-22.000*S** 2
	1 3 4 6	292.000*S** 3
	1 3 6 8	22.000*S** 2
	1 4 6 7	408.000*S** 2

1	6	7	8	28.000*S** 1
2	3	4	5	-400.000*S** 3
2	3	5	8	58.000*S** 2
2	4	5	7	-180.000*S** 2
2	5	7	8	-18.000*S** 1
3	4	5	6	136.000*S** 2
3	5	6	8	-58.000*S** 1
4	5	6	7	180.000*S** 1
5	6	7	8	-78.000*S** 0

FINAL ELEMENT

-78.000*S** 0

132.000*S** 1

422.000*S** 2

-280.000*S** 3

-376.000*S** 4

ROWS OF A

1	2	3	5

PRIME COLUMNS OF A

				2 ELEMENT
				(ADD THE NEXT TERMS)
1	2	3	4	-1075.000*S** 4
1	2	3	8	119.000*S** 3
1	2	4	7	500.000*S** 3
1	2	7	8	68.000*S** 2
1	3	4	6	650.000*S** 3
1	3	6	8	-70.000*S** 2
1	4	6	7	120.000*S** 2
1	6	7	8	-40.000*S** 1
2	3	4	5	-891.000*S** 3
2	3	5	8	153.000*S** 2
2	4	5	7	-4.000*S** 2

2	5	7	8	-68.000*S** 1
3	4	5	6	-157.000*S** 2
3	5	6	8	7.000*S** 1
4	5	6	7	-156.000*S** 1
5	6	7	8	36.000*S** 0

FINAL ELEMENT

36.000*S** 0

-257.000*S** 1

110.000*S** 2

378.000*S** 3

-1075.000*S** 4

ROWS OF A

---

1 2 4 5

PRIME COLUMNS OF A

---

3 ELEMENT

(ADD THE NEXT TERMS)

1	2	3	4	418.000*S** 4
1	2	3	8	-262.000*S** 3
1	2	4	7	-130.000*S** 3
1	2	7	8	-16.000*S** 2
1	3	4	6	-670.000*S** 3
1	3	6	8	280.000*S** 2
1	4	6	7	340.000*S** 2
1	6	7	8	160.000*S** 1
2	3	4	5	-22.000*S** 3
2	3	5	8	-32.000*S** 2
2	4	5	7	-150.000*S** 2
2	5	7	8	24.000*S** 1
3	4	5	6	-278.000*S** 2
3	5	6	8	-28.000*S** 1
4	5	6	7	150.000*S** 1

5 6 7 8	-144.000*S** 0
	FINAL ELEMENT
	-144.000*S** 0
	306.000*S** 1
	144.000*S** 2
	-1084.0000*S** 3
	418.000*S** 4

ROWS OF A

1 3 4 5

PRIME COLUMNS OF A

	4 ELEMENT
	(ADD THE NEXT TERMS)
1 2 3 4	590.000*S** 4
1 2 3 8	-98.000*S** 3
1 2 4 7	-152.000*S** 3
1 2 7 8	-20.000*S** 2
1 3 4 6	-672.000*S** 3
1 3 6 8	108.000*S** 2
1 4 6 7	112.000*S** 2
1 6 7 8	72.000*S** 1
2 3 4 5	-136.000*S** 3
2 3 5 8	28.000*S** 2
2 4 5 7	-248.000*S** 2
2 5 7 8	-4.000*S** 1
3 4 5 6	300.000*S** 2
3 5 6 8	-48.000*S** 1
4 5 6 7	328.000*S** 1
5 6 7 8	-100.000*S** 0
	FINAL ELEMENT
	-100.000*S** 0
	348.000*S** 1

280.000*S** 2

-1058.000*S** 3

590.000*S** 4

ROWS OF A

2 3 4 5

PRIME COLUMNS OF A

1	2	3	4
1	2	3	8
1	2	4	7
1	2	7	8
1	3	4	6
1	3	6	8
1	4	6	7
1	6	7	8
2	3	4	5
2	3	5	8
2	4	5	7
2	5	7	8
3	4	5	6
3	5	6	8
4	5	6	7
5	6	7	8

5 ELEMENT

(ADD THE NEXT TERMS)

-378.000*S** 4

398.000*S** 3

380.000*S** 3

50.000*S** 2

310.000*S** 3

-360.000*S** 2

-280.000*S** 2

-180.000*S** 1

-190.000*S** 3

100.000*S** 2

148.000*S** 2

-58.000*S** 1

-94.000*S** 2

160.000*S** 1

-228.000*S** 1

162.000*S** 0

FINAL ELEMENT

162.000*S** 0

-306.000*S** 1

-436.000*S** 2

898.000*S** 3

-378.000*S** 4

APPENDIX B

Some numerical results achieved after the application of algorithm **UNCBAS**, developed in section 5.7.3 of Chapter 5, are presented next.

Example (B.1):

THE GIVEN MATRIX IS

1.00000	2.00000	3.00000	4.00000	5.00000	6.00000
4.00000	3.00000	2.00000	1.00000	1.00000	3.00000
5.00000	5.00000	5.00000	5.00000	6.00000	9.00000
2.00000	2.00000	1.00000	3.00000	7.00000	4.00000
2.00000	4.00000	6.00000	8.00000	10.00000	12.00000

THE NORMALIZED MATRIX IS

.10483	.20966	.31449	.41931	.52414	.62897
.63246	.47434	.31623	.15811	.15811	.47434
.33942	.33942	.33942	.33942	.40731	.61096
.21953	.21953	.10976	.32929	.76835	.43906
.10483	.20966	.31449	.41931	.52414	.62897

SINGULAR VALUES

.2111042828477067701OE+01

.6600411509277606114 7E+00

.3283958821559913 5879E+00

.2452100642434581372 8E-14

.6872498051009429477 7E-15

THE NUMERICAL RANK OF THE MATRIX IS 3

THE GRAM MATRIX IS

1.00000	.71272	.95357	.92051	1.00000
.71272	1.00000	.89088	.65950	.71272
.95357	.89088	1.00000	.87925	.95357
.92051	.65950	.87925	1.00000	.92051
1.00000	.71272	.95357	.92051	1.00000

VALUE OF COMPOUND ELEMENT .075099298292068183968694 9

ROWS
2 4 5
VALUE OF COMPOUND ELEMENT .075099298292065519433435 8

ROWS
1 2 4
VALUE OF COMPOUND ELEMENT .031493254122479052981020 7

ROWS
2 3 4

```
VALUE OF COMPOUND ELEMENT    .013843188625265623326043 9

ROWS
 1   3   4
VALUE OF COMPOUND ELEMENT    .013843188625265012703380 3

ROWS
 3   4   5
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   4   5
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   2   5
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 2   3   5
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   2   3
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   3   5
```

THE ROW INDEPENDENT MATRIX IS

4.00000	3.00000	2.00000	1.00000	1.00000	3.00000
2.00000	2.00000	1.00000	3.00000	7.00000	4.00000
2.00000	4.00000	6.00000	8.00000	10.00000	12.00000

■

Example (B.2):

THE GIVEN MATRIX IS

3.000000	1.000000	.000000
-3.000000	2.000000	1.000000
6.000000	5.000000	1.000000

THE NORMALIZED MATRIX IS

.948683	.316228	.000000
-.801784	.534522	.267261
.762001	.635001	.127000

SINGULAR VALUES

.14879237062104166966E+01

.88661324403419783380E+00

.16759151454154787521E-14

THE NUMERICAL RANK OF THE MATRIX IS 2

THE GRAM MATRIX IS

1.000000	-.591608	.923702
-.591608	1.000000	-.237595
.923702	-.237595	1.000000

VALUE OF COMPOUND ELEMENT .943548387096765139858689 5

ROWS
2 3
VALUE OF COMPOUND ELEMENT .650000000000002131628207 3

ROWS
1 2
VALUE OF COMPOUND ELEMENT .146774193548387721364179 0

ROWS
1 3

THE ROW INDEPENDENT MATRIX IS

-3.000000	2.000000	1.000000
6.000000	5.000000	1.000000

Example (B.3):

THE GIVEN MATRIX IS

2.00000	-3.00000	1.00000	.00000
2.01000	-3.01000	1.00000	.00000
1.99000	-2.99000	1.00000	.00000
-4.00010	8.00010	-5.00000	1.00000

THE NORMALIZED MATRIX IS

.53452	-.80178	.26726	.00000
.53528	-.80159	.26631	.00000
.53376	-.80197	.26822	.00000
-.38852	.77703	-.48564	.09713

SINGULAR VALUES

.198519805255688908 12E+01

.242874855045482895 83E+00

.704919054269430994 66E-03

.434201286037535027 68E-14

```
        THE NUMERICAL RANK OF THE MATRIX IS    3


                    THE GRAM MATRIX IS

     1.00000       1.00000       1.00000       -.96047
     1.00000       1.00000       1.00000       -.96016
     1.00000       1.00000       1.00000       -.96079
     -.96047       -.96016       -.96079       1.00000


 VALUE OF COMPOUND ELEMENT    .00000007701192281636708 22

 ROWS
  2   3   4
 VALUE OF COMPOUND ELEMENT    .00000001939077682592284 75

 ROWS
  1   3   4
 VALUE OF COMPOUND ELEMENT    .00000001911573363803461 66

 ROWS
  1   2   4
 VALUE OF COMPOUND ELEMENT    .00000000000000000555111 51

 ROWS
  1   2   3


                THE ROW INDEPENDENT MATRIX IS

     2.01000      -3.01000       1.00000        .00000
     1.99000      -2.99000       1.00000        .00000
    -4.00010       8.00010      -5.00000       1.00000
```

Example (B.4):

```
                    THE GIVEN MATRIX IS

     5.00000       2.00000       5.00000       2.00000        .00000
     1.00000       1.00000       1.00000       1.00000        .00000
     4.00000       1.00000       4.00000       1.00000        .00000
     5.00000       3.00000       5.00000       3.00000        .00000
     6.00000       6.00000      14.00000       6.00000       8.00000
    15.00000        .00000      18.00000        .00000       3.00000
     3.00000       3.00000       7.00000       3.00000       4.00000
      .00000       7.00000        .00000      15.00000        .00000
```

THE NORMALIZED MATRIX IS

.65653	.26261	.65653	.26261	.00000
.50000	.50000	.50000	.50000	.00000
.68599	.17150	.68599	.17150	.00000
.60634	.36380	.60634	.36380	.00000
.31277	.31277	.72980	.31277	.41703
.63500	.00000	.76200	.00000	.12700
.31277	.31277	.72980	.31277	.41703
.00000	.42289	.00000	.90618	.00000

SINGULAR VALUES

.2539518094520772706 3E+01

.1070016621034184822 7E+01

.6092216088780482152 7E+00

.1864438510321999586 9E+00

.1619862199537686346 5E-14

THE NUMERICAL RANK OF THE MATRIX IS 4

THE GRAM MATRIX IS

1.00000	.91915	.99083	.98724	.84876	.91718	.84876	.34903
.91915	1.00000	.85749	.97014	.83406	.69850	.83406	.66453
.99083	.85749	1.00000	.95667	.82248	.95834	.82248	.22793
.98724	.97014	.95667	1.00000	.85973	.84706	.85973	.48352
.84876	.83406	.82248	.85973	1.00000	.80768	1.00000	.41569
.91718	.69850	.95834	.84706	.80768	1.00000	.80768	.00000
.84876	.83406	.82248	.85973	1.00000	.80768	1.00000	.41569
.34903	.66453	.22793	.48352	.41569	.00000	.41569	1.00000

VALUE OF COMPOUND ELEMENT .012284886518359350304763 2

ROWS
2 5 6 8
VALUE OF COMPOUND ELEMENT .012284886518359128260158 3

ROWS
2 6 7 8
VALUE OF COMPOUND ELEMENT .008064666679112153513386 8

ROWS
2 3 5 8
VALUE OF COMPOUND ELEMENT .008064666679112153513386 8

ROWS
2 3 7 8
VALUE OF COMPOUND ELEMENT .007869553777558540281057 73

ROWS
4 5 6 8

VALUE OF COMPOUND ELEMENT .007869553775585125254821 2

ROWS
4 6 7 8
VALUE OF COMPOUND ELEMENT .005085519401825527863181 3

ROWS
1 6 7 8
VALUE OF COMPOUND ELEMENT .005085519401825416840878 8

ROWS
1 5 6 8
VALUE OF COMPOUND ELEMENT .004727563225685849079837 9

ROWS
1 2 5 8
VALUE OF COMPOUND ELEMENT .004727563225685849079837 9

ROWS
1 2 7 8
VALUE OF COMPOUND ELEMENT .003038702812336077641042 2

ROWS
3 6 7 8
VALUE OF COMPOUND ELEMENT .003038702812336022129891 0

ROWS
3 5 6 8
VALUE OF COMPOUND ELEMENT .002582801746905880047044 9

ROWS
3 4 5 8
VALUE OF COMPOUND ELEMENT .002582801746905880047044 9

ROWS
3 4 7 8
VALUE OF COMPOUND ELEMENT .001792148150913046300569 7

ROWS
2 4 5 8
VALUE OF COMPOUND ELEMENT .001792148150913046300569 7

ROWS
2 4 7 8
VALUE OF COMPOUND ELEMENT .000772477651255692315812 6

ROWS
1 4 5 8
VALUE OF COMPOUND ELEMENT .000772477651255692315812 6

ROWS
1 4 7 8
VALUE OF COMPOUND ELEMENT .000747932796853312265028 5

ROWS
2 3 6 8

```
VALUE OF COMPOUND ELEMENT    .000556183908904002433093 4

ROWS
 1   3   5   8
VALUE OF COMPOUND ELEMENT    .000556183908904002433093 4

ROWS
 1   3   7   8
VALUE OF COMPOUND ELEMENT    .000438443363672189750923 6

ROWS
 1   2   6   8
VALUE OF COMPOUND ELEMENT    .000239534032979289886189 4

ROWS
 3   4   6   8
VALUE OF COMPOUND ELEMENT    .000166207288189261330468 8

ROWS
 2   4   6   8
VALUE OF COMPOUND ELEMENT    .000071641072495529003333 3

ROWS
 1   4   6   8
VALUE OF COMPOUND ELEMENT    .000051581572197125381135 1

ROWS
 1   3   6   8
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   4   5   7
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   3   4   6
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   4   6   7
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   3   5   7
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   3   4   7
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   5   6   7
VALUE OF COMPOUND ELEMENT    .000000000000000005551115 1

ROWS
 1   2   4   7
```

```
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   5   7   8
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   4   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   4   5
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   4   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   4   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   4   8
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   5   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   5   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   3   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   3   6   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   3   4   8
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   3   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   4   5   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   4   5   7
```

```
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   5   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   4   6   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   4   5   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   3   4   5
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   5   6   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   3   4
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 2   5   7   8
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   3   5
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 3   4   5   6
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 3   4   5   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   2   6   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 3   4   6   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   3   6   7
VALUE OF COMPOUND ELEMENT    .00000000000000000055511151

ROWS
 1   3   5   6
```

VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
3 5 6 7
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
1 2 4 8
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
3 5 7 8
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
1 2 5 7
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
4 5 6 7
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
1 2 3 8
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
4 5 7 8
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
1 2 4 5
VALUE OF COMPOUND ELEMENT .000000000000000000055511151

ROWS
5 6 7 8

THE ROW INDEPENDENT MATRIX IS

1.00000	1.00000	1.00000	1.00000	.00000
6.00000	6.00000	14.00000	6.00000	8.00000
15.00000	.00000	18.00000	.00000	3.00000
.00000	7.00000	.00000	15.00000	.00000

APPENDIX C

In the present Appendix, we present the extension of some algorithms developed in Chapter 6, to unique factorization domains.

C.1 EXTENSION OF EUCLID'S ALGORITHM TO UNIQUE FACTORIZATION DOMAINS [Knu., 1]

If we restrict consideration to polynomials over a field, we are not dealing directly with many important cases, such as polynomials over integers, or polynomials in several variables. Working with polynomials over integers is of principal importance, because it is often preferable to work with integer coefficients instead of arbitrary rational coefficients.

Let us therefore now consider the more general situation that the algebraic system S of coefficients is a unique factorization domain, not necessarily a field. This means that S is commutative ring with identity, and that

(i) $u \cdot v \neq 0$, whenever u and v are nonzero elements of S;

(ii) every nonzero element u of S is either a "unit" or has a "unique" representation of the form

$$u = p_1 \ldots p_t, \quad t \geq 1 \tag{C.1}$$

where $p_1,\ldots,p_t$ are "primes".

Here a "unit" u is an element such that $u \cdot v = 1$ for some $v \in S$: and a "prime" p is a non unit element such that the equation $p = q \cdot r$ can be true only if either q or r is unit. The representation (C.1) is to be unique in the sense that if $p_1 \ldots p_t = q_1 \ldots p_s$, where all the p's and q's are primes, then s=t and there is a permutation $(\pi_1,\ldots,\pi_t)$ such that

$$p_1 = a_1 q_{\pi_1}, \ldots, p_t = a_t q_{\pi_t} \text{ for some units } a_1,\ldots,a_t.$$

In other words, factorization into primes is unique, except for unit multiples and except for the order of the factors.

Any field is a unique factorization domain, in which each nonzero element is a unit and there are no primes. The integers form a unique factorization domain in which the units are +1 and -1, and the primes are ±2, ±3, ±5, +7,...

It can be proved and it is an important fact, that the polynomials over a unique factorization domain [Hod. & Ped., 1] form a unique factorization domain. A polynomial which is "prime" in this domain is usually called an irreducible polynomial.

A set of elements of a unique factorization domain is said to be relatively prime if no prime (of the unique factorization domain) divides all

of them. A polynomial over a unique factorization domain is called primitive if its coefficients are relatively prime.

Lemma (C.1) (Gauss's Lemma) [Knu., 1]: The product of primitive polynomials over a unique factorization domain is primitive. ■

Lemma (C.2) [Knu., 1]: Any nonzero polynomial u(s) over a unique factorization domain S can be factored in the form $u(s)=c\cdot v(s)$, where c is in S and v(s) is primitive. Furthermore, this representation is unique, in the sense that if $u(s)=c_1\cdot v_1(s)=c_2\cdot v_2(s)$, then $c_1=a\cdot c_2$ and $v_2(s)=a\cdot v_1(s)$ where a is a unit of S.

■

Therefore, we may write any nonzero polynomial u(s) as

$$u(s) = cont(u)\cdot pp(u(s)) \tag{C.2}$$

where con(u), the "content" of u, is an element of S and pp(u(s)), the "primitive part" of u(s), is a primitive polynomial over S. When u(s)=0, it is convenient to define cont(u)=pp(u(s))=0. The combination of Lemmas (6.1) and (6.2) gives us the relations

$$\begin{aligned} con(u\cdot v) &= a\cdot cont(u)\cdot cont(v) \\ pp(u(s)\cdot v(s) &= b\cdot pp(u(s))\cdot pp(v(s)) \end{aligned} \tag{C.3}$$

where a and b are units, depending on u and v, with ab=1 when we are working with polynomials over the integers, the only units are +1 and -1, and it is conventional to define pp(u(s)) so that its leading coefficient is positive; then (C.3) is true with a=b=1. When working with polynomials over a field we may take cont(u)=the leading coefficient of the polynomial, so that pp(u(s)) is monic; in this case again (C.3) holds with a=b=1, for all u(s) and v(s).

Example (C.1): If we are dealing with polynomials over the integers, let $u(s)=-26s^2+39$, $v(s)=21s+14$. Then

$$\begin{aligned} cont\ (u) &= -13, \quad & pp(u(s)) &= 2s^2-3, \\ cont\ (v) &= 7, \quad & pp(v(s)) &= 3s+2, \\ cont(u\cdot v) &= -91, \quad & pp(u(s)\cdot v(s)) &= 6s^3+4s^2-9s-6. \end{aligned}$$

■

From equations (C.3) we can deduce the important relations

$$cont((u,v)) = a\cdot(cont(u),cont(v)),$$

$$pp((u(s),v(s))) = b\cdot(pp(u(s)),\ pp(v(s))) \tag{C.4}$$

where a and b are units. Equations (C.4) reduce the problem of finding greatest common divisors of arbitrary polynomials to the problem of finding greatest common divisors of primitive polynomials. Clearly cont(u) is a greatest common divisor of the coefficients of u, and pp(u(x))=u(x)/cont(u).

Algorithm **DIV** (section 6.2 of Chapter 6) for division of polynomials over a field, can be generalized to a pseudodivision of polynomials over any algebraic system which is a commutative ring with identity. We can observe that Algorithm **DIV** requires explicit division only by l(b), the leading coefficient of b(s), and that the main process of this algorithm is carried out exactly m-n+1 times; thus if a(s) and b(s) start with integer coefficients, and if we are working over the rational numbers, then the only denominators which appear in the coefficients of q(s) and r(s) are divisors of $l(b)^{m-n+1}$. This suggests that we can always find polynomials q(s) and r(s) such that

$$l(b)^{m-n+1}a(s) = q(s)b(s)+r(s),\ \deg\{r(s)\}<n \tag{C.5}$$

where m=deg{a(s)}, n=deg{b(s)}, and $m\geq n$, for any polynomials a(s) and $b(s)\neq 0$.

Algorithm PDIV (Pseudodivision of polynomials)

Given polynomials

$$a(s)=a_m s^m+\ldots+a_1 s+a_0,\ b(s)=b_n s^n+\ldots+b_1 s+b_0,\quad b_n\neq 0,\quad m\geq n\geq 0,$$

this algorithm finds polynomials $q(s)=q_{m-n}s^{m-n}+\ldots+q_0$ and $r(s)=r_{n-1}s^{n-1}+\ldots+r_0$ satisfying (C.5)

```
for k = m-n, m-n-1,...,0
    a_k := a_{n+k}·b_n^k
    for j = n+k-1, n+k-2,...,0
        a_j := b_n·a_j - a_{n+k}·b_{j-k}
```

Alg: C.1

When the algorithm is terminated $a_{n-1}=r_{n-1},\ldots,a_0=r_0$. In the second loop of the above algorithm when j becomes < k we treat b_{-1}, b_{-2},..., as zero. It

is worth noting that this algorithm does not use any divisions. The coefficients of q(s) and r(s) are themselves certain polynomial functions of the coefficients of a(s) and b(s). If $b_n=1$, the algorithm is identical to Algorithm **DIV**. If $b_n\neq 0$ and if a(s) and b(s) are polynomials over a unique factorization domain, we can prove as before that the polynomials q(s) and r(s) of (C.5) are unique; therefore another way to do the pseudodivision over a unique factorization domain, which may sometimes be preferable, is to multiply a(s) by b_n^{m-n+1} and apply Algorithm **DIV**.

Example (C.2): Let $a(s)=3s^2+3s+4$, $b(s)=2s+3$ be two given polynomials.

(i) If we take them over a field e.g. over R[s], then after applying Algorithm **DIV** we get:

$$q(s) = \frac{3}{2}s - \frac{3}{4},\quad r(s) = \frac{25}{4} \text{ and therefore}$$

$$3s^2+3s+4=\left(\frac{3}{2}s-\frac{3}{4}\right)(2s+3)+\frac{25}{4}$$

(ii) If we take them over the integers which forms a unique factorization domain, then after the application of Algorithm **PDIV** we get:

$q(s)=6s-3$, $r(s)=25$ and the next relation holds

$$2^2(3s^2+3s+4) = (6s-3)(2s+3)+25$$

■

Algorithm **PDIV** can be extended to a "generalized Euclidean algorithm" for primitive polynomials over a unique factorization domain, in the following way: Let a(s) and b(s) be primitive polynomials with $\deg\{a(s)\}\geq\deg\{b(s)\}$, and determine r(s) satisfying (C.5) by means of Algorithm **PDIV**. Now (a(s),b(s))=(b(s),r(s)): For any common divisor of a(s) and b(s) divides b(s) and r(s); conversely, any common divisor of b(s) and r(s) divides $l(b)^{m-n+1}$ a(s), and it must be primitive (since b(s) is primitive) so it divides a(s). If r(s)=0, we therefore have (a(s),b(s))=b(s); if $r(s)\neq 0$, we have (a(s),b(s))=(b(s), pp(r(s))) since b(s) is primitive, so the process can be iterated.

Algorithm GEUCLID (Generalized Euclidean algorithm or The primitive P.R.S. algorithm)

Given nonzero polynomials $a(s)=a_ms^m+\ldots+a_1s+a_0$ and $b(s)=b_ns^n+\ldots+b_1s+b_0$ over a unique factorization domain S, this algorithm calculates a greatest

common divisor d(s) of a(s) and b(s). We assume that an auxiliary algorithm exists for calculating greatest common divisors of elements of S.

STEP 1: cont(a) := $(a_m, a_{m-1}, \ldots, a_1, a_0)$

cont(b) := $(b_n, b_{n-1}, \ldots, b_1, b_0)$

d := (cont(a), cont(b))

pp(a(s)) := a(s)/cont(a)

pp(b(s)) := b(s)/cont(b)

a(s) := pp(a(s))

b(s) := pp(b(s))

STEP 2: Calculate the remainder r(s) using Algorithm **PDIV**

(It is unnecessary to calculate the quotient polynomial q(s)).

if r(s) = 0 **then**

d(s) := d·b(s)

quit

else if deg{r(s)}=0 **then**

Replace b(s) by the constant polynomial "1"

d(s) := d·b(s)

quit

STEP 3: a(s) := b(s)

b(s) := pp(r(s))

Repeat **STEP 2**

Alg: C.2

Example (C.3): Let us calculate the greatest common divisor of

$$a(s) = s^8+s^6-3s^4-3s^3+8s^2+2s-5,$$
$$b(s) = 3s^6+5s^4-4s^2-9s+21 \qquad (C.6)$$

(i) over the integers using Algorithm **GEUCLID**

(ii) over the rational numbers using Algorithm **EUCLID**.

(i) These polynomials are primitive, so **STEP 1** sets d:=1.

In **STEP 2** we do the pseudodivision:

$1(b)^{m-n+1}a(s)=q(s)b(s)+r(s)$ and we find

$3^3a(s)=(9s^2-6)b(s)+(-15s^4+3s^2-9)$

In the sequel, in **STEP 3** a(s) is replaced by b(s) and b(s) by $pp(r(s))=5s^4-s^2+3$. The subsequent calculations may be summarized as follows:

a(s)	b(s)	r(s)
$s^8+s^6-3s^4-3s^3+8s^2+2s-5$	$3s^6+5s^4-4s^2-9s+21$	$-15s^4+3s^2-9$
$3s^6+5s^4-4s^2-9s+21$	$5s^4-s^2+3$	$-585s^2-1125s+2205$
$5s^4-s^2+3$	$13s^2+25s-49$	$-233150s+307500$
$13s^2+25s-49$	$4663s-6150$	143193869

Thus, d(s)=1 and evidently the polynomials are coprime.

(ii) The following sequence occurs:

a(s)	b(s)	r(s)
$s^8+s^6-3s^4-3s^3+8s^2+2s-5$	$3s^6+5s^4-4s^2-9s+21$	$-5/9s^4+1/9s^2-1/3$
$3s^6+5s^4-4s^2-9s+21$	$-5/9s^4+1/9s^2-1/3$	$-117/25s^2-9s+441/25$
$-5/9s^4+1/9s^2-1/3$	$-117/25s^2-9s+441/25$	$\frac{2331505}{6591}s-\frac{102500}{2197}$
$-\frac{117}{25}s^2-9s+\frac{441}{25}$	$\frac{2331505}{6591}s-\frac{102500}{2197}$	$\frac{1288744821}{543589225}$

To improve that algorithm, we can reduce a(s) and b(s) to monic polynomials at each step, since this removes "unit" factors which make the coefficients more complicated than necessary; this is actually Algorithm **GEUCLID** over rationals:

a(s)	b(s)	r(s)
$s^8+s^6-3s^4-3s^3+8s^2+2s-5$	$s^6+\frac{5}{3}s^4-\frac{4}{3}s^2-3s+7$	$s^4-\frac{1}{5}s^2+\frac{3}{5}$
$s^6+\frac{5}{3}s^4-\frac{4}{3}s^2-3s+7$	$s^4-\frac{1}{5}s^2+\frac{3}{5}$	$s^2+\frac{25}{13}s-\frac{49}{13}$
$s^4-\frac{1}{5}s^2+\frac{3}{5}$	$s^2+\frac{25}{13}s-\frac{49}{13}$	$s-\frac{6150}{4663}$
$s^2+\frac{25}{13}s-\frac{49}{13}$	$s-\frac{6150}{4663}$	1

Comparing (i) and (ii) we conclude the following: In (ii) the sequence of polynomials is essentially the same as in (i), which was obtained by Algorithm **GEUCLID** over the integers; the only difference is that the polynomials have been multiplied by certain rational numbers. Whether we have $5s^4-s^2+3$ or $-\frac{5}{9}s^4+\frac{1}{9}s^2-\frac{1}{3}$ or $s^4-\frac{1}{5}s^2+\frac{3}{5}$, the computations are essentially the same. But either algorithm using rational arithmetic will run noticeably slower than the all-integer Algorithm **GEUCLID**, since rational arithmetic requires many more evaluations of gcd's of integers within each step. Therefore it is definitely preferable to use the all-integer algorithm instead of rational arithmetic, when the g.c.d. of polynomials with integer or rational coefficients is desired.

■

Collins's Algorithm [Col., 1] An ingenious algorithm which is generally superior to Algorithm **GEUCLID**, and which gives us further information about Algorithm **GEUCLID**'s behavior, has been given by Collins [Col., 1]. His algorithm avoids the calculation of primitive part in **STEP 3**, dividing instead by an element of S which is known to be a factor of r(s):

Algorithm COLLINS (The reduced P.R.S. algorithm). This algorithm has the same input and output assumptions as Algorithm **GEUCLID**, and has the advantage that less calculations of greatest common divisors of coefficients are needed.

STEP 1: cont(a) := $(a_m, a_{m-1}, \ldots, a_1, a_0)$

cont(b) := $(b_n, b_{n-1}, \ldots, b_1, b_0)$

d := (cont(a), cont(b))

pp(a(s)) := a(s)/cont(a)

pp(b(s)) := b(s)/cont(b)

a(s) := pp(a(s))

b(s) := pp(b(s))

c := 1

STEP 2: f := $l(b)^{m-n+1}$

Calculate the remainder r(s) using Algorithm **PDIV**

(It is unnecessary to calculate the quotient polynomial q(s))

if r(s) = 0 **then**

d(s) := d·pp(b(s))

quit

else if deg{r(s)} = 0 **then**

Replace b(s) by the constant polynomial "1"

d(s) := d·pp(b(s))

quit

STEP 3: a(s) := b(s)

b(s) := r(s)/c

c := f

Repeat **STEP 2**

Alg: C.3

Example (C.4): If we apply this algorithm to the polynomials (C.6) considered earlier, the following sequence of results is obtained:

a(s)	b(s)	c
$s^8+s^6-3s^4-3s^38s^2+2s-s$	$3s^6+5s^4-4s^2-9s+21$	1
$3s^6+5s+4s^2-9s+21$	$-15s^4+3s^2-9$	27
$-15s^4+3s^2-9$	$585s^2+1125s-2205$	$(-15)^3$
$585s^2+1125s-2205$	$-18885150s+24907500$	$(585)^3$

At the end of the algorithm, r(s)/c=527933700. The sequence of polynomials concists of integral multiples of the polynomials in the sequence produced by Algorithm **GEUCLID**. The example shows that in spite of the fact that the polynomials are not reduced to primitive form, the coefficients are kept to a reasonable size because the reduction factor c is large.

■

C.2 EXTENSION OF ROUTH'S ALGORITHM TO UNIQUE FACTORIZATION DOMA

To extend Routh's array $[r_{ij}]$ developed in section 6.3 of Chapter 6 to unique factorization domain (u.f.d.) we must avoid the divisions by $r_{i-1,1}$ used in formula (6.9). This can be achieved by setting:

$$s_{ij}=r_{1j},\quad s_{2j}=r_{2j},\quad s_{ij}=k_i\cdot r_{ij},\quad i\geq 3$$

where (C.7)

$$k_3=r_{21},\ k_4=r_{21}\cdot r_{31},\quad k_i=k_{i-1}\cdot k_{i-2}\cdot r_{i-1,1},\quad i\geq 5$$

which is equivalent to defining

$$s_{ij} = -\begin{vmatrix} s_{i-2,1} & s_{i-2,j+1} \\ s_{i-1,1} & s_{i-1,j+1} \end{vmatrix},\quad i=3,4,\ldots \qquad (C.8)$$

and in equation (C.4) the r_{ij} are replaced by s_{ij} to give polynomials $g_i(s)$. Starting with polynomials with integer coefficients and applying successively formula (C.8), the generating elements will remain in the integers so the array $[s_{ij}]$ can be termed fraction free.

Example (C.5): We construct the array defined by (C.7), (C.8) for the polynomials

$$a(s) = s^4+5s^3+2s^2+4s+3,\quad b(s) = 4s^3+3s^2+s+2$$

Table (C.1)

$[s_{1j}]$	1	5	2	4	3
$[s_{2j}]$	4	3	1	2	
$[s_{3j}]$	17	7	14	12	
$[s_{4j}]$	23	-39	14		
$[s_{5j}]$	824	560	276		
$[s_{6j}]$	-45016	-17884			
$[s_{7j}]$	-104772544	-12424416			
$[s_{8j}]$	-372006533760				

However the elements of the above array are growing very fast and one might encounter storage problems. This could be dealt with by dividing each row through its g.c.d. However, finding the g.c.d. is again time consuming.

The following technique proposed in [Jel., 1] avoids the phenomenon of rapid coefficient growth without computing a g.c.d.

Let $P[a,b]$ be the ring of polynomials in the variables a_i, b_j, $i=1,2,\ldots,m$, $j=1,2,\ldots,n$ and $R[a,b]$ the rational functions in the same variables. The rows r_{1j}, r_{2j} are polynomials in $P[a,b]$ of degree 1.

Definition (C.1) [Jel., 1]: An array $s[k]=\{s_{ij}\}$ is called scaled free Routh array if there exist scaling factors $k_i \epsilon R[a,b]$ such that

$$s_{ij} = k_i \cdot r_{ij} \; \epsilon P[a,b] \tag{C.9}$$

■

Definition (C.2): The normalized degree v_i of the i-th row of a scaled fraction free Routh array is the largest degree of $s_{i1}, s_{i2}, \ldots$ considered as polynomials in $P[a,b]$.

■

For the array s_{ij} one gets from (C.8) easily the recursion relation

$$v_i = v_{i-1}+v_{i-2}, \quad i=3,4,\ldots \tag{C.10}$$

and from (C.7) one obtains $v_1=v_2=1$.
Hence the normalized degrees are the Fibonacci numbers which are known to grow exponentially. This implies that the computational complexity will grow exponentially. Hence it is natural to ask for the scaled fraction free Routh array with smallest normalized degrees.

Theorem (C.1) [Jel., 1]: The optimal fraction free Routh algorithm which achieves $v_i=i$ is given by the recurrence relation

$$n_{ij} = -\frac{1}{d_i} \begin{vmatrix} n_{i-2,1} & n_{i-2,j+1} \\ n_{i-1,1} & n_{i-1,j+1} \end{vmatrix}, \quad j=1,2,3,\ldots \tag{C.11}$$

where

$$d_i = \begin{cases} 1 & \text{for } i=3,4 \\ n_{i-3,1} & \text{for } i=5,6,\ldots \end{cases} \tag{C.12}$$

The recursion is started with

$$n_{1j} = r_{1j} = a_j$$

$$, \; j = 1,2,3,... \qquad (C.13)$$

$$n_{2j} = r_{2j} = b_j$$

■

As long as $d_i \neq 0$ one has no breakdown in the algorithm. The case with $d_i = 0$ for some i has to be handled in similar ways as in the regular Routh algorithm.

Remark (C.1): Algorithm **ROUTH** can be easily modified so as to be extended to u.f.d. The only modifications needed are slight changes in the formulas (C.2) defining the rows of the Routh array accordingly to (C.8) if only fraction free Routh array is required or in the formulas (C.11), (C.12) if optimal fraction free Routh array is desired.

■

Example (C.6): We construct the array defined by (C.11) and (C.12) for the polynomials in Example (C.5).

Table (C.2)

$[n_{1j}]$	1	5	2	4	3
$[n_{2j}]$	4	3	1	2	
$[n_{3j}]$	17	7	14	12	
$[n_{4j}]$	23	-39	-14		
$[n_{5j}]$	206	140	69		
$[n_{6j}]$	-662	-263			
$[n_{7j}]$	-1674	-1986			
$[n_{8j}]$	-4245				

We can see from the above values that optimal fraction-free Routh array has produced a remarkable reduction in the magnitude of the elements compared with the array in the Example (C.5).

■

APPENDIX D

In the present Appendix, we present a description of algorithm **GAUSS**, required for the development of the main algorithm of Chapter 7 that achieves the computation of the g.c.d. of several polynomials. The computational complexity of algorithm **GAUSS** and its error analysis as well, are also described.

Algorithm GAUSS [Gol. & Loan, 1]

Let $A = [\underline{r}_1^t, \underline{r}_2^t, \ldots, \underline{r}_m^t] \in R^{mxn}$ a given matrix. The following algorithm transforms A into upper trapezoidal form if $m<n$ or into upper triangular form if $m \geq n$ using the strategy of partial pivoting.

```
for  k = 1,2,...,min{m-1,n}
     Determine p ε{k,k+1,...,m } so
     |a_pk| = max |a_ik|
              1≤i≤m
     Swap r_k^t and r_p^t
     for  j = k+1,...,n
          w_j := a_kj
          for  i = k+1,...,m
               v := a_ik/a_kk
               a_ik := 0
               for  j = k+1,...,n
                    a_ij := a_ij-v·w_j
```

Alg: 7.6

Computational complexity of **GAUSS** [Gol. & Loan, 1], [Stew., 1]

The computational complexity is measured by the number of flops and comparisons required by the algorithm. (We shall use C.B. Moler's concept of a flop. A flop is more or less the amount of work associated with each program statement). The entire algorithm requires:

$$\sum_{k=1}^{s=\min\{m-1,n\}} (m-k)\cdot(n-k) \quad \text{flops}$$

It is possible to evaluate this sum using standard formulas. However, we are only interested in the sum involving the highest power of s in the final result. To obtain this, we may approximate the sum by an iterated integral; namely we replace the sum by integral with the summation limits as limits of integration.

$$\int_1^s (m-k)\cdot(n-k)dk = \int_1^s (mn-(m+n)k+k^2)dk =$$

$$= s^3/3-(m+n)\cdot s^2/2+mns+m+n-mn-1/3 \approx O(s^3/3)$$

Thus, the time required for the execution of the algorithm is proportional to $s^3/3$.

The number of comparisons required are given by:

$$\sum_{k=1}^{s=\min\{m-1,n\}} (m-k)$$

We approximate the sum by the integral:

$$\int_1^s (m-k)dk = -s^2/2+ms-m+1/2 \approx O(s^2/2)$$

Error Analysis of **GAUSS** [Wilk., 1], [For. & Mol., 1]

Because pivoting only involves permutations of the row subscripts, is irrelevant to our error analysis. The decomposition of matrix $A\epsilon R^{mxn}$ consists of computing a sequence of matrices $A^{(1)}= A$, $A^{(2)},\ldots,A^{(s)}$ where $s=\min\{m-1,n\}$ and $A^{(k)}$ is zero below the diagonal in the first k-1 columns. The matrix $A^{(k+1)}$ is obtained from $A^{(k)}$ by subtracting a multiple of the k-th row from each of the rows below it the rest of $A^{(k)}$ is left unchanged. The multipliers are chosen so that if there were no rounding errors, $A^{(k+1)}$ would have zeros below the diagonal in the k-th column. We do not calculate this elements but take them to be zero by definition. More precisely, let $A^{(k)}$ have elements $a_{ij}^{(k)}$. Then let

$$m_{i,k} = fl\left(a_{ik}^{(k)} / a_{kk}^{(k)} \right) , \quad i \geq k+1 \tag{D.1}$$

and

$$a_{ij}^{(k+1)} = \begin{cases} 0 & \text{for } i \geq k+1,\ j=k \\ fl\left(a_{ij}^{(k)} - m_{i,k} \cdot a_{kj}^{(k)} \right) & \text{for } i \geq k+1,\ j \geq k+1 \\ a_{ij}^{(k)} & \text{otherwise} \end{cases} \tag{D.2}$$

These steps are carried out for k= 1,2,...,s.

Finally, let

$$U = A^{(s)} \quad \text{and} \quad L = \begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{32} & 1 & & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ m_{m1} & m_{m2} & m_{m3} & \cdots & 1 \end{bmatrix} \tag{D.3}$$

Clearly L is lower triangular and U is upper triangular or upper trapezoidal.

According to the technique of backward error analysis we will prove that:

(I) $L \cdot U = A + E$, where E is a matrix of small elements which account for the rounding errors.

(II) We must bound the size of E.

Let us examine them separately:

(I) The error in the calculation of the multipliers m_{ik} is expressed by

$$m_{ik} = \left(a_{ik}^{(k)} / a_{kk}^{(k)} \right) (1+\varepsilon) , \quad |\varepsilon| \leq u \tag{D.4}$$

or

$$0 = a_{ik}^{(k)} - m_{ik} a_{kk}^{(k)} + a_{ik}^{(k)} \cdot \varepsilon$$

That is,

$$\varepsilon_{ik}^{(k)} = a_{ik}^{(k)} \cdot \varepsilon , \quad i \geq k+1 , \quad |\varepsilon| \leq u \tag{D.5}$$

is the error made by setting $a_{ik}^{(k+1)}$ equal to zero.

For the second case of (D.2) we have :

$$\overset{(k+1)}{a_{ij}} = fl\ (\overset{(k)}{a_{ij}} - fl(m_{ik}\cdot\overset{(k)}{a_{kj}}) = fl\ (\overset{(k)}{a_{ij}} - m_{ik}\cdot\overset{(k)}{a_{kj}}(1+\varepsilon_1)) =$$

$$= (\overset{(k)}{a_{ij}} - m_{ik}\cdot\overset{(k)}{a_{kj}}(1+\varepsilon_1))\cdot(1+\varepsilon_2)\ ,\ |\varepsilon_1|,|\varepsilon_2| \leq u \qquad (D.6)$$

The error in the calculation of the $\overset{(k+1)}{a_{ij}}$ is:

$$\overset{(k)}{\varepsilon_{ij}} = a_{ij}^{(k+1)} - (\ a_{ij}^{(k)} - m_{ik}\cdot a_{kj}^{(k)}\ ) \qquad (D.7)$$

(that is $\overset{(k)}{\varepsilon_{ij}}$ is the difference between the accepted value $\overset{(k+1)}{a_{ij}}$ and the exact value which would be obtained using the computed $\overset{(k)}{a_{ij}}$,m_{ik} , $\overset{(k)}{a_{kj}}$.)

Relation (D.7) using relation (D.6) becomes:

$$\overset{(k)}{\varepsilon_{ij}} = \ (\overset{(k+1)}{a_{ij}}\cdot\varepsilon_2)/(1+\varepsilon_2) - m_{ik}\cdot\overset{(k)}{a_{kj}}\cdot\varepsilon_1\ ,\ |\varepsilon_1|,|\varepsilon_2| \leq u$$

$$\text{for } i \geq k+1,\ j \geq k+1 \ \text{ and } k= 1,2,\ldots,s$$

Thus, the previous $\overset{(k)}{\varepsilon_{ij}}$ is the error in calculating the $\overset{(k+1)}{a_{ij}}$ for the new part of $A^{(k+1)}$. The rest of $A^{(k+1)}$ is taken directly from $A^{(k)}$, so there is no error. In summary,

$$\overset{(k)}{\varepsilon_{ij}} = \begin{cases} \overset{(k)}{a_{ik}}\cdot\varepsilon & \text{for } i\geq k+1,\ j=k \\ (a_{ij}\cdot\varepsilon_2)/(1+\varepsilon_2) - m_{ik}\cdot a_{kj}\cdot\varepsilon_1 & \text{for } i\geq k+1,\ j\geq k+1 \\ 0 & \text{otherwise} \end{cases} \qquad (D.8)$$

Thus if we let $E^{(k)}$ be the matrix with elements $\overset{(k)}{\varepsilon_{ij}}$ and let

$$L^{(k)} = \begin{bmatrix} 0 & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \ldots & 0 \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ 0 & & m_{k+1k} & & 0 \\ \cdot & & m_{k+2k} & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ 0 & & m_{mk} & & 0 \end{bmatrix}$$

then the equation

$$A^{(k+1)} = A^{(k)} - L^{(k)}A^{(k)} + E^{(k)}$$

completely describes one step of the decomposition, including the rounding error. Adding these equations for k= 1,2,...,s we have

$$L^{(1)}\cdot A^{(1)} + L^{(2)}\cdot A^{(2)} + \ldots + L^{(s-1)}\cdot A^{(s-1)} + A^{(s)} = A^{(1)} + E^{(1)} + E^{(2)} + \ldots + E^{(s-1)}$$

The matrix $L^{(k)}\cdot A^{(k)}$ depends upon only the k-th row of $A^{(k)}$, and this row is equal to the k-th row of $A^{(s)}$. Thus we have:

$$(L^{(1)} + L^{(2)} + \ldots + L^{(s-1)} + I)\cdot A^{(s)} = A^{(1)} + E^{(1)} + \ldots + E^{(s-1)} \qquad (D.9)$$

That is, $L\cdot U = A^{(1)} + E$, $E = E^{(1)} + E^{(2)} + \ldots + E^{(s-1)}$ (D.10)
where L,U are defined by (D.3) and E is the sum of the errors at the individual steps.

(II) It is clear that if we are to obtain a satisfactory bound for the $\varepsilon_{ij}^{(k)}$ we will need good bounds for the m_{ik} and $a_{ij}^{(k)}$ which appear in the $\varepsilon_{ij}^{(k)}$. Pivoting is used just to keep these bounds small and to ensure that $|m_{ik}| \leq 1$ for all i,k. This is done by partial pivoting. We shall denote the maximum element in any $|A^{(r)}|$ by g. There is no real loss of generality in assuuming $|a_{ij}^{(i)}| \leq 1$ since this may be achieved by scaling (without rounding errors). The equation $L\cdot U = A^{(1)} + E$ is true, provided $A^{(1)}$ is used to denote the original matrix with its rows suitable permuted.

Let us give bounds for the error expressed in (D.8)

$$|\varepsilon_{ij}^{(k)}| = |\ (a_{ij}^{(k)}\cdot\varepsilon_2)/(1+\varepsilon_2) - m_{ik}\cdot a_{kj}^{(k)}\cdot\varepsilon_1\ | \leq g\cdot u/(1+u) + g\cdot u \leq (2.01)g\cdot u$$

$$|\varepsilon_{ij}^{(k)}| = |a_{ik}^{(k)}\cdot\varepsilon| \leq g\cdot u \leq (2.01)\cdot g\cdot u$$

Thus

$$\varepsilon_{ij} = \begin{cases} (2.01)\cdot g\cdot u\ , & i\geq k+1,\ j=k \\ (2.01)\cdot g\cdot u\ , & i\geq k+1,\ j\geq k+1 \\ 0 \qquad\quad \text{otherwise} \end{cases} \qquad (D.11)$$

Combining relations (D.10), (D.11) we have

$$|E| \leq (2.01)\cdot g\cdot u \begin{bmatrix} 0 & 0 & 0 & . & . & . & . & . & 0 & 0 \\ 1 & 1 & 1 & . & . & . & . & . & 1 & 1 \\ 1 & 2 & 2 & . & . & . & . & . & 2 & 2 \\ 1 & 2 & 3 & . & . & . & . & . & 3 & 3 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ 1 & 2 & 3 & . & . & . & . & . & (s-1) & (s-1) \end{bmatrix} \qquad \text{(D.12)}$$

In practice, using pivoting, g is almost invariably of order unity and is quite unimportant.

From relation (D.12) taking norms we have:

$$\| E \|_\infty \leq s^2\cdot(2.01)\cdot g\cdot u$$

Since we are assuming that all $a_{ij}^{(k)}$ are less than unity, then

$$\| E \|_\infty \leq s^2\cdot(2.01)\cdot u$$

We conclude, giving the following theorem:

Theorem (D.1): The matrices L,U computed by Gaussian elimination with partial pivoting, using floating-point arithmetic with unit round-off u satisfy

with
$$L\cdot U = A+E$$
$$\| E \|_\infty \leq (2.01)\cdot s^2\cdot u \qquad \text{(D.13)}$$

In other words L and U form the exact decomposition of some slightly perturbed matrix.

■

REFERENCES

[Akr., 1] AKRITAS, A., " A new method for computing polynomial greatest common divisors", TR-86-9, Univer. of Kansas, Dept. of Comp. Scien., Lawrence, Ks 66045, 1986.

[Atk., 1] ATKINSON, K., "An Introduction to Numerical Analysis", John Wiley & Sons, 1978.

[Ayr., 1] AYRES, J., "Matrices", Schaum Outline Series, McGraw Hill, 1974.

[Bar., 1] BARNETT, S., " Greatest Common Divisor of two polynomials", Lin. Alg. and its Appl., Vol. 3, pp. 7-9., 1970.

[Bar., 2] BARNETT, S., "Greatest Common Divisor of several polynomials", Proc. Cambr. Phil. Soc., Vol. 70, pp. 263-268, 1971.

[Bar., 3] BARNETT, S., "A new look at classical algorithms for polynomial Resultant and G.C.D. calculation", SIAM Rev., Vol. 16, pp. 193-206, 1974.

[Bar., 4] BARNETT., S., " Greatest Common Divisors from Generalized Sylvester Resultant Matrices", Lin. and Mult. Alg.,, Vol. 8, pp. 271-279.

[Bar., 5] BARNETT, S., "Generalized polynomials and Linear Systems Theory", 3rd IMA Conf. on Control Theory, Edit. I. E. Marshall etc., Academic Press, pp. 3-30, 1981.

[Bau., 1] BAUER., F., "Optimally scaled matrices", Num. Math., Vol., 5, pp. 73-87, 1963.

[Blan., 1] BLANKISHIP, W., "A new version of Euclid Algorithm", Amer. Math. Mon., Vol. 70, pp. 742-744, 1963.

[Boch., 1] BOCHER, M., "Introduction to Higher Algebra", Macmillan Company, Dover, 1964.

[Bro. & Byr., 1] BROCKETT, R.W. and BYRNES, C.I., "Multivariable Nyquist criterion, Root loci and Pole placement. A geometric viewpoint" IEEE Trans. Aut. Control, Vol. AC-26, pp. 271-283, 1981.

[Cal. & Des., 1] CALLIER, F.M. and DESOER, C.A., "Multivariable Feedback Systems", Springer-Verlang, New York, 1982.

[Carl., 1] CARLSON, D., "What are Shur Complements Anyway?", Lin. Alg. and its Appl., Vol. 74, pp. 257-275.

[Char., 1] CHARALAMBIDES, CH., "Combinatorial Analysis", Athens, 1984.

[Chen, 1] CHEN, C.T., "Linear Systems Theory and Design", Holt-Saunders International Editions, 1984.

[Col., 1] COLLINS, G., "Subresultants and reduced PRS", J. ACM, Vol. 14, pp. 128-142, 1967.

[Cur. & Glo., 1] CURTAIN, R.F. and GLOVER, K., "Robust stabilization of infinite-dimensional systems by finite-dimensional controllers", Systems and Control Letters, Vol. 7, pp. 41-48, 1986.

[Des. & Var., 1] DESOER, C.A. and VARAIYA, P., "The minimal realization of a nonanticipative impulse response matrix", SIAM J. Appl. Math., Vol. 15, pp. 754-776, 1967.

[Ever. & Rys., 1] EVERETT, C. and RYSER, H., "The Gram matrix and Hadamard Theorem", Amer. Math. Month., Vol. 53, pp. 21-23, 1946.

[For. & Mol., 1] FORSYTHE, G. and MOLER, E., "Computer Solution of Linear Algebraic Systems", Prentice Hall Inc., 1967.

[For. & Str., 1] FORSYTHE, G. and STRAUS, E., "On best conditioned matrices", Proc. Amer. Math. Soc., Vol. 6, pp. 340-345, 1955.

[Forn., 1] FORNEY, D.G., "Minimal bases of rational vector spaces with application to multivariable linear systems", SIAM J. Control, Vol. 13, pp. 493-520, 1975.

[Fost., 1] FOSTER, L., "Rank and Null space calculations using matrix decomposition without column interchanges", Lin. Alg. and its Appl., Vol. 74, pp. 47-71, 1986.

[Fra., 1] FRANCIS, B.A., "A course in H∞ Contol Theory", Lecture Notes in Control and Inform. Scien., Springer-Verlag, Vol. 88, 1987.

[Fry., 1] FRYER, W., "Applications of Routh's Algorithm to Network Theory Problems", IRE Tran. Cir. Theory, Vol. 6, pp. 144-150, 1958.

[Gant., 1] GANTMACHER, F., "The Theory of Matrices", Vol. 1, Chelsea, New York, 1959.

[Gia., Kal. & Kar., 1] GIANNAKOPOULOS, C., KALOGEROPOULOS, G. and KARCANIAS, N., " The Grassmann variety of nondynamic compensators and the determinantal assignment problem of linear systems", Bull. of Greek Math. Soc., Vol. 24, pp. 33-57, 1985.

[Gia. & Kar., 1] GIANNAKOPOULOS, C. and KARCANIAS, N., "Pole assignment of strictly proper and proper linear systems by constant output feedback", Int. J. Control, Vol. 42, pp. 543-565, 1985.

[God., 1] GODMENT, R., "Algebra", Hermann, Paris, 1968.

[Gol., Klem. & Stew., 1] GOLUB, G., KLEMA, V. and STEWART, G., "Rank degeneracy and Least Squares Problems", STAN-CS-76-559, Univer. of Stanford, Dept. of Comp. Scien., 1976.

[Gol. & Loan, 1] GOLUB, G. and VAN LOAN, F., "Matrix Computations", North Oxford Academic, Oxford, 1983.

[Gol. & Rein., 1] GOLUB, G. and REINSCH, C., "Singular Value Decomposition and Least Squares Solutions", Num. Math., Vol. 14, pp. 403-420.

[Gol. & Var., 1] GOLUB, G. and VARAH, J., "On a characterization of the best l_2-scaling of a matrix", SIAM J. Num. Anal., vol. 11, No. 3, pp. 472-479, 1974.

[Ham., 1] HAMMER, J., "Non-Linear Systems: Stability and Rationality", Int. J. Control, Vol. 40, pp. 1-35, 1984.

[Hayn., 1] HAYNSWORTH, E., "Reduction of a matrix using properties of the Schur complement", Lin. Alg. and its Appl., Vol. 3, pp. 23-29, 1970.

[Hir. & Sma., 1] HIRSCH, M.W. and SMALE, S., "Differential Equations, Dynamical Systems and Linear Algebra", Academic Press, New York, 1974.

[Hod. & Ped., 1] HODGE, W. and PEDOE, D., "Methods of Algebraic Geometry", Vol. 1,2, Cambr. Univ. Press, 1952.

[Horn, 1] HORN, R., "Matrix Analysis", Cambr. Univ. Press, Cambridge, 1985.

[Jaf. & Kar., 1] JAFFE, S. and KARCANIAS, N., "Matrix Pencil characterization of almost (A,B)-invariant subspaces: A classification of geometric concepts", Int. J. Control, Vol. 33, pp. 51-93, 1981.

[Jel., 1] JELTSCH, R., "An optimal fraction free Routh array", Int. J. Control, Vol. 30, No. 4, pp. 661-668.

[Kail., 1] KAILATH, T., "Linear Systems", Prentice Hall, Englewood Cliffs, N.J., 1980.

[Kal., 1] KALMAN, R.E., "On partial realizations, transfer functions and canonical forms", ACTA, Polytechnica Scandinavica, Ma 31, pp. 9-32, 1979.

[Kal., 2] KALMAN, R.E., "Irreducible realizations and the degree of a rational matrix", SIAM J., Vol. 13, pp. 520-544, 1965.

[Kal., 3] KALMAN, R.E., "Mathematical Description of linear dynamical systems", SIAM J. Control, Vol. 1, pp. 152-192, 1963.

[Kal., Fal. & Arb., 1] KALMAN, R., FALB, P. and ARBIB, M., "Topics in Mathematical System Theory", McGraw Hill, New York, 1969.

[Kar., 1] KARCANIAS, N. "Invariance properties and characterization of the greatest common divisor of a set of polynomials", Int. J. Control, Vol. 46, pp. 1751-1760, 1987.

[Kar., 2] KARCANIAS, N., "Matrix pencil approach to geometric system theory", Proc. IEE, Vol. 126, pp. 585-590, 1979.

[Kar., 3] KARCANIAS, N., "Survey of Relevant Issues from Control Theory", ESPRIT 2090, EPIC, Deliverable D 1.1, 1989.

[Kar., 4] KARCANIAS, N., "Lecture Notes on Exterior Algebra and System Theory Applications", Control Engin. Centre, Department of Electrical Electronic and Information Engineering, City University, London, 1983.

[Kar., 5] KARCANIAS, N., "System Theoretic Characterizations of the Greatest Common Divisor of a set of polynomials", IMA Conf. on Control Th., Bradford, 1988.

[Kar. & Gia., 1] KARCANIAS, N. and GIANNAKOPOULOS, C., "Grassmann invariants, almost zero and the determinantal zero, pole assignment problems of linear systems", Int. J. Control, Vol. 40, pp. 673-698, 1984.

[Kar. & Gia., 2] KARCANIAS, N. and GIANNAKOPOULOS, C., "Grassmann Matrices, Decomposability of Multivectors and the Determinantal Assignment Problem", Linear Circuits Systems and Signal Processing: Theory and Appl., Ed. C. I. Byrness etc; North Holland, pp. 307-312, 1988.

[Kar. & Gia., 3] KARCANIAS, N. and GIANNAKOPOULOS, C., "Necessary and sufficient conditions for zero assignment by constant squaring down", Lin. Alg. and its Appl. Special Issue on Control Theory, Vol. 122/123/124, pp. 415-446, 1989.

[Kar., Gia. & Hub., 1] KARCANIAS, N., GIANNAKOPOULOS, C. and HUBBARD, M., "Almost zeros of a set of polynomials of R[s]", Int. J. Control, Vol. 38, pp. 1213-1238, 1983.

[Kar. & Had., 1] KARCANIAS, N. and HADAD-ZARIF, M., "Generic Values of Linear Systems Invariants", Control Engin. Centre, Res. Report, City Univ., under preparation.

[Kar., Lai. & Gia., 1] KARCANIAS, N., LAIOS, B. and GIANNAKOPOULOS, C., "The Decentralised Determinantal Assignment problem: Fixed and almost fixed modes and zeros", CEC/NK.BL.CG/33, Control Engin. Centre, City Univ., 1986.

[Kar. & MacB., 1] KARCANIAS, N. and MacBEAN, P., "Structural invariants and canonical forms of linear multivariable systems", 3rd IMA Conf. on Control Theory, Academic Press, pp. 257-282, 1981.

[Knu., 1] KNUTH, D., "The art of computer programming", Vol. 2: Seminumerical Algorithms, Addison Wesley, Reading Mass, 1969.

[Kron., 1] KRONSJO, L., "Algorithms: Their Complexity and Efficiency", John Wiley, 1987.

[Kuc., 1] KUCERA, K., "Discrete Linear Control: The Polynomial Equation Approach", John Wiley & Sons, New York, 1979.

[Lai., 1] LAIDACKER, M., "Another theorem relating Sylvester's matrix and the greatest common divisor", Math. Mag., Vol. 42, pp. 126-128, 1969.

[Laub, 1] LAUB, A., "A Schur method for solving algebraic Ricatti equations", IEEE Trans. Aut. Control, Vol. AC-24, pp. 913-921, 1979.

[Lips., 1] LIPSCHUTZ, S., "Linear Algebra", Schaum's Outline Series, MacGraw Hill, 1974.

[Mac., 1] MacDUFFEE, C., "Some applications of matrices in theory of equations", Amer. Math. Month., Vol. 57, pp. 154-161, 1950.

[Macl. & Bir., 1] MACLANE, S. and BIRKHOFF, G., "Algebra", MacMillan, London, 1967.

[Mar., 1] MARATOS, N., "Exact Penalty function algorithms for finite dimensional and control optimization problems", Ph. D. Thesis, Imperial College, London, 1978.

[Mar. & Her., 1] MARTIN, C. and HERMANN, R., "Applications of algebraic geometry to systems theory: The MacMillan degree and Kronecker indices,...", SIAM J. Control & Opt., Vol. 16, pp. 743-755, 1978.

[Marc., 1] MARCUS, M., "Finite Dimensional Multilinear Algebra", Vol. 1,2, Marcel Dekker, New York, 1973.

[Marc. & Minc, 1] MARCUS, M. and MINC, H., "A Survey of Matrix Theory and Matrix Inequalities", Allyn and Bacon, Boston, 1964.

[Mard., 1] MARDEN, M., "The Geometry of Zeros of a Polynomial in a complex variable", Amer. Math. Soc. Publications, 1949.

[Mit. & Kar., 1] MITROULI, M. and KARCANIAS, N., "Lists of programs for nongeneric computations", Control Engin. Centre, Res. Report, City University, CEC/MM.NK./102, 1991.

[Nob. & Dan., 1] NOBLE, B. and DANIEL, J., "Applied Linear Algebra", Prentice Hall, Englewood Cliffs, 1977.

[Pac. & Bar., 1] PACE, S. and BARNETT, S., "Comparison of algorithms for calculation of g.c.d. of polynomials", Int. J. Systems Sci., Vol. 4, pp. 211-226, 1973.

[Pop., 1] POPOV, V.M., "Some properties of control systems with matrix transfer functions" Lecture Notes in Math, Vol. 144, Springer Verlag, pp. 169-180, 1969.

[Ros., 1] ROSENBROCK, H., "State Space and Multivariable Theory", Nelson, London, 1970.

[Ros., 2] ROSENBROCK, H., "Computer-Aided Control System Design", Academic Press, London, 1974.

[Stew., 1] STEWART, G., "Introduction to Matrix Computations", Academic Press, 1973.

[Temp., 1] TEMPELMEIER, U., "A new proof of the cross-rule for the ε-algorithm based on Schur complements", Jour. of Comp. and Appl. Math., Vol. 21, pp. 55-61, 1988.

[VanDoor., 1] Van DOOREN P., "The generalized eigenstructure problem in Linear System Theory", IEEE Trans. Aut. Control, Vol. AC-26, pp. 111-129, 1981.

[Vard. & Kar., 1] VARDULAKIS, A.I.G. and KARCANIAS, N., "Structure, Smith-McMillan form and coprime MFDs of a rational matrix inside a region $P=\Omega U\{\infty\}'$", Int. J. Control, Vol. 38, pp. 927-957, 1983.

[Vard. & Kar., 2] VARDULAKIS, A.I.G. and KARCANIAS, N., "On the Stable Exact Model Matching and Stable Minimal Design Problems", Multiv. Control: New Concepts and Tools, Ed. S.G. Tzafestas, D. Reidel Co., pp. 233-263, 1984.

[Vard., Lim. & Kar., 1] VARDULAKIS, A.I.G., LIMEBEER, D.J. and KARCANIAS, N., "Structure and Smith-McMillan form of rational matrix at infinity", Int. J. Control, Vol. 35, pp. 701-725, 1982.

[Vard. & Sto., 1] VARDULAKIS, A.I.G. and STOYLE, P.N.R., "Generalized Resultant Theorem", J. Inst. of Maths. and its Appl., Vol. 22, pp. 103-141, 1979.

[Vid., 1] VIDYASAGAR, M., "Control System Synthesis: A Factorization Approach", MIT Press, Cambridge, Mass, 1985.

[Wang, 1] WANG, C., "Gramian expansions and their applications", Utilitas Mathematica, Vol. 15, pp. 97-111.

[War. & Eck., 1] WARREN, M.E. and ECKBERG, A.E., "On the dimensions of controllability subspaces: A characterisation via polynomial matrices

and Kronecker invariants", SIAM J. Control, Vol. 33, pp. 434-445, 1975.

[Wein., 1] WEINSTOCK, R., "Greatest common divisor of several integers and an associated linear diophantine equation", Amer. Math. Mon., Vol. 67, pp. 664-667, 1960.

[Wil., 1] WILLEMS, J.C., "Models for Dynamics", Systems and Control Group Report, Math. Institit., Univ. of Groningen. To appear in Dynamics Reported, 1987.

[Wilk., 1] WILKINSON, J., "Rounding Errors in Algebraic Processes", Her Majesty's Stationery Office, London, 1963.

[Wilk., 2] WILKINSON, J., "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford, 1965.

[Wol., 1] WOLOVICH, W.A., "Linear Multivariable Systems", Applied Math. Sciences, Vol. 11, Springer Verlag, New York, 1974.

[Won., 1] WONHAM, W.M., "Linear Multivariable Control: A Geometric Approach", Springer Verlag, New York Sec. Edit., 1984.