Institut National d'Electricité et d'Electronique

INELEC - Boumerdes

DEPARTMENT OF RESEARCH

ALGORITHMS FOR THE FACTORIZATION OF MATRIX POLYNOMIALS

THESIS

Presented in Partial fulfilment of the requirements of the

DEGREE OF MAGISTER

in Electronic System Engineering

by

Abdelhakim DAHIMENE

Acknowledgements

I wish to express my sincere appreciation to Dr. K. Hariche for his guidance and encouragement throughout this thesis.

I wish also to thank all the staff of INELEC's library and especially Mrs. F. Zerari and Mr. Y. Boucetta without whom no research work could be done at INELEC.

To my wife, Farida, I thank her for her infinite patience and love.

Abstract

The factorization (root finding) of scalar polynomials is an important tool of analysis and design for linear systems. This thesis is a part of an ongoing effort to generalize these tools to multivariable systems via the factorization of matrix polynomials.

The main contributions of this thesis can be summarized as follows:

(1) the development of the Q.D. algorithm, which is a global method capable of producing a complete factorization of a matrix polynomial;

(2) establishment of an existence theorem for the Q.D. algorithm;

(3) production of convergence theorems for the Q.D. algorithm;

(4) study of the initialization of the algorithm;

(5) applicability of Broyden's method to matrix polynomial problems.

As a by-product, some important results have been produced:

(6) location of the latent roots of a matrix polynomial in the complex plane;

(7) Investigation of the existence of the solvents of a monic matrix polynomial;

(8) derivation of an incomplete partial fraction expansion of a matrix rational fraction.

Table of Contents

Chapter 1

Introduction

In the early days of Control and System theory, frequency domain techniques were the principal tools of analysis, modelling and design for linear systems. The work of Nyquist [29] and Bode [3] laid down the foundations of feedback and control science. However, only dynamic systems that can be modelled by a scalar m^{th} order linear differential equation with constant coefficients are amenable to this type of analysis (see ref. [9] for example). Those systems have a single input and a single output (SISO).

In this case, the transfer function is a ratio of two scalar polynomials. The dynamic properties of the system (time response, stability, etc...) depend on the roots of the denominator or in other words on the solution of the underlying homogeneous differential equation (difference equation)[1].

---

1 It is understood that the same results apply to discrete time systems. The use of the z operator as forward shift operator transforms a scalar m^{th} order difference equation with constant coefficients into a m^{th} order transfer function. This transfer function is also a rational fraction.

The advent of powerful computing facilities modified the view of system engineers and the emphasis was put on time domain analysis (the State Space approach) [44,45]. This approach is essentially the modelling of the system by a first order differential (difference) equation. It allows the use of matrix theory [26] and powerful numerical procedures [41].

In the state space approach, the dynamic properties of the system under study depend mostly on the eigenvalues of a state matrix. This method has the advantage to model with the same ease single input single output systems and multiple input multiple output systems (MIMO).

When one studies high order MIMO systems, the size of the matrices involved becomes prohibitive. This is why there is a reappearance nowadays of transfer function (which become rational matrices) description [5,21,24]. In this context, the dynamic properties of the system under study are determined by the latent roots of a polynomial matrix. This is why we find quite a lot of publications at the present time about those matrices in system and control journals. References [1,31,32,35] are a few sample of this trend.

To clarify these concepts, let us consider the following dynamic system:

$$A_0 y^{(m)}(t) + A_1 y^{(m-1)}(t) + \ldots + A_m y(t) = B_1 u(t) + B_2 u'(t) + \ldots + B_m u^{(m-1)}(t) \quad (1.1)$$

$$y(t) \in C^r \qquad u(t) \in C^p$$

$$A_k \in C^{r \times r} \quad B_k \in C^{r \times p}$$

The "modes" of this system are the solutions of the homogeneous equation:

$$A_0 y^{(m)}(t) + A_1 y^{(m-1)}(t) + \ldots + A_m y(t) = 0 \tag{1.2}$$

using $y(t) = Ce^{\lambda t}$, $C \in C^r$, λ is a complex number, equation (1.2) is transformed to:

$$[A_0 \lambda^m + A_1 \lambda^{m-1} + \ldots + A_m] C e^{\lambda t} = 0 \tag{1.3}$$

In other words, (1.3) implies that $y(t)$ belongs to the null space of the following square matrix:

$$A(\lambda) = A_0 \lambda^m + A_1 \lambda^{m-1} + \ldots + A_m \tag{1.4}$$

The square matrix $A(\lambda)$ is a matrix polynomial of degree m. The general solution of (1.2) is derived in reference [10].

Another approach to equation (1.1) can be found by using the Laplace transform. (1.1) is transformed to:

$$Y(s) = [A(s)]^{-1} B(s) U(s) \tag{1.5}$$

where A(s) is the previously defined matrix polynomial, B(s) is an r x p matrix polynomial with coefficients $B_1, \ldots, B_m$, U(s) and Y(s) are the Laplace transforms of u(t) and y(t) respectively. Equation (1.5) defines a matrix transfer function for the system (1.1)[2]. This particular

2 We obtain the same result by applying the z-transform to a difference equation. Thus λ as a variable in a matrix polynomial can represent either the variable s of the Laplace transform or the variable z (or z^{-1}) of the z-transform.

matrix is called the "left matrix fraction description" (LMFD) of the system (1.1). A right matrix fraction (RMFD) can be defined [24]. In this representation the matrix polynomial A(s) is a sort of denominator of the transfer matrix.

A state space description of system (1.1) can be found by transforming the m^{th} order differential equation (1.1) into a first order differential equation. For details on this analysis, the reader should consult references [5,21,44].

To see the importance of factorization, let us consider equation (1.2). we assume that $A(\lambda) = A_1(\lambda)A_2(\lambda)$. In this case (1.2) can be simplified to:

$$A_1\left(\frac{d}{dt}\right)y_1(t) = 0$$

$$A_2\left(\frac{d}{dt}\right)y(t) = y_1(t) \tag{1.6}$$

λ is the derivative operator in this case [10]. Thus the m^{th} order differential equation has been transformed into two smaller order differential equations.

In chapter 2, we will derive a partial fraction expansion for the inverse of a matrix polynomial and in chapter 4, we will demonstrate that a factorization of the "denominator" of a transfer matrix leads to an incomplete partial fraction expansion. Those partial fractions are of course transfer functions of reduced order linear systems.

The purpose of this thesis is to derive a global method for computing a particular factorization. This global method is then followed by a local (but fast converging) method. There have been some algorithms that have been published [8,23,34,39]. However, those methods can factorize only a linear factor at a time. The global method that we propose to use is a generalization of the scalar quotient-difference (Q.D.) algorithm [15-18]. The use of the Q.D. algorithm to matrix polynomial factorization has been suggested by Hariche in [14]. The local method that we propose in our work is Broyden's algorithm [6] which presents some advantages over the classical Newton's method [23,34].

In the following section, we give a brief presentation of the thesis.

Chapter two provides the basic theoretical tools for the rest of the thesis.

Chapter three examines the existing global methods. The proofs presented therein are different from the original ones.

Chapter four constitutes the heart of the thesis and represents the main contribution of this work. In this chapter, we study the convergence and the conditions of existence of the matrix Q.D. algorithm.

Chapter five completes our analysis by providing an alternate proof of convergence using block matrix methods.

In chapter six, we analyse local techniques and look at the applicability of Broyden's method to our problem.

In chapter seven, we present some numerical results. We have tested the Q.D. algorithm and Broyden's method on a large number of matrix polynomials.

Finally, in chapter eight, we provide the conclusions of this thesis and we suggest topics for further research.

Chapter 2

Theory of Matrix Polynomials

2.1 General Definitions.

We have seen in the introduction that matrix polynomials arise naturally in the study of linear time invariant dynamic systems. In this chapter we will attempt to give a more precise meaning to these polynomials. There exist a quite large confusion in their definition. Dennis et Al. in [7], Gohberg et Al. in [10] and Kucera in [24] give three different definitions that do not correspond to the same entity. The definition we will use in our work will be the one used by Gohberg et Al.

Definition 2.1:

Given the set of $r \times r$ complex matrices $A_0, A_1, \ldots, A_m$, the following matrix valued function of the complex variable λ is called a matrix polynomial of degree m and order r:

$$A(\lambda) = A_0\lambda^m + A_1\lambda^{m-1} + \ldots + A_{m-1}\lambda + A_m. \tag{2.1}$$

An equivalent definition is the one of λ-matrices which are also called polynomial matrices in Kucera [24]:

Definition 2.2:

The following $r \times r$ matrix:

$$A(\lambda) = \begin{pmatrix} a_{11}(\lambda) & \dots & a_{1r}(\lambda) \\ . & \dots & . \\ a_{r1}(\lambda) & \dots & a_{rr}(\lambda) \end{pmatrix} \qquad (2.2)$$

is called a λ-matrix of order r where $a_{ij}(\lambda)$ are scalar polynomials over the field of complex numbers C.

Dennis et Al. definition of a matrix polynomial [7] is called in our work the right evaluation of the matrix polynomial $A(\lambda)$.

Definition 2.3:

The following $r \times r$ matrix valued function of the $r \times r$ matrix X is called the right evaluation of the matrix polynomial $A(\lambda)$ at X:

$$A_R(X) = A_0 X^m + A_1 X^{m-1} + \dots + A_m. \qquad (2.3)$$

We also define the left evaluation of $A(\lambda)$ at X by:

$$A_L(X) = X^m A_0 + X^{m-1} A_1 + \dots + A_m. \qquad (2.4)$$

So we can see that the two definitions 2.1 and 2.2 are equivalent. However, definition 2.1 emphasises the polynomial character of the matrix polynomial while definition 2.2 emphasises the matrix one. We will use most of the time definition 2.1. Definition 2.3 is very different and will be mostly used when we will present the local methods.

The following definitions are also useful.

Definition 2.4:

The matrix polynomial $A(\lambda)$ is called:

- *Monic if A_0 is the identity matrix*
- *Comonic if A_m is the identity matrix*
- *Unimodular if its determinant is a nonzero constant and*
- *Regular if its determinant is not identically zero.*

There are also definitions (i.e. the Smith and Hermite normal forms) which are very useful in the study of λ-matrices. However, because we will not use those concepts in the rest of the presentation, we will not present them here. The interested reader should consult the appropriate literature on matrix theory (i.e. Lancaster et Al. ref.[26]).

2.2 Latent Structure of Matrix Polynomials.

Definition 2.5:

The complex number λ_0 is called a latent root if it is a solution of the scalar polynomial equation $\det A(\lambda)=0$.

The non trivial vector v, solution of $A(\lambda_0)v=0$ is called a primary right latent vector associated with λ_0.

From the definition we can see that a latent problem of a matrix polynomial is a generalization of the concept of eigenproblem for square matrices. Indeed, we can consider the classical eigenvalue/vector problem as finding the latent root/vector of a linear matrix polynomial $\lambda I - A$. An interesting problem is the number of latent roots in a given region of the complex plane. This is answered by the following theorem.

Theorem 2.1:

The number of latent roots of the regular matrix polynomial $A(\lambda)$ *in the domain* $\mathcal{D}$ *enclosed by a contour* Γ *is given by:*

$$N = \frac{1}{2\pi j}\oint_{\Gamma} trace[A^{-1}(\lambda)A'(\lambda)]d\lambda \qquad (2.6)$$

[1]

each latent root being counted according to its multiplicity.

Proof:

In this proof, we will make use of the following result from the theory of functions of a complex variable (see Henrici ref.[15] for example):

"The number of zeros of a function $f(z)$ analytic in a domain $\mathcal{D}$ enclosed by a contour Γ is given by:

$$N = \frac{1}{2\pi j}\oint_{\Gamma} \frac{f'(z)}{f(z)}dz$$

each zero being counted according to its multiplicity."

---

1 $A'(\lambda)$ is the derivative of $A(\lambda)$.

Let us now consider the scalar polynomial $d(\lambda) = \det A(\lambda)$. Being analytic in any domain in the complex plane then the number of its roots inside a curve Γ is given by:

$$N = \frac{1}{2\pi j}\oint_{\Gamma} \frac{d'(\lambda)}{d(\lambda)} d\lambda$$

in ref. [25] Lancaster shows that:

$$\frac{d'(\lambda)}{d(\lambda)} = trace[A^{-1}(\lambda)A'(\lambda)]$$

(Q.e.d).

At this point, we can also define the spectrum of a matrix polynomial $A(\lambda)$ as being the set of all its latent roots (notation $\sigma(A)$). It is essentially the same definition as the one of the spectrum of a square matrix.

A generalization of the latent root/vector is the Jordan chain which is defined by:

Definition 2.6:

A set of vectors $x_0, x_1, \ldots, x_k \in C^r$ is called a right Jordan chain of length k+1 associated with the latent root λ_0 and primary right latent vector x_0 if they satisfy the relations:

$$\sum_{p=0}^{j} \frac{1}{p!} A^{(p)}(\lambda_0) x_{j-p} = 0, \quad j = 0, 1, \ldots, k \tag{2.6}$$

[2]

---

2 $A^{(p)}(\lambda)$ is the p^{th} order derivative of $A(\lambda)$.

The set of all Jordan chains of a particular monic matrix polynomial can be grouped in the following triple:

Definition 2.7: *(the Jordan Triple)*

The following matrices X, J and Y of size respectively $r \times mr$, $mr \times mr$ and $mr \times r$ is called a Jordan triple of the monic matrix polynomial $A(\lambda)$ of degree m and order r.

J is a block diagonal matrix composed of Jordan blocks each corresponding to a particular latent root.

Each column of X is an element of a Jordan chain associated with the appropriate Jordan block in J and Y is a matrix of left latent vectors which can be computed by:

$$\begin{bmatrix} X \\ XJ \\ . \\ . \\ XJ^{m-1} \end{bmatrix} Y = \begin{bmatrix} 0 \\ . \\ . \\ 0 \\ I \end{bmatrix}$$

Example 2.1:

let

$$A(\lambda) = \begin{pmatrix} \lambda^3 & \sqrt{2}\lambda^2 - \lambda \\ \sqrt{2}\lambda^2 + \lambda & \lambda^3 \end{pmatrix}$$

this matrix polynomial has the following spectrum

$$\sigma(A) = \{0, 1, -1\}$$

each latent root has an algebraic multiplicity of 2.

To the latent root 0 correspond 2 independent latent vectors which can be taken as:

$$v_1^{(0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \;;\; v_2^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

To the latent root 1 correspond only one latent vector, and we have to complete it by a generalized latent vector belonging to the Jordan chain associated with the latent root 1. $v_{10}^{(1)}$ is the primary latent vector:

$$v_{10}^{(1)} = \begin{pmatrix} -\sqrt{2}+1 \\ 1 \end{pmatrix}$$

and $v_{11}^{(1)}$ is a generalized latent vector:

$$v_{11}^{(1)} = \begin{pmatrix} \sqrt{2}-2 \\ 0 \end{pmatrix}$$

the latent root -1 has exactly the same character. We have the following Jordan chain associated with it:

$$v_{10}^{(-1)} = \begin{pmatrix} \sqrt{2}+1 \\ 1 \end{pmatrix} \;;\; v_{11}^{(-1)} = \begin{pmatrix} \sqrt{2}+2 \\ 0 \end{pmatrix}$$

and hence the Jordan triple associated with $A(\lambda)$ is:

$$J = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$X = \begin{pmatrix} 1 & 0 & -\sqrt{2}+1 & \sqrt{2}-2 & \sqrt{2}+1 & \sqrt{2}+2 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$Y=\begin{pmatrix} 0 & 1 \\ -1 & 0 \\ \frac{\sqrt{2}+2}{4} & 0 \\ \frac{-\sqrt{2}-1}{4} & \frac{1}{4} \\ \frac{-\sqrt{2}+2}{4} & 0 \\ \frac{-\sqrt{2}+1}{4} & -\frac{1}{4} \end{pmatrix}$$

Another triple which is very important in the study of matrix polynomials is the standard triple defined by:

Definition 2.8: *(Standard triple)*

A set of 3 matrices (Z,T,W) is called a standard triple of the monic matrix polynomial $A(\lambda)$ if it is related to the Jordan triple (X,J,Y) by the following similarity transformation:

Let S be a nonsingular $mr\times mr$ matrix, then

$$Z=XS^{-1}\ ;\ T=SJS^{-1}\ ;\ W=SY \tag{2.7}$$

The standard triple allows a representation of a matrix polynomial using its spectral information and this is shown in the next theorem:

Theorem 2.2:

Let $A(\lambda)$ be a monic matrix polynomial of degree m and order r with standard triple (X,T,Y), then $A(\lambda)$ has the following representations:

1) right canonical form:

$$A(\lambda)=\lambda^m I-XT^m(V_1+V_2\lambda+\ldots+V_m\lambda^{m-1}) \tag{2.8}$$

where V_i are $rm\times r$ matrices such that:

$$[V_1,\ldots,V_m]=\begin{bmatrix} X \\ XT \\ \cdot \\ \cdot \\ XT^{m-1}\end{bmatrix}^{-1} \tag{2.9}$$

2) left canonical form:

$$A(\lambda)=\lambda^m I-(W_1+\lambda W_2+\ldots+\lambda^{m-1}W_m)T^m Y \tag{2.10}$$

where W_i are $r\times rm$ matrices such that:

$$\begin{bmatrix} W_1 \\ W_2 \\ \cdot \\ \cdot \\ W_m\end{bmatrix}=[Y,TY,\ldots,T^{m-1}Y]^{-1} \tag{2.11}$$

3) Resolvent form:

$$[A(\lambda)]^{-1}=X(\lambda I-T)^{-1}Y \qquad \lambda\notin\sigma(A) \tag{2.12}$$

Proof: see Gohberg et Al. ref. [10,11,12].

The following standard triples associated with $A(\lambda)$ will be used quite extensively in the rest of the presentation:

The Lower Block Companion Form:

$$P_1 = [I, 0, \ldots, 0] \qquad Q_1 = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ I \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & I & 0 & \ldots & 0 & 0 \\ 0 & 0 & I & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \ldots & \cdot & \cdot \\ 0 & 0 & 0 & \ldots & 0 & I \\ -A_m & -A_{m-1} & -A_{m-2} & \ldots & -A_2 & -A_1 \end{bmatrix} \tag{2.13}$$

The Right Block Companion Form:

$$P_2 = [0, \ldots, 0, I] \qquad Q_2 = \begin{bmatrix} I \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 0 & \ldots & 0 & -A_m \\ I & 0 & \ldots & 0 & -A_{m-1} \\ 0 & I & \ldots & 0 & -A_{m-2} \\ \cdot & \cdot & \ldots & \cdot & \cdot \\ 0 & 0 & \ldots & 0 & -A_2 \\ 0 & 0 & \ldots & I & -A_1 \end{bmatrix} \tag{2.14}$$

We must remark that all the matrices defined above have block elements which are themselves $r \times r$ matrices. From the previous definition, it is clear that the Jordan structure of a matrix polynomial $A(\lambda)$ is directly related to the Jordan structure of its block companion matrices. The relation between the eigenvectors of C_1 and the latent vectors of $A(\lambda)$ is shown in Hariche ref. [14].

2.3 The Division Algorithm.

One can also remark that the coefficient space of matrix polynomials is the non-commutative ring of square matrices. It is useful at this point to present some general theorems from Algebra. The most important ones being the "division theorem" and the "remainder theorem".

For this presentation we consider a ring R with identity and x is an indeterminate over R. We define also $R[x]$ as the space of polynomials over R.

Theorem 2.3: *(The division theorem)*

Let:

$$a(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_n \in R[x]$$
$$b(x) = b_0 x^m + b_1 x^{m-1} + \dots + b_m \in R[x]$$

where $a_0 \neq 0$ and b_0 is not a zero divisor, then there exist unique polynomials $q(x)$ and $r(x)$ in $R[x]$ such that:

$$a(x) = q(x)b(x) + r(x) \tag{2.15}$$

and $r(x) = 0$ or degree of $r(x)$ < degree of $b(x)$.

Similarly, there exist unique polynomials $s(x)$ and $u(x)$ in $R[x]$ such that:

$$a(x) = b(x)s(x) + u(x) \tag{2.16}$$

and $u(x) = 0$ or degree of $u(x)$ < degree of $b(x)$.

Proof: see Marcus ref. [28].

Theorem 2.4: *(The remainder theorem)*

If $a(x) \in R[x]$, $a(x) \neq 0$ and $c \in R$, then there exist unique polynomials $q(x)$ and $s(x)$ in $R[x]$ such that:

$$\begin{aligned} a(x) &= q(x)(x-c) + a_R(c) \\ a(x) &= (x-c)s(x) + a_L(c) \end{aligned} \tag{2.17}$$

where $a_R(c)$ and $a_L(c)$ are respectively the right and left evaluation of $a(x)$ at c.

Proof: see Marcus ref. [28].

If $a_R(c) = 0$, **c** is called a right solvent of $a(x)$ and if $a_L(c) = 0$, **c** is called a left solvent of $a(x)$. If the ring R is commutative, **c** is called a root of $a(x)$.

One can now specialize theorems 2.3 and 2.4 to matrix polynomials and state the following corollaries:

Corollary 2.1: *(The division theorem)*

Given the matrix polynomial $A(\lambda)$ with $A_0 \neq 0$ and $B(\lambda)$ with B_0 nonsingular, there exist unique matrix polynomials $Q(\lambda)$ and $R(\lambda)$ such that:

$$A(\lambda) = Q(\lambda)B(\lambda) + R(\lambda) \tag{2.18}$$

$R(\lambda) = 0$ or degree of $R(\lambda)$ < degree of $B(\lambda)$.

Similarly, there exist unique matrix polynomials $S(\lambda)$ and $U(\lambda)$ such that:

$$A(\lambda)=B(\lambda)S(\lambda)+U(\lambda) \tag{2.19}$$

$U(\lambda)=0$ or degree of $U(\lambda)$ < degree of $B(\lambda)$.

When the divisor is linear, i.e. $B(\lambda)=\lambda I-X$, we can write:

Corollary 2.2: *(The remainder theorem)*

Given $A(\lambda)\neq 0$ and $X\in C^{r\times r}$, there exist unique matrix polynomials $Q(\lambda)$ and $S(\lambda)$ such that:

$$\begin{aligned} A(\lambda)&=Q(\lambda)(\lambda I-X)+A_R(X)\\ A(\lambda)&=(\lambda I-X)S(\lambda)+A_L(X). \end{aligned} \tag{2.20}$$

Corollaries 2.1 and 2.2 are simply a rewriting of theorems 2.3 and 2.4. We can now state the following relation between a matrix polynomial and its right and left evaluation:

$$A(\lambda)=A_R(\lambda I)=A_L(\lambda I) \tag{2.21}$$

Corollary 2.2 also gives the fundamental relation that exists between right solvent and right linear factor, left solvent and left linear factor:

$$\begin{aligned} A_R(X)=0 \ \ iff \ \ A(\lambda)=Q(\lambda)(\lambda I-X)\\ A_L(X)=0 \ \ iff \ \ A(\lambda)=(\lambda I-X)S(\lambda) \end{aligned} \tag{2.22}$$

Along with this algebraic framework, the following definitions (from Kucera ref.[24], Gohberg et Al. ref.[10,11,12] and Hariche ref.[14]) will be used:

Definition 2.9:

Consider matrix polynomials A,B and C of order r. If $A = BC$ then B is a left divisor of A and A is a right multiple of B, while C is a right divisor of A and A is left multiple of C.

Consider matrix polynomials A and B of order r. If G_1 is a left divisor of both A and B, then it is termed a common left divisor of A and B; furthermore, if G_1 is a right multiple of every common left divisor of A and B, then G_1 is a greatest common left divisor of A and B.

Similarly, if G_2 is a right divisor of both A and B, it is termed a common right divisor of A and B and if G_2 is a left multiple of every common right multiple of A and B, then G_2 is a greatest common right divisor of A and B.

According to these definitions, it is clear that if a matrix polynomial divides another, then the remainder of the division is equal to zero. However, we can also remark that we can define divisors even in cases where the division algorithm cannot be used.

Another property of divisors of $A(\lambda)$ is that their Jordan chains are part of the Jordan chain of $A(\lambda)$ [14]. Thus if we have an algorithm that can factorize matrix polynomials, we will have a tool for the study of their Jordan structure. Linear factors become thus very important and it is apparent that the Jordan structure of solvents is part of the Jordan structure of $A(\lambda)$. Given a particular matrix R, establishing whether it is a solvent of a matrix polynomial $A(\lambda)$ or no is important because this might lead to discover methods for solving the matrix equation $A_R(X)=0$. Hariche in ref.[14] characterizes R via its Jordan structure.

Theorem 2.5:

Given the matrix polynomial $A(\lambda)$ and the $r\times r$ matrix $R = MJM^{-1}$, R is a right solvent of $A(\lambda)$ if and only if $\overline{A}$ is rank deficient. [3]

$$\overline{A}=(J^T)^m\otimes A_0+(J^T)^{m-1}\otimes A_1+\ldots+J^T\otimes A_{m-1}+I\otimes A_m$$

Proof: see Hariche ref. [14].

Gohberg et Al. in [10] provide another characterization via invariant subspace of the block companion matrix C_1.

Theorem 2.6:

The monic matrix polynomial $A(\lambda)$ has a right solvent R if and only if there exists an invariant subspace $\mathcal{M}$ of the block companion matrix C_1 of the form:

$$\mathcal{M}=Im\begin{bmatrix} I \\ R \\ . \\ R^{m-1}\end{bmatrix}$$

4

---

3 $\otimes$ is the Kronecker product of matrices (see [13]).

4 Im T stands for the range space (or image) of the matrix T.

Proof: see Gohberg et Al. ref.[10].

This theorem is the basis of the block power algorithm [39] and as we will see later it can be used also to characterize the block Bernoulli method [8]. The next theorem is based on the algorithm of synthetic division applied to matrix polynomial.

Let R be a right solvent of the matrix polynomial $A(\lambda)$, then using the relations (2.22), we can write:

$$A(\lambda) = Q(\lambda)(\lambda I - R)$$

where

$$Q(\lambda) = \lambda^{m-1} Q_0 + \lambda^{m-2} Q_1 + \ldots + Q_{m-1}$$
$$Q_0 = I$$

we can compute the coefficients Q_i using the algorithm of synthetic division:

$$\begin{aligned} Q_0 &= I \\ Q_1 &= Q_0 A_1 + Q_0 R \\ Q_2 &= Q_0 A_2 + Q_1 R \\ &\ldots \\ Q_k &= Q_0 A_k + Q_{k-1} R \qquad k = 1, \ldots, m-1 \\ 0 &= Q_0 A_m + Q_{m-1} R \end{aligned} \tag{2.23}$$

let us introduce the following notation for the block row vector $\bar{Q}$ of size $r \times mr$:

$$\bar{Q} = (Q_{m-1} \quad Q_{m-2} \quad \cdots \quad Q_0)$$

then the set of equations (2.23) can be rephrased in block matrix form:

$$\bar{Q}C_1 = (Q_{m-1}R \quad Q_{m-2}R \quad \dots \quad Q_0R)$$

so finally

$$\bar{Q}C_1 = \bar{Q}[I \otimes R] \tag{2.24}$$

we can thus state the following theorem:

Theorem 2.7:

R is a right solvent of the monic matrix polynomial $A(\lambda)$ if and only if $\mathcal{N}ull[C_1 - I \otimes R] = r$.[5]

Proof:

Let $C = C_1 - I \otimes R$ then equation (2.24) can be written as $\bar{Q}C = 0$. However, the last block element of $\bar{Q}$ is the $r \times r$ identity matrix. This implies that $\bar{Q}$ has a rank of r. So the dimension of the null space of C is larger than r.

We also have:

$$C_1 - I \otimes R = \begin{bmatrix} -R & I & 0 & \dots & 0 \\ 0 & -R & I & \dots & 0 \\ . & . & . & \dots & . \\ 0 & 0 & 0 & \dots & I \\ -A_m & -A_{m-1} & -A_{m-2} & \dots & -A_1 - R \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

5 $\mathcal{N}ull$ stands for nullity.

with:

$$C_{11}=\begin{bmatrix}-R\\0\\.\\.\\0\end{bmatrix}\qquad C_{21}=[-A_m]$$

$$C_{22}=[-A_{m-1}\quad\dots\quad -A_1-R]$$

$$C_{12}=\begin{bmatrix}I & 0 & \dots & 0\\ -R & I & \dots & 0\\ . & . & \dots & .\\ 0 & 0 & \dots & I\end{bmatrix}$$

C_{12} is a square matrix of size $(m-1)r\times(m-1)r$ and is clearly nonsingular, this implies that the rank of C is larger (or equal) than (m-1)r so $\mathcal{N}ull[C_1-I\otimes R]\leq r$ and finally $\mathcal{N}ull[C_1-I\otimes R]=r$.

Conversely:

let $\mathcal{N}ull[C_1-I\otimes R]=r$, then $\mathcal{R}ank[C_1-I\otimes R]^T=(m-1)r$. This implies that we can find a full rank block row vector $Y=(Y_{m-1}\ \dots\ Y_0)$ of size $r\times mr$ such that:

$$Y[C_1-I\otimes R]=0 \tag{2.25}$$

If we develop equation (2.25), we obtain:

$$Y_{m-1}R=-Y_0A_m$$

$$Y_k=Y_0A_k+Y_{k-1}R \qquad k=1,\dots,m-1$$

solving recursively this set of equations, we obtain:

$$Y_0A_R(R)=Y_0(R^m+A_1R^{m-1}+\dots+A_m)=0 \tag{2.26}$$

and

$$Y_k = Y_0 A_k + Y_{k-1} R \qquad k = 1, \ldots, m-1 \quad (2.27)$$

so if $Y_0 = I$ then (2.27) defines the remaining blocks of Y in a unique manner and in this case (2.26) implies that R is a right solvent.

(Q.e.d.)

2.4 Spectral Divisors.

In the next sections we will consider only monic or comonic matrix polynomials. It is evident that results that apply to monic matrix polynomials apply also to comonic ones. If $A(\lambda)$ is a monic matrix polynomial of degree m, then $F(z) = z^m A(z^{-1})$ is a comonic matrix polynomial of same degree with inverse spectrum and same latent vectors (assuming that A does not have zero as latent root).

We have seen in the preceding section that the spectral information of a monic matrix polynomial is given by its standard triple. If we multiply two matrix polynomials, then the resulting triple is determined by the following rule:

Theorem 2.8:

If $A_k(\lambda)$ are monic matrix polynomials with standard triple (Q_k, T_k, R_k) for $k = 1,2$, then $A(\lambda) = A_2(\lambda) A_1(\lambda)$ has the following standard triple:

$$Q = [Q_1 \quad 0] \quad T = \begin{bmatrix} T_1 & R_1 Q_2 \\ 0 & T_2 \end{bmatrix} \quad R = \begin{bmatrix} 0 \\ R_2 \end{bmatrix} \qquad (2.28)$$

Proof: see ref. [10].

The algorithms that will be described in later chapters will all deal with a particular type of factorization of matrix polynomials: The spectral factorization [10].

Definition 2.10:

If $A(\lambda)=A_1(\lambda)A_2(\lambda)$ is a particular factorization of the monic matrix polynomial $A(\lambda)$, with $\sigma(A_1)\cap\sigma(A_2)=\emptyset$, then the monic matrix polynomials $A_1(\lambda)$ and $A_2(\lambda)$ are called spectral divisors of $A(\lambda)$.

Clearly, if a matrix polynomial possesses spectral divisors, then there exists a similarity transformation that can transform the block companion matrix C_1 associated with $A(\lambda)$ to a block diagonal one (see theorem 2.8). We can accomplish this transformation directly on C_1 (see Bavely et Al. ref.[4]) or act on the matrix polynomial $A(\lambda)$ using the Q.D. algorithm.

A particular class of spectral divisors is the class of left and right spectral divisors corresponding to the same set of latent roots. In this case, the matrix polynomial $A(\lambda)$ has at least two different factorizations, $A(\lambda)=A_2(\lambda)A_1(\lambda)$ and $A(\lambda)=B_1(\lambda)B_2(\lambda)$ with A_k and B_k being monic matrix polynomials, $\sigma(A_k)=\sigma(B_k)$.

The following theorem from ref. [10] gives explicitly the conditions under which the matrix polynomial $A(\lambda)$ has the above mentioned property.

Theorem 2.9:

Let $A(\lambda)$ be a monic matrix polynomial and Γ a contour consisting of regular points of $A(\lambda)$ having exactly kr eigenvalues of A (counted according to multiplicities) inside Γ. Then A has both a Γ-spectral right divisor and a Γ-spectral left divisor if and only if the following kr x kr matrix $M_{k,k}$ defined by

$$M_{k,k}=\frac{1}{2\pi j}\oint_{\Gamma}\begin{bmatrix} A^{-1}(\lambda) & \dots & \lambda^{k-1}A^{-1}(\lambda) \\ . & \dots & . \\ \lambda^{k-1}A^{-1}(\lambda) & \dots & \lambda^{2k-2}A^{-1}(\lambda)\end{bmatrix}d\lambda$$

is non singular. In this condition, the Γ-spectral right (resp. left) divisor $A_1(\lambda)=\lambda^k I+A_{11}\lambda^{k-1}+\dots+A_{1k}$ (resp. $A_2(\lambda)=\lambda^k I+A_{21}\lambda^{k-1}+\dots+A_{2k}$) is given by the formula:

$$[A_{1k}\ \dots\ A_{11}]=-\frac{1}{2\pi j}\oint_{\Gamma}[\lambda^k A^{-1}(\lambda)\ \dots\ \lambda^{2k-1}A^{-1}(\lambda)]d\lambda . M_{k,k}^{-1}$$

resp.

$$\begin{bmatrix} A_{2k} \\ . \\ . \\ A_{21}\end{bmatrix}=-M_{k,k}^{-1}.\frac{1}{2\pi j}\oint_{\Gamma}\begin{bmatrix}\lambda^k A^{-1}(\lambda) \\ . \\ . \\ \lambda^{2k-1}A^{-1}(\lambda)\end{bmatrix}d\lambda$$

.

Proof: See ref. [10].

An interesting consequence of theorem 2.9 is the case k = 1, i.e. the existence of spectral right and left solvent. Let Γ be a contour that contains r latent roots and let the r x r matrix

$$M = \frac{1}{2\pi j}\oint_{\Gamma} A^{-1}(\lambda)d\lambda \tag{2.29}$$

be nonsingular, then we have as right solvent:

$$R = \frac{1}{2\pi j}\oint_{\Gamma} \lambda A^{-1}(\lambda)d\lambda . M^{-1} \tag{2.30}$$

and as left solvent:

$$L = M^{-1} . \frac{1}{2\pi j}\oint_{\Gamma} \lambda A^{-1}(\lambda)d\lambda \tag{2.31}$$

and we see that R M = M L, i.e. M is a similarity transformation between R and L.

2.5 Complete set of solvents and complete factorization.

We have seen that solvents are quite important in the study of matrix polynomials. In this section we are going to study matrix polynomials which are completely described by a set non-interacting solvents. However, we have to present certain definitions first.

Definition 2.11:

Given the set of r x r matrices $R_1, R_2, \ldots, R_k$, the following rk x rk matrix

$$V(R_1, R_2, \ldots, R_k) = \begin{bmatrix} I & I & \ldots & I \\ R_1 & R_2 & \ldots & R_k \\ . & . & \ldots & . \\ R_1^{k-1} & R_2^{k-1} & \ldots & R_k^{k-1} \end{bmatrix} \tag{2.32}$$

is called a block Vandermonde matrix of order k.

Definition 2.12:

Given a monic matrix polynomial $A(\lambda)$, the following set of solvents $R_1, R_2, \ldots, R_m$ is called complete if the following conditions are met:

$$\sigma(R_k) \cap \sigma(R_j) = \emptyset \quad ; \; k \neq j$$

$$\bigcup_{k=1}^{m} \sigma(R_k) = \sigma(A(\lambda))$$

$$\det V(R_1, R_2, \ldots, R_m) \neq 0 \tag{2.33}$$

In this case, we can find a particularly simple standard triple and we can express the inverse of a matrix polynomial in partial fraction [14,40]. This is given by the following theorem.

Theorem 2.10:

Let $R_1, R_2, \ldots, R_m$ form a complete set of solvents for the monic matrix polynomial $A(\lambda)$, then $A(\lambda)$ admits the following standard triple:

$$X = [I \quad I \quad \dots \quad I] \qquad T = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & \dots & R_m \end{bmatrix}$$

$$Y = [V(R_1, \dots, R_m)]^{-1} \begin{bmatrix} 0 \\ . \\ . \\ 0 \\ I \end{bmatrix} \tag{2.34}$$

Proof:

Let us consider the standard triple corresponding to the lower block companion form: P_1, C_1, Q_1. From theorem 2.6, we have:

$$C_1 \begin{bmatrix} I \\ R_k \\ . \\ . \\ R_k^{m-1} \end{bmatrix} = \begin{bmatrix} I \\ R_k \\ . \\ . \\ R_k^{m-1} \end{bmatrix} . R_k$$

$$k = 1, 2, \dots, m$$

so, the block Vandermonde matrix $V(R_1, \dots, R_m)$ is a similarity transformation matrix (it is non singular by definition). We have:

$$T = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & \dots & R_m \end{bmatrix} = [V(R_1, \dots, R_m)]^{-1} . C_1 . V(R_1, \dots, R_m)$$

while $X = P_1 . V(R_1, \dots, R_m)$ and $Y = [V(R_1, \dots, R_m)]^{-1} . Q_1$.

(Q.e.d)

Let us write the block column vector Y as:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ . \\ . \\ Y_m \end{bmatrix}$$

Using theorem 2.2, we have the following result:

$$[A(\lambda)]^{-1} = X(\lambda I - T)^{-1} Y$$

giving:

$$[A(\lambda)]^{-1} = \sum_{k=1}^{m} (\lambda I - R_k)^{-1} . Y_k \qquad (2.35)$$

The above result is a partial fraction expansion of the inverse of $A(\lambda)$ [14,40]. Furthermore, since Y is the last block column of $[V(R_1,\dots,R_m)]^{-1}$, its block elements Y_k can be computed if the m block Vandermonde matrices:

$$V_k = V(R_1,\dots,R_{k-1},R_{k+1},\dots,R_m) \qquad k = 1,\dots,m$$

are nonsingular. The kth block element of Y is:

$$Y_k = \{R_k^{m-1} - (R_1^{m-1},\dots,R_{k-1}^{m-1},R_{k+1}^{m-1},\dots,R_m^{m-1}).V_k^{-1}.(I,R_k,R_k^2,\dots R_k^{m-1})\}^{-1}$$

We can remark that this element is the inverse of a right evaluation of a monic matrix polynomial of degree m-1, $B_k(\lambda)$.

$$B_k(\lambda) = \lambda^{m-1} I + B_{k1}\lambda^{m-2} + \dots + B_{k,m-1}$$

$$(B_{k,m-1},\dots,B_{k1}) = -(R_1^{m-1},\dots,R_{k-1}^{m-1},R_{k+1}^{m-1},\dots,R_m^{m-1}).V_k^{-1}$$

So, finally: $Y_k = [B_{kR}(R_k)]^{-1}$

Another standard triple is defined when $A(\lambda)$ has a complete factorization:

$$A(\lambda)=(\lambda I-Q_m)(\lambda I-Q_{m-1})\ldots(\lambda I-Q_1) \tag{2.36}$$

and it is the following one:

$$X_1=[I,0,\ldots,0] \qquad Y_1=\begin{bmatrix}0\\ .\\ .\\ 0\\ I\end{bmatrix}$$

$$T_1=\begin{bmatrix}Q_1 & I & 0 & \ldots & 0\\ 0 & Q_2 & I & \ldots & 0\\ . & . & . & \ldots & .\\ 0 & 0 & 0 & \ldots & I\\ 0 & 0 & 0 & \ldots & Q_m\end{bmatrix} \tag{2.37}$$

This result can be established either by repeated application of theorem 2.8 or by using the concept of linearization [10,14,26]. This is what we do in the next theorem.

Theorem 2.11:

$\lambda I-T_1$ is a linearization of $A(\lambda)$,

i.e.

$$\lambda I-T_1 \sim \begin{bmatrix}A(\lambda) & 0\\ 0 & I\end{bmatrix}$$

.

Proof:

We define the following two matrices:

$$F(\lambda) = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 \\ -(\lambda I - Q_1) & I & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & -(\lambda I - Q_m) & I \end{bmatrix}$$

and

$$E(\lambda) = \begin{bmatrix} B_{m-1}(\lambda) & B_{m-2}(\lambda) & \dots & B_1(\lambda) & B_0(\lambda) \\ -I & 0 & \dots & 0 & 0 \\ 0 & -I & \dots & 0 & \\ \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & \dots & -I & 0 \end{bmatrix}$$

$B_0(\lambda) = I$

$B_k(\lambda) = B_{k-1}(\lambda)(\lambda I - Q_{m-k+1}) \qquad k = 1,\dots,m-1$

We see that $det E(\lambda) = \pm 1$ and $det F(\lambda) = 1$. So E and F are unimodular matrix polynomials. Furthermore, by simply computing the product, we have:

$$E(\lambda)(\lambda I - T_1) = \begin{bmatrix} (\lambda I - Q_m)\dots(\lambda I - Q_1) & 0 \\ 0 & I \end{bmatrix} . F(\lambda)$$

(Q.e.d)

We now show that X_1, T_1, Y_1 is a standard triple by using the resolvent form of $A(\lambda)$ equation (2.12).

$$[A(\lambda)]^{-1} = X_1(\lambda I - T_1)^{-1} Y_1$$

Theorem 2.12:

If $A(\lambda)$ admits the complete factorization (2.36), then X_1, T_1, Y_1 is a standard triple for A.

Proof:

$$\begin{bmatrix} [A(\lambda)]^{-1} & 0 \\ 0 & I \end{bmatrix} = F(\lambda)(\lambda I - T_1)^{-1} E^{-1}(\lambda) \tag{2.38}$$

the first r columns of $E^{-1}(\lambda)$ are:

$$Y_1 = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ I \end{bmatrix} \quad (B_0(\lambda) = I)$$

and we have: $X_1 F(\lambda) = X_1$. Let us multiply (2.38) by X_1 on the left and X_1^T [6] on the right. This gives:

$$A^{-1}(\lambda) = X_1 F(\lambda)(\lambda I - T_1)^{-1} E^{-1}(\lambda) X_1^T$$

$$\Rightarrow \qquad A^{-1}(\lambda) = X_1 (\lambda I - T_1)^{-1} Y_1$$

This implies that X_1, T_1, Y_1 is a standard triple.

(Q.e.d)

Theorem 2.12 and the standard triple given by (2.37) will be useful when we will present an algorithm that can factorize $A(\lambda)$ completely.

6 X^T is the transpose of X

Chapter 3

Global Methods

3.1 Some Definitions.

In this chapter, we are going to present some existing algorithms that can factorize a linear term from a given matrix polynomial. We will see later that the Q.D. algorithm can be viewed as a generalization of these methods. The methods of interest are : Bernoulli's method [8,10] and Traub's method [10]. However, we have first to define what a global method is and what a local method is. Global methods are defined by opposition to local ones.

Definition 3.1:

A numerical method for solving a given problem is said to be local if it is based on local (simpler) model of the problem around the solution.

From the definition, we can see that in order to use a local method, one has to provide an initial approximation of the solution. This initial approximation can be provided by a global method. As we will see later, local methods are fast converging while global ones are quite slow. This implies that a good strategy is to start solving the problem by using a global method and then refine the solution by a local method.

The convergence of the global methods that will be presented in this chapter is based on the following relation of order (partial) between square matrices.[8,10]

Definition 3.2:

A square matrix A is said to dominate a square matrix B (not necessarily of the same size) if all the eigenvalues of A are greater, in modulus, than those of B.

As a notation, we will write $A > B$. This definition is important because of the following lemma. In the remainder of the thesis, we will use matrix norms for our convergence proofs. Since we are working in a finite dimensional space, all matrix norms are equivalent. The only specific property that we require is the consistency property: $|AB| \leq |A||B|$.

Lemma 3.1:

Let A and B be square matrices such that $A > B$ *then A is nonsingular and*

$$\lim_{n \to \infty} |B^n||A^{-n}| = 0 \tag{3.1}$$

Proof: see ref. [10]

The same lemma (in a slightly different form) is proved in Dennis et Al. ref.[8]. In some of the results that we will present, we need to estimate the convergence rate. In that aspect, the following result is interesting. Let the largest modulus eigenvalue of the matrix A be λ_A, then for any matrix norm [36,37], we have:

$$\lim_{k\to\infty}|A^k|^{\frac{1}{k}} = |\lambda_A| \tag{3.2}$$

In particular, for any positive number $\rho > \epsilon > 0$, we have the following result:

$$a_k = O(\rho^{-k}) \Rightarrow a_k|A^k| = O\left(\left|\frac{\lambda_A}{\rho - \epsilon}\right|^k\right) \tag{3.3}$$

We will extend definition 3.2 to matrix polynomials by saying that the matrix polynomial $A_1(\lambda)$ dominates $A_2(\lambda)$ if $T_1 > T_2$, T_k being a linearization of A_k.

3.2 The Homogeneous Difference Equation.

The numerical methods that will follow are based on the exponential nature of the solution of a homogeneous constant coefficient matrix difference equation. This matrix difference equation is associated with the matrix polynomial that we want to factorize. In fact, we can associate two difference equations with a particular matrix polynomial $A(\lambda)$: a right matrix difference equation and a left one.

To:

$$A(\lambda) = \lambda^m I + A_1\lambda^{m-1} + \dots + A_m \tag{3.4}$$

we associate the following right matrix difference equation:

$$U_k + A_1 U_{k-1} + \dots + A_m U_{k-m} = 0 \tag{3.5}$$

$$U_j \in C^{r \times r} \qquad j = 0, 1, \dots$$

and the following left matrix difference equation:

$$V_k + V_{k-1} A_1 + \dots + V_{k-m} A_m = 0 \tag{3.6}$$

$$V_j \in C^{r \times r} \qquad j = 0, 1, \dots$$

We should point that the left difference equation (3.6) can be written as a right difference equation associated with $[A(\lambda)]^T$ and the right difference equation (3.5) can be written as a left difference equation associated with $[A(\lambda)]^T$. So, for the rest of the presentation, result will be presented only for right difference equation. The result for the left one can be obtained by transposition.

The general solution of (3.5) is derived in ref. [10] and in ref. [8], we can find the solution for the complete set of solvent case (see definition 2.12). This solution is presented as a function of the standard triple in the following theorem.

Theorem 3.1:

Given a matrix polynomial $A(\lambda)$ having (X,T,Y) as a standard triple, the general solution of (3.5) is:

$$U_k = X T^k C \tag{3.7}$$

$$C \in C^{mr \times r}$$

and the general solution of (3.6) is:

$$V_k = DT^k Y \tag{3.8}$$

$$D \in C^{r \times mr}$$

Proof:

Using the definition of a standard pair [10,11,12], the following identity is satisfied:

$$XT^m + A_1 XT^{m-1} + \ldots + A_m X = 0 \tag{3.9}$$

If we multiply (3.9) on the right by $T^{k-m}C$, $C \in C^{mr \times r}$, we obtain:

$$XT^k C + A_1 XT^{k-1} C + \ldots + A_m XT^{k-m} C = 0$$

and thus $XT^k C$ verifies the equation (3.5).

The proof of (3.8) can be derived by using the fact that the standard triple of $[A(\lambda)]^T$ is (Y^T, T^T, X^T).

(Q.e.d).

Corollary 3.1:

The solution of (3.5) corresponding to the initial conditions:

$$U_0 = U_1 = \ldots = U_{m-2} = 0 \quad U_{m-1} = I \tag{3.10}$$

is:

$$U_k = XT^k Y \tag{3.11}$$

Proof:

Using (3.7), we can write the following set of equations:

$$\begin{aligned} XC &= 0 \\ XTC &= 0 \\ &\cdot \\ &\cdot \\ XT^{m-2}C &= 0 \\ XT^{m-1}C &= I \end{aligned}$$

giving:

$$\begin{bmatrix} X \\ XT \\ \cdot \\ \cdot \\ XT^{m-1} \end{bmatrix} C = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ I \end{bmatrix}$$

which is the definition of Y.

(Q.e.d).

By using transposition, we obtain the following result.

Corollary 3.2:

The solution of (3.6) corresponding to the initial conditions:

$$V_0 = V_1 = \ldots = V_{m-2} = 0 \quad V_{m-1} = I$$

is:

$$V_k = XT^kY$$

We remark that for this particular set of initial conditions, the right and left difference equations produce the same result.

The solution of the difference equation becomes particularly simple if we have a complete set of solvents.

Corollary 3.3:

If the matrix polynomial $A(\lambda)$ has a complete set of solvents then the solution of (3.5) subject to the initial conditions (3.10) is:

$$U_k = \sum_{j=1}^{m} R_j^k Y_j \tag{3.12}$$

$$Y = \begin{bmatrix} Y_1 \\ \cdot \\ \cdot \\ Y_m \end{bmatrix}$$

Proof:

From theorem 2.10, relations (2.34) give us the standard triple:

$$X = [I, I, \ldots, I] \quad T = \begin{bmatrix} R_1 & 0 & \ldots & 0 \\ 0 & R_2 & \ldots & 0 \\ \cdot & \cdot & \ldots & \cdot \\ 0 & 0 & \ldots & R_m \end{bmatrix} \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \cdot \\ Y_m \end{bmatrix}$$

Replacing in (3.11) produces (3.12).

(Q.e.d)

The result (3.12) could have been obtained directly from the partial fraction expansion (2.35) by using the inverse z-transform. Of course, in this case, we are considering that λ is the forward shift operator z.

3.3 Bernoulli's Method.

In this section, we are going to present a global algorithm, the Bernoulli's iteration, that is based on the form of the solution of the difference equation (exponential) and on lemma 3.1. Just as in the scalar case [15,17], the matrix Bernoulli's method is based on the "ratio" of two successive iterates of the difference equation (3.5) (or (3.6)). In the literature, we can find two different statements for the convergence of the matrix Bernoulli's iteration. In Gohberg et Al. ref.[10], it is stated in terms of a general standard triple while in Dennis et Al. ref [8], it is stated for the particular case of a complete set of solvents. The next theorem, similar to the one in Gohberg et Al. is thus the more general one.

Theorem 3.2:

Let $A(\lambda)$ be a monic matrix polynomial of degree m and order r. Assume that $A(\lambda)$ has a dominant right solvent R and a dominant left solvent L. Let $U_k, k=0,1,\ldots$ be the solution of (3.5) subject to the initial conditions (3.10).

Then U_k is not singular for k large enough, and:

$$\lim_{k\to\infty} U_{k+1}U_k^{-1} = R \qquad (3.13)$$

$$\lim_{k\to\infty} U_k^{-1}U_{k+1} = L \qquad (3.14)$$

Proof:

The existence of a dominant right solvent and a dominant left solvent implies (see theorem 2.9) that $A(\lambda)$ has the following Jordan triple:

$$X=[X_1 \;\; X_2] \quad J=\begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \quad Y=\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \tag{3.15}$$

where X_1, J_1 and Y_1 are r x r matrices. X_1 and Y_1 are nonsingular and from the relations (2.30) and (2.31), we can write:

$$R=X_1 J_1 X_1^{-1} \quad L=Y_1^{-1} J_1 Y_1 \tag{3.16}$$

We now use the result of corollary 3.1:

$$\begin{aligned} U_k &= XJ^kY = X_1J_1^kY_1 + X_2J_2^kY_2 \\ &= [X_1J_1^kX_1^{-1} + (X_2J_2^kY_2)Y_1^{-1}X_1^{-1}]X_1Y_1 \end{aligned}$$

let $M=X_1Y_1$ and $E_k=X_2J_2^kY_2$, then

$$\begin{aligned} U_k &= (R^k+E_kM^{-1})M \\ U_k &= (I+E_kM^{-1}R^{-k})R^kM \end{aligned} \tag{3.17}$$

The same factorization can be done from the left, in which case we obtain:

$$U_k = ML^k(I+L^{-k}M^{-1}E_k) \tag{3.18}$$

As a general remark, from lemma 3.1, we know that R and L are nonsingular.

For k large enough, we have:

$$|E_k M^{-1} R^{-k}| < |E_k||R^{-k}||M^{-1}|$$

and the right handside of the above inequation converges to zero by lemma 3.1. This implies that U_k is nonsingular for k large enough. The same argument can be applied to equation (3.18).

Finally:

$$U_{k+1} U_k^{-1} = (I + E_{k+1} M^{-1} R^{-k-1}) R^{k+1} M M^{-1} R^k (I + E_k M^{-1} R^{-k})^{-1}$$

So:

$$\lim_{k \to \infty} U_{k+1} U_k^{-1} = R$$

and from (3.18), we obtain:

$$\lim_{k \to \infty} U_k^{-1} U_{k+1} = L$$

(Q.e.d).

It is interesting to look at the case of a complete set of solvents. In this particular case, the convergence is stated in terms of block Vandermonde matrices [8].

Theorem 3.3:

Let $A(\lambda)$ be a monic matrix polynomial of degree m and order r such that:

(i) it has a complete set of solvents $R_1, R_2, \ldots, R_m$,

(ii) R_1 is a dominant right solvent,

(iii) $V(R_1, \ldots, R_m)$ and $V(R_2, \ldots, R_m)$ are nonsingular,

then:

$$\lim_{k\to\infty} U_{k+1}U_k^{-1} = R_1$$

Proof:

From corollary 3.3, the solution of the difference equation (3.5) is:

$$U_k = \sum_{j=1}^{m} R_j^k Y_j$$

Looking back at relations (2.35), we see that Y_1 is not singular if V(R2, ... , Rm) is nonsingular. So:

$$\begin{aligned} U_k &= R_1^k Y_1 + \sum_{j=2}^{m} R_j^k Y_j \\ U_k &= \left(I + \sum_{j=2}^{m} R_j^k Y_j Y_1^{-1} R_1^{-k}\right) R_1^k Y_1 \\ U_k &= (I + H_k) R_1^k Y_1 \end{aligned}$$

and $|H_k|$ converge toward zero. Thus, for large enough k, U_k is nonsingular and we can write:

$$U_{k+1}U_k^{-1} = (I + H_{k+1}) R_1^{k+1} Y_1 Y_1^{-1} R_1^{-k} (I + H_k)^{-1}$$

So, finally:

$$\lim_{k\to\infty} U_{k+1}U_k^{-1} = R_1$$

(Q.e.d)

There are some general remarks that we can make about the conditions for convergence of the Bernoulli method. In the above theorem, V(R2, ... , Rm) nonsingular implies the

$$R = X_1 J_1 X_1^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}$$

We can remark that Y_1 is singular. This precludes the existence of a dominant left solvent. In this case Bernoulli's iteration breaks down. The solution of (3.5) using $U_0 = U_1 = 0$, $U_2 = I$ as initial conditions is:

$$U_3 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \quad U_4 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad U_5 = U_3 \quad U_6 = U_4$$

So, we see that the sequence of matrices, solution of the difference equation associated with the matrix polynomial, is a sequence of singular matrices.

3.4 Traub's Algorithm.

In Dennis et Al. ref. [8], another algorithm is presented. This method is a generalization of Traub's algorithm [38] to matrix polynomials. The algorithm is presented without a proof of convergence in this section. However, since the Q.D. algorithm of chapter 4 can be seen as a generalization of both Traub's and Bernoulli's algorithm, the interested reader can adjust the proof that we will present in chapter 4.

The Algorithm:

Let $A(\lambda)$ be a monic matrix polynomial of degree m and order r. We define the following sequence of matrix polynomials of degree m-1 and order r by:

$$B_{n+1}(\lambda) = B_n(\lambda)\lambda - B_0^{(n)} A(\lambda) \tag{3.20}$$

This implies that this algorithm has a built in deflation process. And one can think of repeating the same iteration starting from the limit of the iteration (3.20).

3.5 The Block Power Method.

Using block matrices, we can show that both Bernoulli´ iteration and Traub´s method can be seen as an application of the power method. For this purpose, we are going to use the standard triple (P_1, C_1, Q_1) defined in (2.13).

With the square matrix C1, we can in fact associate two iterations: a right block power method and a left block power method [8]. The right block power method is considered in [39].

The Bernoulli´s iteration (3.5) can be written as:

$$\begin{bmatrix} U_{t-m+2} \\ . \\ . \\ U_t \\ U_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & I & \cdots & 0 \\ . & . & \cdots & . \\ 0 & 0 & \cdots & I \\ -A_m & -A_{m-1} & \cdots & -A_1 \end{bmatrix} \begin{bmatrix} U_{t-m+1} \\ . \\ . \\ U_{t-1} \\ U_t \end{bmatrix} \tag{3.25}$$

Iteration (3.25) is a right block eigenvector powering iteration if we define $\vec{\mathbf{U}}_t$ as:

$$\vec{\mathbf{U}}_t = \begin{bmatrix} U_{t-m+1} \\ . \\ . \\ U_{t-1} \\ U_t \end{bmatrix} \tag{3.26}$$

Theorem 2.6 shows that the fixed point of the above defined iteration is the following invariant subspace:

$$Im\begin{bmatrix} I \\ R \\ R^2 \\ . \\ R^{m-1} \end{bmatrix}$$

and this property is used in reference [39].

Using now block row vectors, we can define another iteration:

$$\vec{\mathbf{B}}_{n+1} = \vec{\mathbf{B}}_n C_1 \tag{3.27}$$

where:

$$\vec{\mathbf{B}}_n = [B_{m-1}^{(n)}, \ldots, B_0^{(n)}] \tag{3.28}$$

It is not hard to show that (3.27) is equivalent to Traub's iteration (3.20) (see ref. [8]). And thus the block power method can be seen as a unifying concept between Bernoulli's and Traub's iteration. In the next chapter, we will generalize (3.27) to obtain a method that can produce a complete factorization of the monic matrix polynomial $A(\lambda)$.

Chapter 4

The Quotient-Difference Algorithm

4.1 Introduction.

In this chapter, we are going to present a new algorithm for factorizing matrix polynomials: the quotient-difference algorithm (Q.D.). The proposed algorithm is a generalization of the scalar Q.D. [17,18] algorithm to matrix polynomials. The use of the Q.D. algorithm for such purpose has been suggested by Hariche in reference [14]. The scalar Q.D. algorithm is just one of the many global methods that are commonly used for finding the roots of a scalar polynomial. Another global method that is quite popular is the Graeffe's iteration [2,19]. However, it seems that it is quite dependent on the fact the scalar polynomial coefficients commute. So, we do not see how we can generalize it to matrix polynomials.

A major problem that we encountered when we wanted to generalize the convergence (and existence) proofs of the scalar method to matrix polynomials is the fact that those proofs are given in terms of determinants [15,16,18]. The proofs that we present in this chapter are based on a generalization of Traub's iteration as presented in chapter 3. It is essentially the same approach as Stewart's [37], but applied to matrix polynomials.

We have seen in the preceding chapter that we have two Bernoulli's iterations due to the lack of commutativity in the algebra of square matrices. Likewise, there exist two matrix Q.D. algorithms: the right Q.D. algorithm and the left Q.D. algorithm. The subsequent presentation will be given only for the left Q.D. algorithm. We can obtain the same results for the right Q.D. algorithm by transposition. In the next chapter, we will be using block matrix methods to provide a convergence proof for the right Q.D. algorithm.

4.2 The Algorithm.

The Quotient-Difference scheme for matrix polynomials can be defined just like the scalar one [15,17,18] by a set of recurrence equations. The algorithm consists on building a table that we call the Q.D. tableau (in this chapter, we define the left Q.D. tableau, the right one can easily be defined by transposition).

The left Q.D. scheme is generated via the following relations (the "rhombus" rules):

$$
\begin{gathered}
Q_k^{(n+1)} + E_{k-1}^{(n+1)} = Q_k^{(n)} + E_k^{(n)} \\
Q_k^{(n+1)} E_k^{(n+1)} = E_k^{(n)} Q_{k+1}^{(n)} \qquad (4.1) \\
E_0^{(n)} = E_m^{(n)} = 0 \\
k = 1, \ldots, m-1 \ ; \ n = 1, 2, \ldots
\end{gathered}
$$

These rules define the following table (the Q.D. tableau):

$$
\begin{array}{cccccc}
 & Q_1^{(0)} & & & & \\
0 & & E_1^{(0)} & & & \\
 & Q_1^{(1)} & & Q_2^{(0)} & & \\
0 & & E_1^{(1)} & & E_2^{(0)} & \\
 & Q_1^{(2)} & & Q_2^{(1)} & & \cdot \\
0 & & E_1^{(2)} & & E_2^{(1)} & \\
 & Q_1^{(3)} & & Q_2^{(2)} & & \\
\cdot & & \cdot & & \cdot &
\end{array}
\qquad (4.2)
$$

The Q.D. tableau can be generated by columns (we need the first two columns as initial conditions) or by rows. In this chapter, we will study the column generation of the Q.D. scheme (from Bernoulli's iteration) and in the next chapter, we will show that it is possible to generate the Q.D. tableau by rows. This alternate generation of the scheme is more stable numerically.

4.3 Applicable Class of Matrix Polynomials.

Because it simplifies notation, we are going to present the Q.D. algorithm for comonic matrix polynomials. The reader should recall that if $A(\lambda)$ is a monic matrix polynomial of degree m and order r, then $F(z)=z^m A(z^{-1})$ is a comonic matrix polynomial of same degree with inverse spectrum and same latent vectors. This of course imposes the restriction that zero is not a latent root of $A(\lambda)$. In this chapter, we are going to consider comonic matrix polynomials that have a nonsingular leading coefficient.

$$F(z)=I+A_1z+A_2z^2+\ldots+A_mz^m \qquad (4.3)$$
$$A_k\in C^{r\times r} \; ; \; k=1,2,\ldots,m \; ; \; z\in C$$

In order to have more concise statements in the convergence theorems, let us define the following property (see theorem 2.9).

Definition 4.1:

We say that F(z) has the property Γ_k *if it possesses a right and left* Γ*-spectral factor of degree k.*

In other words, if $F(z)=\Pi_k(z)F_k(z)$ where Π_k is a comonic matrix polynomial of degree k, Then $F(z)=\overline{F}_k(z)\overline{\Pi}_k(z)$ along with:

$$deg\Pi_k=deg\overline{\Pi}_k \; ; \; degF_k=deg\overline{F}_k$$
$$\sigma(\Pi_k)=\sigma(\overline{\Pi}_k) \; ; \; \sigma(F_k)=\sigma(\overline{F}_k)$$
$$\sigma(\Pi_k)\cap\sigma(F_k)=\emptyset \; ; \; \sigma(\overline{\Pi}_k)\cap\sigma(\overline{F}_k)=\emptyset$$

In particular, if F(z) has a right solvent R, it will also have a left solvent L that possesses the same spectrum.

4.4 The Linear Diophantine Equation.

If we have a comonic matrix polynomial F(z) of degree m and order r that possesses the property Γ_k, then any matrix polynomial of degree less than m can be expressed in terms of its factors. In order to show this property we need the following fact.

Theorem 4.1:

The following equation

$$X(z)A(z)+Y(z)B(z)=C(z) \tag{4.4}$$

(X, Y, A, B and C are matrix polynomials) has a solution if and only if the greatest common right divisor of A and B is a right divisor of C.

Proof: see Kucera ref.[24].

Corollary 4.2:

If A(z) and B(z) are right coprime matrix polynomials, then (4.4) has always a solution.

In our work, we consider matrix polynomials that have the property Γ_k. Then the factors of F(z) have disjoint spectra by definition. In this case and using the notation of definition 4.1, we can state the following proposition.

Proposition 4.3:

Any matrix polynomial G(z) of degree less than m and order r can always be written as:

$$G(z)=X(z)\overline{\Pi}_k(z)+Y(z)F_k(z) \tag{4.5}$$

$$\deg Y<k \ ; \ \deg X<m-k$$

Proof:

(4.5) is a linear diophantine equation and since $\sigma(\overline{\Pi}_k)\cap\sigma(F_k)=\emptyset$, then it has always a solution. The general solution of (4.4) is [24]:

$$X(z)=X_0(z)-T(z)B_1(z)$$
$$Y(z)=Y_0(z)+T(z)A_1(z)$$

where T(z) is an arbitrary matrix polynomial, $X_0(z)$ and $Y_0(z)$ is a particular solution of (4.5) while $A_1(z)$ and $B_1(z)$ are coprime matrix polynomials such that:

$$B_1(z)\overline{\Pi}_k(z)=A_1(z)F_k(z)$$

Using the fact that F(z) has the property Γ_k, $F(z)=\Pi_k(z)F_k(z)=\overline{F}_k(z)=\overline{\Pi}_k(z)$, so $A_1=\Pi_k$ and $B_1=\overline{F}_k$. We divide Y_0 by Π_k giving:

$$Y_0(z)=U(z)\Pi_k(z)+V(z) \quad ; \quad degV<k$$
$$Y(z)=U(z)\Pi_k(z)+V(z)+T(z)\Pi_k(z)$$

Using T(z) = -U(z) (T being arbitrary), we obtain:

$$Y(z)=V(z)$$
$$X(z)=X_0(z)+U(z)\overline{F}_k(z)$$

So:

$$G(z)=Y(z)F_k(z)+X(z)\overline{\Pi}_k(z)$$

and since $:deg F_k < m-k,$ $deg Y < k$ and $deg G < m,$ then $deg X(z)\overline{\Pi}_k(z) < m$. So $deg X < m-k$.

(Q.e.d.).

The above proposition has the consequence that proper matrix rational fractions can be expanded in incomplete partial fractions. Let F(z) have the property Γ_k, and C(z) be the following matrix rational fraction, $C(z)=G(z)F^{-1}(z)$, G(z) being a matrix polynomial of degree less than m. In this case, proposition 4.3 applies to G(z) and we can write:

$$F(z)=\Pi_k(z)F_k(z)=\overline{F}_k(z)\overline{\Pi}_k(z) \qquad (\Gamma_k)$$
$$G(z)=U(z)F_k(z)+V(z)\overline{\Pi}_k(z)$$
$$deg U < m-k \quad ; \quad deg V < k$$

So, the rational matrix C(z) becomes:

$$C(z)=U(z)\Pi_k^{-1}(z)+V(z)\overline{F}_k^{-1}(z) \tag{4.6}$$

Each term in the above sum is a proper matrix rational fraction.

4.5 Power Series.

In this section, we are going to present some useful bounds on the elements of a Bernoulli's iteration. This is achieved by identifying the solution of a difference equation with the coefficients of a formal power series.

Proposition 4.4:

Let $V(z)$ and $F(z)$ be matrix polynomials with:

$$V(z)=V_0+V_1z+\ldots+V_{m-1}z^{m-1}$$
$$F(z)=A_0+A_1z+\ldots+A_mz^m$$

along with A_0 nonsingular. $C(z)=V(z)F^{-1}(z)$ can always be developed into the following power series:

$$C(z)=C_0+C_1z+C_2z^2+\ldots \tag{4.7}$$

Proof:

let $C(z)=\sum_{i=0}^{\infty}C_iz^i$, we have:

$$\begin{aligned} V_0 &= C_0A_0 \\ V_1 &= C_1A_0 + C_0A_1 \\ &\;\;\vdots \\ V_{m-1} &= C_{m-1}A_0 + C_{m-2}A_1 + \ldots + C_0A_{m-1} \\ 0 &= C_kA_0+C_{k-1}A_1+\ldots+C_{k-m}A_m \;\; ; \;\; k\geq m \end{aligned} \tag{4.8}$$

(4.8) is a recursion that can always be solved since A_0 is nonsingular.

(Q.e.d.).

The power series defined in (4.7) converges under the following circumstances.

Proposition 4.5:

Let λ *be the latent root of smallest modulus of F(z) (*λ *is different from zero since* A_0 *is nonsingular). The power series (4.7) converges for all z such that* $|z| \leq \rho < \lambda$ *and the coefficients* C_n *are given by:*

$$C_n = \frac{1}{2\pi j}\oint_{\Gamma} C(z)z^{-n-1}dz \qquad (4.9)$$

Γ *is a circle of radius* ρ.

Proof:

$$F^{-1}(z) = \frac{1}{\det F(z)} adj[F(z)]$$

$$\Rightarrow C(z) = V(z) adj[F(z)] \frac{1}{\det F(z)}$$

So the elements of C(z) are scalar rational fractions and the denominator of each element is a factor of det F(z). So all the elements of C(z) are analytic in Γ and we can develop each one in Taylor series.

(Q.e.d.).

Corollary 4.6:

Under the conditions of proposition (4.5), the sequence of coefficients C_n *is bounded as:*

$$|C_n| \leq M\rho^{-n} \qquad (4.10)$$

Proof:

Let $M = \max|C(z)|$: $z \in \Gamma$ then using the Riemman definition of the integral:

$$|C_n| \leq \frac{1}{2\pi}\oint_{\Gamma}|C(z)||z^{-n}|\frac{ds}{|z|} \leq M\rho^{-n}$$

(Q.e.d.).

If now F(z) possesses the property Γ_k, we can factorize it and under this condition we can have a cancellation of factors between the "numerator" and "denominator" of C(z). Keeping the same notation as in definition 4.1, we can state:

Proposition 4.7:

Let $F(z) = \Pi_k(z)F_k(z) = \bar{F}_k(z)\bar{\Pi}_k(z)$ *(property* Γ_k*) and* $C(z) = Q(z)F^{-1}(z)$ *along with* $Q(z) = V(z)\bar{\Pi}_k(z)$*, then C(z) is analytic for all z such that* $|z| \leq \rho < |\lambda_k|$ *where* λ_k *is the latent root of smallest modulus of* $F_k(z)$.

Proof:

$$F(z) = \Pi_k(z)F_k(z) = \bar{F}_k(z)\bar{\Pi}_k(z)$$

$$\Rightarrow \quad C(z) = V(z)\bar{\Pi}_k(z)\bar{\Pi}_k^{-1}(z)\bar{F}_k^{-1}(z)$$

$$\Rightarrow \quad C(z) = V(z)\bar{F}_k^{-1}(z)$$

(Q.e.d.)

4.6 A Generalization of Linear Independence.

The following definitions will be used in our work. In this section, we show that matrix polynomials of a given degree can be seen as elements of a left (right) module over the non-commutative ring of square matrices.

Definition 4.2:

Let $\vec{B}=(B_0,B_1,\ldots,B_{m-1})$ *be a block row vector (a rxmr matrix) where each block is an rxr matrix. The set* $\{\vec{B}_1,\vec{B}_2,\ldots,\vec{B}_k\}$ *is said to form a linearly independent set of block row vectors if and only if the following matrix is of rank kr:*

$$\mathbf{B}=\begin{bmatrix}\vec{B}_1\\ \vec{B}_2\\ \cdot\\ \cdot\\ \vec{B}_k\end{bmatrix}$$

An easily proved proposition follows:

Proposition 4.8:

Let $\{\vec{B}_1,\vec{B}_2,\ldots,\vec{B}_k\}$ *be a set of linearly independent block row vectors, then:*

$$\Delta_1\vec{B}_1+\Delta_2\vec{B}_2+\ldots+\Delta_k\vec{B}_k=0 \iff \Delta_1=\Delta_2=\ldots=\Delta_k=0$$

$$\Delta_n\in C^{r\times r} \ ; \ n=1,2,\ldots,k$$

The interest of proposition 4.8 is that it allows us to write any matrix polynomial of degree less than m as a linear combination of m linearly independent matrix polynomials. Let us introduce the following notation.

Notation:

Let $B(z)=B_0+B_1z+\ldots+B_{m-1}z^{m-1}$ be a matrix polynomial. To B(z), we associate the block row vector formed by it coefficients $\vec{B}=(B_0,B_1,\ldots,B_{m-1})$.

We have the following relation between B(z) and $\vec{B}$:

$$B(z) = \vec{B}\begin{bmatrix} I \\ zI \\ \cdot \\ \cdot \\ z^{m-1}I \end{bmatrix}$$

$$z\in C \;;\; I=r\times r \text{ identity}$$

The above notation allows us to define linear independence for matrix polynomials.

Definition 4.3:

The matrix polynomials $B_1(z),\ldots,B_k(z)$ are said to be linearly independent if and only if their associated block row vectors form a linearly independent set.

Proposition 4.9:

Let $B_1(z), B_2(z), \ldots, B_m(z)$ be a set of m linearly independent matrix polynomials of degree $m-1$ and order r. Any matrix polynomial $G(z)$ of degree $m-1$ or less and of order r can be expressed in a unique manner as:

$$G(z) = \sum_{k=1}^{m} \Delta_k B_k(z) \tag{4.11}$$

$$\Delta_k \in C^{r \times r}$$

Proof:

Let:

$$\mathbf{B} = \begin{bmatrix} \vec{B}_1 \\ \vec{B}_2 \\ . \\ . \\ \vec{B}_m \end{bmatrix}$$

Because of the hypothesis of the proposition, $\mathbf{B}$ is a square nonsingular matrix. Thus the equation $\vec{G} = \vec{\Delta}\mathbf{B}$ has a unique solution.

(Q.e.d.).

A particular set of linearly independent matrix polynomials is defined in Dennis et Al. ref.[7].

4.7 A Generalization of the Power Method.

In this section, we are going to generalize Traub's iteration using a Treppen like iteration [41]. Our presentation of the Q.D. algorithm is essentially a generalization of Stewart's paper on companion operators [37] to matrix polynomials. Let us consider the following matrix polynomial:

$$F(z) = z^m A(z^{-1}) = I + A_1 z + A_2 z^2 + \dots + A_m z^m \tag{4.12}$$

A_k are rxr matrices and A_m is nonsingular.

To F(z) we can associate the following block companion matrix:

$$\mathbf{C}_F = \begin{bmatrix} 0 & I & 0 & \dots & 0 & 0 \\ 0 & 0 & I & \dots & 0 & 0 \\ . & . & \dots & . & . & \\ 0 & 0 & 0 & \dots & 0 & I \\ -A_m^{-1} & -A_m^{-1}A_1 & -A_m^{-1}A_2 & \dots & -A_m^{-1}A_{m-2} & -A_m^{-1}A_{m-1} \end{bmatrix} \tag{4.13}$$

Since A_m is nonsingular and (4.12) is comonic, this matrix is nonsingular and has the following inverse:

$$\mathbf{F} = \mathbf{C}_F^{-1} = \begin{bmatrix} -A_1 & -A_2 & \dots & -A_{m-1} & -A_m \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ . & . & \dots & . & . \\ 0 & 0 & \dots & I & 0 \end{bmatrix} \tag{4.14}$$

It is clear that the spectrum of **F** is the same as the one of $A(\lambda)$ and is of course composed of the inverses of the latent roots of F(z).

To a matrix polynomial $B(z) = B_0 + B_1 z + \ldots + B_{m-1} z^{m-1}$, we associate the block row vector $\vec{B} = (B_0, B_1, \ldots, B_{m-1})$ as in section 4.6.

We are going to apply the block power method to the block matrix **F**. The iteration can be written more compactly as an iteration between matrix polynomials.

Proposition 4.10:

If $\vec{B}^* = \vec{B}\mathbf{F}$ *then*

$$B^*(z) = z^{-1}[B(z) - B(0)F(z)] \tag{4.15}$$

Proof:

In the equation $(B_0^*, \ldots, B_{m-1}^*) = (B_0, \ldots, B_{m-1})\mathbf{F}$, we use the fact that $B_0 = B(0)$. We obtain:

$$\begin{aligned} -B_0 I + B_0 &= 0 \\ -B_0 A_1 + B_1 &= B_0^* \\ -B_0 A_2 + B_2 &= B_1^* \\ &\cdots \\ -B_0 A_m \qquad &= B_{m-1}^* \end{aligned}$$

We multiply each equation respectively by $z^0, z^1, z^2, \ldots, z^m$ and we add them. The result is equation (4.15).

(Q.e.d.).

Notation:

In the remainder of the thesis, we will sometimes use the notation $\{\vec{B}\mathbf{F}\}(z)$ instead of the notation of equation (4.15).

Remark:

Equation (4.15) is the same as Traub's iteration (3.27). The use of **F** instead of C_1 comes from the fact that we are using comonic matrix polynomials.

In the rest of the chapter, we will consider matrix polynomials that have the property Γ_k according to definition 4.1. So:

$$F(z)=\Pi_k(z)F_k(z)=\overline{F}_k(z)\overline{\Pi}_k(z) \tag{4.16}$$

where $\Pi_k(z)=I+\Pi_{k,1}z+\ldots+\Pi_{k,k}z^k$ and $F_k(z)=I+A_{k,1}z+\ldots+A_{k,m-k}z^{m-k}$.

Let λ_k be the largest modulus latent root of $\Pi_k(z)$ and λ_{k+1} be the smallest modulus latent root of $F_k(z)$. We assume that $|\lambda_k|<|\lambda_{k+1}|$. In this case $F_k(z)$ dominates $\Pi_k(z)$. We have also:

$$\sigma(F_k(z))\cap\sigma(\Pi_k(z))=\varnothing$$

$$\sigma(F_k(z))\cup\sigma(\Pi_k(z))=\sigma(F(z))=\sigma(\mathbf{C}_F)=\sigma(\mathbf{F}^{-1})$$

From proposition 4.3, any matrix polynomial G(z) of degree less than m can be written as:

$$G(z)=U(z)F_k(z)+V(z)\overline{\Pi}_k(z)$$

along with $\deg U(z)<k$ and $\deg V(z)<m-k$.

For this particular factorization of F(z), we can define two classes of matrix polynomials:

$$\mathcal{P}_k = \{P(z) \mid P(z) = U(z)F_k(z) ; \deg U(z) < k\} \tag{4.17}$$

$$\mathcal{D}_k = \{Q(z) \mid Q(z) = V(z)\overline{\Pi}_k(z) ; \deg V(z) < m-k\} \tag{4.18}$$

It is not hard to show that if a matrix polynomial H(z) belongs to $\mathcal{P}_k$ (resp. $\mathcal{D}_k$), then its image by the operator **F** defined by (4.15) belongs also to $\mathcal{P}_k$ (resp. $\mathcal{D}_k$). The case k=1 is particularly interesting.

Let $F(z) = (I - Lz)F_1(z) = \overline{F}_1(z)(I - Rz)$. L and R being respectively the dominant left solvent and the dominant right solvent of the matrix polynomial $A(\lambda)$. Let us consider also the block row vectors associated with $F_1(z)$ and $\overline{F}_1(z)$.

$$\vec{F}_1 = (I, A_{1,1}, \dots, A_{1,m-1})$$

$$\vec{\overline{F}}_1 = (I, \overline{A}_{1,1}, \dots, \overline{A}_{1,m-1})$$

Applying the operator **F** to $F_1(z)$ gives:

$$\begin{aligned}\{\vec{F}_1\mathbf{F}\}(z) &= z^{-1}[F_1(z) - F_1(0)F(z)] \\ &= z^{-1}[F_1(z) - (I - Lz)F_1(z)] \\ \{\vec{F}_1\mathbf{F}\}(z) &= LF_1(z)\end{aligned}$$

In other words, $\vec{F}_1$ represents the span of a left invariant subspace for the block companion matrix **F**.

$$\vec{F}_1\mathbf{F} = L\vec{F}_1$$

Applying now the operator **F** to $\overline{F}_1(z)$ gives:

$$\left(\overrightarrow{F}_1 \mathbf{F}\right)(z) = z^{-1}[\overline{F}_1(z) - \overline{F}_1(0)F(z)]$$
$$= z^{-1}[\overline{F}_1(z) - \overline{F}_1(z)(I - Rz)] = \overline{F}_1(z)R$$

The above relation can be rephrased as:

$$\overrightarrow{F}_1 \mathbf{F} = \overrightarrow{F}_1(I \circledast R)$$

which is the same as equation (2.24).

For the general factorization of F(z), the set $\mathcal{P}_k$ dominates $\mathcal{D}_k$ in the following sense:

Proposition 4.11:

Let $\vec{Q}_n = \vec{Q}_0 \mathbf{F}^n$ *with* $Q_0(z) \in \mathcal{D}_k$, *then, for any* w *such that:* $|w| < \rho < |\lambda_{k+1}|$, *we have:*

$$|Q_n(w)| \le \frac{K|F(w)|\rho^{-n}}{1 - \rho^{-1}|w|} \tag{4.19}$$

Proof:

We have:

$$Q_{n+1}(z) = z^{-1}[Q_n(z) - Q_n(0)F(z)]$$

Let us define the following rational matrices: $H_n(z) = Q_n(z)F^{-1}(z)$. If we replace the definition of Q_{n+1} in H_{n+1}, we obtain:

$$H_{n+1}(z) = z^{-1}[H_n(z) - Q_n(0)]$$

However: $H_n(0) = Q_n(0)F^{-1}(0) = Q_n(0)$ because $F(0) = I$. Hence:

$$H_{n+1}(z)=z^{-1}[H_n(z)-H_n(0)] \tag{4.20}$$

So if $H_0(z)=C_0+C_1z+C_2z^2+\dots$ then $H_n(z)=C_n+C_{n+1}z+C_{n+2}z^2+\dots$

Now, proposition 4.7 shows that $H_0(z)$ is analytic for $|z|\le\rho<|\lambda_{k+1}|$. In this case, corollary 4.6 implies that:

$$|C_n|\le K\rho^{-n}$$

Thus, for $|w|<\rho$, we can write:

$$|H_n(w)|\le\sum_{i=0}^{\infty}K\rho^{-n-i}|w|^i\le\frac{K\rho^{-n}}{1-\rho^{-1}|w|}$$

Since $Q_n(w)=H_n(w)F(w)$ then:

$$|Q_n(w)|\le|H_n(w)|\,|F(w)|$$

$$\Rightarrow \qquad |Q_n(w)|\le\frac{K|F(w)|\rho^{-n}}{1-\rho^{-1}|w|}$$

(Q.e.d.).

Corollary 4.12:

Under the same conditions as proposition 4.11, if $Q_n(z)=Q_{n,0}+Q_{n,1}z+\dots+Q_{n,j}z^j$ *then*

$$|Q_{n,k}|\le L\rho^{-n} \quad k=0,1,\dots,j \tag{4.21}$$

Proof:

Any polynomial can be considered as a Taylor series around zero. So we can compute the coefficients of $Q_n(z)$ from:

$$Q_{n,k} = \frac{1}{2\pi j} \oint_{\Gamma} Q_n(z) z^{-k-1} dz$$

Γ being a circle of radius $r = |w| < \rho$. Thus:

$$|Q_{n,k}| \leq \frac{1}{2\pi} \oint_{\Gamma} |Q_n(z)| \, |z^{-k}| \frac{ds}{|z|}$$

$$|Q_{n,k}| \leq \left[\frac{K|F(w)|}{1 - \rho^{-1}|w|} \right] \frac{1}{|w|^k} \rho^{-n}$$

Let

$$L = \max_k \frac{K|F(w)| \, |w|^{-k}}{1 - \rho^{-1}|w|} \quad ; \quad k \in \{0, 1, \ldots, j\}$$

then

$$|Q_{n,k}| \leq L \rho^{-n}$$

(Q.e.d.).

Proposition 4.11 suggests that if we use the block power method as defined by iteration (4.15) with an almost arbitrary polynomial G(z) of degree less than m, then after a number of iterations, the part of G(z) that belongs to $\mathcal{D}_k$ will be dominated by the part that belongs to $\mathcal{P}_k$. As a consequence, we can use a combination of matrix polynomials that will converge toward an element of $\mathcal{P}_k$. This is shown in more details in the following theorem.

Theorem 4.13:

Let $G_i^{(0)}(z)$, i=1,2,...,k matrix polynomials of degree less than m, define $G_i^{(n)}(z)$ such that $\vec{G}_i^{(n)}=\vec{G}_i^{(0)}\mathrm{F}^n$ and F(z) as in (4.16), then, under certain restrictions[1] on $G_i^{(0)}(z)$, i=1,2,...,k, for large n, there exist rxr matrices $\Phi_{k,1}^{(n)},\Phi_{k,2}^{(n)},\ldots,\Phi_{k,k}^{(n)}$ such that:

$$U_k^{(n)}(z)=\Phi_{k,1}^{(n)}G_1^{(n)}(z)+\Phi_{k,2}^{(n)}G_2^{(n)}(z)+\ldots+\Phi_{k,k}^{(n)}G_k(z)=z^{k-1}[I+B_{k,1}^{(n)}z+B_{k,2}^{(n)}z^2+\ldots] \quad (4.22)$$

and if $|\lambda_k|<\rho<|\lambda_{k+1}|$ and $|z|\le\rho$ then:

$$\left|U_k^{(n)}(z)-z^{k-1}F_k(z)\right|=O\left(\left|\frac{\lambda_k}{\rho}\right|^n\right) \quad (4.23)$$

Proof:

Let $G_i^{(n)}(z)=P_i^{(n)}(z)+Q_i^{(n)}(z)$, i=1,...,k along with $P_i^{(n)}(z)=P_{i,0}^{(n)}(z)+P_{i,1}^{(n)}z+\ldots\in\mathcal{P}_k$ and $Q_i^{(n)}(z)=Q_{i,0}^{(n)}+Q_{i,1}^{(n)}z+\ldots\in\mathcal{D}_k$ and let

$$M_k^{(n)}=\begin{pmatrix} P_{1,0}^{(n)} & P_{1,1}^{(n)} & \cdots & P_{1,k-1}^{(n)} \\ . & . & \cdots & . \\ P_{k,0}^{(n)} & P_{k,1}^{(n)} & \cdots & P_{k,k-1}^{(n)} \end{pmatrix}$$

and

$$N_k^{(n)}=\begin{pmatrix} Q_{1,0}^{(n)} & Q_{1,1}^{(n)} & \cdots & Q_{1,k-1}^{(n)} \\ . & . & \cdots & . \\ Q_{k,0}^{(n)} & Q_{k,1}^{(n)} & \cdots & Q_{k,k-1}^{(n)} \end{pmatrix}$$

1 The restrictions will be defined in the proof.

Let $\vec{\Phi}_k^{(n)}=(\Phi_{k,1}^{(n)},\Phi_{k,2}^{(n)},\ldots,\Phi_{k,k}^{(n)})$ be a block row vector of size rxkr. We must have:

$$\vec{\Phi}_k^{(n)}(M_k^{(n)}+N_k^{(n)}) = (0,0,\ldots,0,I)$$

We have to show that the matrix $M_k^{(n)}+N_k^{(n)}$ is nonsingular for large n, so that $\vec{\Phi}_k^{(n)}$ is uniquely defined.

Let $F_k(z)=I+A_{k,1}z+A_{k,2}z^2+\ldots$ and let us define $H_i^{(n)}(z)$ such that $P_i^{(n)}(z)=H_i^{(n)}(z)F_k(z)$ (because $P_i^{(n)}(z)\in\mathcal{P}_k$). So we can write:

$$H_i^{(n)}(z)=H_{i,0}^{(n)}+H_{i,1}^{(n)}z+\ldots+H_{i,k-1}^{(n)}z^{k-1}$$

Let us also define the following square matrices:

$$\mathbf{A}_k=\begin{bmatrix} I & A_{k,1} & A_{k,2} & \ldots & A_{k,k-1} \\ 0 & I & A_{k,1} & \ldots & A_{k,k-2} \\ . & . & . & \ldots & . \\ 0 & 0 & 0 & \ldots & I \end{bmatrix}$$

and

$$\mathbf{H}_k^{(n)}=\begin{bmatrix} H_{1,0}^{(n)} & \ldots & H_{1,k-1}^{(n)} \\ . & \ldots & . \\ H_{k,0}^{(n)} & \ldots & H_{k,k-1}^{(n)} \end{bmatrix}$$

We can see that $M_k^{(n)}=\mathbf{H}_k^{(n)}\mathbf{A}_k$.

Let us consider also the inverse of the block companion matrix associated with $\Pi_k(z)$: $\mathbf{F}_{\Pi_k}$. We can use it to perform a block power iteration on matrix polynomials of degree less than k, i.e. $\vec{H}^*=\vec{H}\mathbf{F}_{\Pi_k}$ with $H^*(z)=z^{-1}[H(z)-H(0)\Pi_k(z)]$. Furthermore, if $P(z)=H(z)F_k(z)$, then $\vec{P}^*=\vec{P}\mathbf{F}$ gives:

$$P^*(z) = z^{-1}[P(z) - P(0)F(z)]$$
$$P^*(z) = z^{-1}[H(z)F_k(z) - H(0)F_k(0)\Pi_k(z)F_k(z)]$$
$$P^*(z) = z^{-1}[H(z) - H(0)\Pi_k(z)]F_k(z)$$

So we can write:

$$\{\vec{P}\mathrm{F}\}(z) = \{\vec{H}\mathrm{F}_{\pi_k}\}(z)F_k(z)$$

and if $P_i^{(n)}(z) = H_i^{(n)}(z)F_k(z)$ then $P_i^{(n+1)}(z) = H_i^{(n+1)}(z)F_k(z)$ along with $\vec{H}_i^{(n+1)} = \vec{H}_i^{(n)}\mathrm{F}_{\pi_k} = \vec{H}_i^{(0)}\mathrm{F}_{\pi_k}^{n+1}$.

Hence:

$$\mathbf{H}_k^{(n)} = \mathbf{H}_k^{(0)}\mathbf{F}_{\pi_k}^{n}$$

where:

$$\mathbf{H}_k^{(0)} = \begin{pmatrix} H_{1,0}^{(0)} & \cdots & H_{1,k-1}^{(0)} \\ . & \cdots & . \\ H_{k,0}^{(0)} & \cdots & H_{k,k-1}^{(0)} \end{pmatrix}$$

Finally, if $\mathbf{H}_k^{(0)}$ is nonsingular[2], then $M_k^{(n)}$ will be nonsingular and we have:

$$[M_k^{(n)}]^{-1} = \mathbf{A}_k^{-1}\mathbf{F}_{\pi_k}^{-n}[\mathbf{H}_k^{(0)}]^{-1}$$

From (4.20) and corollary 4.12, we can write:

$$\left|N_k^{(n)}\right| = O(\rho^{-n}) \quad \textit{since} \quad \rho < |\lambda_{k+1}|$$

and from (3.3), the previous relation implies that:

2 This is the restriction stated in the theorem.

$$\left|[M_k^{(n)}]^{-1}N_k^{(n)}\right| = O\left(\left|\frac{\lambda_k}{\rho-\epsilon}\right|^n\right)$$

where ϵ is an arbitrary small positive number (less than ρ). Thus there exists an integer N such that: for n>N $M_k^{(n)}+N_k^{(n)}$ is nonsingular and

$$[M_k^{(n)}+N_k^{(n)}]^{-1} = (I+E_k^{(n)})[M_k^{(n)}]^{-1}$$

with:

$$\left|E_k^{(n)}\right| = O\left(\left|\frac{\lambda_k}{\rho'}\right|^n\right)$$

ρ' is practically equal to ρ (taking into account that ϵ is arbitrarily small).

To show (4.23), we introduce the following block column vectors:

$$\mathbf{G}_n(z) = \begin{bmatrix} G_1^{(n)}(z) \\ \cdot \\ \cdot \\ G_k^{(n)}(z) \end{bmatrix} \quad \mathbf{Q}_n(z) = \begin{bmatrix} Q_1^{(n)}(z) \\ \cdot \\ \cdot \\ Q_k^{(n)}(z) \end{bmatrix} \quad \mathbf{P}_n(z) = \begin{bmatrix} P_1^{(n)}(z) \\ \cdot \\ \cdot \\ P_k^{(n)}(z) \end{bmatrix}$$

We can write:

$$for \quad |z| < \rho \qquad |\mathbf{Q}_n(z)| = O(\rho^{-n})$$

and:

$$\mathbf{P}_n(z) = \mathbf{H}_k^{(n)} \begin{pmatrix} I & A_{k,1} & \dots & A_{k,k-1} & A_{k,k} & \dots \\ 0 & I & \dots & A_{k,k-2} & A_{k,k-1} & \dots \\ \cdot & \cdot & \dots & \cdot & \cdot & \dots \\ 0 & 0 & \dots & I & A_{k,1} & \dots \end{pmatrix} \begin{bmatrix} I \\ zI \\ \dots \\ z^{m-1}I \end{bmatrix}$$

where the above matrix is a block Sylvester matrix. We can thus write the following relations:

$$\mathbf{e}_k[M_k^{(n)}]^{-1}\mathbf{P}_n(z) = z^{k-1}F_k(z)$$

$$\left|\mathbf{e}_j[M_k^{(n)}]^{-1}\mathbf{P}_n(z)\right| = O(1) \quad for \quad j \neq k$$

where $\mathbf{e}_j = (0,\ldots,0,I,0,\ldots,0)$ is a block row vector with k block elements (each being an rxr matrix) and the j^{th} block is an rxr identity matrix. So:

$$U_k^{(n)}(z) = \vec{\Phi}_k^{(n)}\mathbf{G}_n(z) = \mathbf{e}_k(M_k^{(n)} + N_k^{(n)})^{-1}\mathbf{G}_n(z)$$
$$= \mathbf{e}_k(I + E_k^{(n)})[M_k^{(n)}]^{-1}\mathbf{G}_n(z)$$
$$= \mathbf{e}_k(I + E_k^{(n)})[M_k^{(n)}]^{-1}(\mathbf{P}_n(z) + \mathbf{Q}_n(z))$$

So:

$$U_k^{(n)}(z) = \mathbf{e}_k[M_k^{(n)}]^{-1}\mathbf{P}_n(z) + \mathbf{e}_kE_k^{(n)}[M_k^{(n)}]^{-1}\mathbf{P}_n(z) + \mathbf{e}_k(I + E_k^{(n)})[M_k^{(n)}]^{-1}\mathbf{Q}_n(z)$$
$$= z^{k-1}F_k(z) + K_n$$

where $|K_n| = O(|\lambda_k/\rho'|^n)$.

Thus, if the matrix polynomials $G_j^{(0)}$ are selected such that $\mathbf{H}_k^{(0)}$ is nonsingular and if $|\lambda_k| < \rho < |\lambda_{k+1}|$, then, for $|z| < \rho$ we have:

$$\lim_{n\to\infty} U_k^{(n)}(z) = z^{k-1}F_k(z)$$

(Q.e.d.)

Remark: we have $U_k^{(n)}(0) = 0$ so $\{\bar{U}_k^{(n)}\mathrm{F}\}(z) = z^{-1}U_k^{(n)}(z)$.

The set of polynomials defined by (4.22) is essentially the same as Bauer's Treppen-iteration [41], but generalized to block vectors. In [41] Wilkinson shows that the Treppen-iteration is related to the scalar Q.D.. It is thus normal that we find such relation for matrix polynomials. Theorem 4.13 allows us to extract a factor from a matrix polynomial F(z). The next proposition shows that it is possible to generate the matrix polynomials $U_k^{(n)}(z)$ recursively. However, we need the following definition first.

Definition 4.4:

We say that the matrix polynomial $U_k^{(n)}(z)$ is well defined if the corresponding matrix $M_k^{(n)}+N_k^{(n)}$ is nonsingular and if the coefficient $\Phi_{k,k}^{(n)}$ is nonsingular also.

Proposition 4.14:

Let $U_{l-1}^{(n+1)}(z)$, $U_l^{(n)}(z)$ and $U_l^{(n+1)}(z)$ be well defined, then there exists a nonsingular matrix $Q_l^{(n)}$ such that:

$$z^{-1}U_l^{(n)}(z)-U_{l-1}^{(n+1)}(z)=Q_l^{(n)}U_l^{(n+1)}(z) \qquad (4.24)$$

Proof:

Let us compute $z^{-1}U_l^{(n)}(z)-U_{l-1}^{(n+1)}(z)$. According to (4.23) and the previous remark, we have:

$$z^{-1}U_i^{(n)}(z) - U_{i-1}^{(n+1)}(z) = (B_{i,1}^{(n)} - B_{i-1,1}^{(n+1)})z^{i-1} + \dots$$

$$= \left\{\vec{U}_i^{(n)}\mathbf{F}\right\}(z) - U_{i-1}^{(n+1)}(z)$$

Equation (4.22) implies:

$$z^{-1}U_i^{(n)}(z) - U_{i-1}^{(n+1)}(z) = \left\{\left[\sum_{j=1}^{i}\Phi_{i,j}^{(n)}\vec{G}_j^{(n)}\right]\mathbf{F}\right\}(z) - \sum_{j=1}^{i-1}\Phi_{i-1,j}^{(n+1)}G_j^{(n+1)}(z)$$

$$= \Phi_{i,i}^{(n)}G_i^{(n+1)} + \sum_{j=1}^{i-1}(\Phi_{i,j}^{(n)} - \Phi_{i-1,j}^{(n+1)})G_j^{(n+1)}(z)$$

So, we find that $z^{-1}U_i^{(n)}(z) - U_{i-1}^{(n+1)}(z)$ is a linear combination of $G_1^{(n+1)},\dots,G_i^{(n+1)}$ and its lowest power is i-1. $U_i^{(n+1)}(z)$ is well defined, we can write:

$$U_i^{(n+1)}(z) = z^{i-1} + \dots = \Phi_{i,1}^{(n+1)}G_1^{(n+1)}(z) + \dots + \Phi_{i,i}^{(n+1)}G_i^{(n+1)}(z)$$

Let:

$$\vec{X} = (\Phi_{i,1}^{(n)} - \Phi_{i-1,1}^{(n+1)},\dots,\Phi_{i,i-1}^{(n)} - \Phi_{i-1,i-1}^{(n+1)},\Phi_{i,i}^{(n)})$$

$$\vec{Y} = (\Phi_{i,1}^{(n+1)},\Phi_{i,2}^{(n+1)},\dots,\Phi_{i,i}^{(n+1)})$$

So, we have:

$$\vec{X}(M_i^{(n+1)} + N_i^{(n+1)}) = (0,0,\dots,B_{i,1}^{(n)} - B_{i-1,1}^{(n+1)})$$

$$\vec{Y}(M_i^{(n+1)} + N_i^{(n+1)}) = (0,0,\dots,0,I)$$

$\vec{X}$ being of rank r and $M_i^{(n+1)} + N_i^{(n+1)}$ being nonsingular, the above relation implies that $B_{i,1}^{(n)} - B_{i-1,1}^{(n+1)}$ is nonsingular. So, $\vec{X} = Q_i^{(n)}\vec{Y}$ along with $Q_i^{(n)} = B_{i,1}^{(n)} - B_{i-1,1}^{(n+1)}$ nonsingular.

(Q.e.d.).

We can rewrite (4.24) as:

$$U_{i-1}^{(n+1)}(z) = z^{-1}U_i^{(n)}(z) - Q_i^{(n)}U_i^{(n+1)}(z)$$

and this relation can be iterated if the whole table of $U_i^{(n)}(z)$ is well defined. This generalization is given by the following theorem.

Theorem 4.15:

Let the whole table $U_i^{(n)}(z)$, $i=1,\dots,m-1$, $n=0,1,\dots$ be well defined, then there exist unique constants rxr matrices $C_j^{(n)}(i,p)$ such that:

$$U_{i-p}^{(n+p)}(z) = z^{-p}U_i^{(n)} + C_1^{(n)}(i,p)z^{-p+1}U_i^{(n+1)}(z) + \dots + C_p^{(n)}(i,p)U_i^{(n+p)}(z) \quad (4.25)$$

Proof:

Relation (4.25) is trivially verified for p=0 if we define $C_0^{(n)}(i,p) = I$. For p=1, (4.25) is just a rewriting of (4.24) with $C_1^{(n)}(i,1) = -Q_i^{(n)}$.

Let us assume that (4.25) is true for p.

$$U_{i-p}^{(n+p)}(z) = z^{-p}U_i^{(n)}(z) + C_1^{(n)}(i,p)z^{p+1}U_i^{(n+1)}(z) + \dots + C_p^{(n)}(i,p)U_i^{(n+p)}(z)$$

multiplying the above relation by z^{-1}, we obtain:

$$z^{-1}U_{i-p}^{(n+p)}(z) = z^{-p-1}U_i^{(n)}(z) + C_1^{(n)}(i,p)z^{-p}U^{(n+1)}(z) + \dots + C_p^{(n)}(i,p)z^{-1}U_i^{(n+p)}(z)$$

Using (4.24), we can write:

$$z^{-1}U_{i-p}^{(n-p)}(z) = U_{i-p-1}^{(n+p+1)}(z) + Q_{i-p}^{(n+p)}U_{i-p}^{(n+p+1)}(z)$$

So:

$$U_{i-p-1}^{(n+p+1)}(z) = z^{-p-1}U_i^{(n)}(z) + C_1^{(n)}(i,p)z^{-p}U_i^{(n+1)}(z) + \ldots + C_p^{(n)}(i,p)z^{-1}U_i^{(n+p)}(z) - Q_{i-p}^{(n+p)}U_{i-p}^{(n+p+1)}(z)$$

However, $U_{i-p}^{(n+p+1)}(z)$ can be expressed as:

$$U_{i-p}^{(n+p+1)}(z) = z^{-p}U_i^{(n+1)}(z) + C_1^{(n+1)}(i,p)z^{-p+1}U_i^{(n+2)}(z) + \ldots + C_p^{(n+1)}(i,p)U_i^{(n+p+1)}(z)$$

So, finally, we have an expression of $U_{i-p-1}^{(n+p+1)}(z)$ as:

$$U_{i-p-1}^{(n+p+1)}(z) = z^{-p-1}U_i^{(n)}(z) + C_1^{(n)}(i,p+1)U_i^{(n+1)}(z) + \ldots + C_{p+1}^{(n)}(i,p+1)U_i^{(n+p+1)}(z)$$

if we define the matrices $C_j^{(n)}(i,p+1)$ by the following recursion:

$$C_0^{(n)}(i,p) = I$$
$$C_j^{(n)}(i,p+1) = C_j^{(n)}(i,p) - Q_{i-p}^{(n+p)}C_{j-1}^{(n+1)}(i,p) \qquad (4.26)$$
$$j = 1,\ldots,p+1$$

The uniqueness of the set of matrices C is demonstrated by defining the following block row vector:

$$\vec{Y} = (I, C_1^{(n)}(i,p),\ldots,C_p^{(n)}(i,p))$$

We can express (4.25) in block matrix form as:

$$\vec{Y}\begin{pmatrix} I & B_{i,1}^{(n)} & \ldots & B_{i,p}^{(n)} \\ 0 & I & \ldots & B_{i,p-1}^{(n+1)} \\ . & . & \ldots & . \\ 0 & 0 & \ldots & I \end{pmatrix} = (I, B_{i-p,1}^{(n+p)},\ldots)$$

Since the block matrix on the left is nonsingular, $\vec{Y}$ is uniquely defined.

(Q.e.d.).

The set of matrices C for a given i,n,p can be constructed from the recursion (4.26). Those coefficients can also be used to defined a new set of matrix polynomials. Let:

$$V_{i,p}^{(n)}(z) = I + C_1^{(n)}(i,p)z + \ldots + C_p^{(n)}(i,p)z^p \tag{4.27}$$

The comonic matrix polynomials V can be generated recursively using the following algorithm:

$$V_{i,0}^{(n)}(z) = I$$
$$V_{i,p+1}^{(n)}(z) = V_{i,p}^{(n)}(z) - zQ_{i-p}^{(n+p)}V_{i,p}^{(n+1)}(z) \tag{4.28}$$

The matrix polynomials V are important because they converge to factors of F(z) if the conditions of the following theorem are satisfied.

Theorem 4.16:

Let F(z) be a comonic matrix polynomial of degree m and order r that satisfies the properties Γ_k and Γ_l; k<l. Let F(z) have the following factorization: $F(z) = \Pi_k(z)G_{kl}(z)F_l(z)$. Let also λ_k be the latent root of largest modulus of $\Pi_k(z)$, λ_{k+1} the latent root of minimum modulus of $G_{kl}(z)$, λ_l the latent root of maximum modulus of $G_{kl}(z)$ and λ_{l+1} the latent root minimum modulus of $F_l(z)$ ($\deg \Pi_k(z) = k$ and $\deg \Pi_k(z)G_{kl}(z) = l$). If $|\lambda_k| < |\lambda_{k+1}| \le |\lambda_l| < |\lambda_{l+1}|$, then

$$G_{kl}(z) = \lim_{n \to \infty} V_{l,l-k}^{(n)}(z)$$

Proof:

If we take the limit when $n \to \infty$ in theorem 4.15 (along with i=1 and p=l-k), we find:

$$\lim_{n\to\infty} U_k^{(n+l-k)}(z) = \lim_{n\to\infty} [z^{-l+k} I + C_1^{(n)}(l, l-k) z^{-l+k+1} + \ldots + C_{l-k}^{(n)}(l, l-k)] \lim_{n\to\infty} U_l^{(n+l-k)}(z)$$

Using now theorem 4.13, we find:

$$F_k(z) = \left[\lim_{n\to\infty} V_{l, l-k}(z)\right] F_l(z)$$

(Q.e.d.).

The recursion (4.28) for the particular case of p=1 generates the following first degree polynomial:

$$V_{k,1}^{(n)}(z) = I - Q_k^{(n)} z$$

and this form of V provides the following corollary.

Corollary 4.17:

If $F(z) = \Pi_{k-1}(z)(I - C^{-1} z) F_k(z)$, *F(z) has the property* Γ_k *and* Γ_{k-1} *and* $|\lambda_{k-1}| < \min_{\lambda \in \sigma(C)} |\lambda| \leq \max_{\lambda \in \sigma(C)} |\lambda| < |\lambda_{k+1}|$ *then:*

$$\lim_{n\to\infty} Q_k^{(n)} = C^{-1}$$

4.8 Relation with Dennis et Al. Algorithm.

Using the relation (4.24) given by proposition 4.14 with i=1, we obtain:

$$z^{-1} U_1^{(n)}(z) - U_0^{(n+1)}(z) = Q_1^{(n)} U_1^{(n+1)}(z)$$

Let $U_0^{(k)}(z) = F(z)$, we have:

$$z^{-1}U_1^{(n)}(z)-F(z)=Q_1^{(n)}U_1^{(n+1)}(z)$$

We introduce now $\lambda=z^{-1}$ and $A(\lambda)=\lambda^{m}F(\lambda^{-1})$, we find:

$$\lambda G^{(n)}(\lambda)-A(\lambda)=Q_1^{(n)}G^{(n+1)}(\lambda)$$

with $G^{(n)}(\lambda)=\lambda^{m}U_1^{(n)}(\lambda^{-1})$. If we define also $W_{1,1}^{(n)}(\lambda)=\lambda(V_{1,1}^{(n)}(\lambda^{-1}))$. Corollary 4.17 shows under which condition the above relation converges and at the limit, we have: $W_{1,1}^{(n)}(\lambda)\to\lambda I-L$ and $G^{(n)}(\lambda)\to G(\lambda)$, the corresponding quotient (L being the dominant left solvent). So we see that Dennis et Al. algorithm I. is just a special case of this more general algorithm.

4.9 The Left Q.D. Algorithm.

In the previous section, we started the block Treppen-iteration with an almost arbitrary set of matrix polynomials $G_i^{(0)}(z)$. Let us now use only one starting polynomial: $G_1^{(0)}(z)$ and use $\vec{G}_i^{(0)}=\vec{G}_1^{(0)}\mathbf{F}^{i-1}$, i=1,...,k. This implies that $G_i^{(n)}(z)=G_1^{(n+i-1)}(z)$.

For this particular choice of starting polynomial, the conditions for convergence as stated in theorem 4.13 can be expressed as conditions on $G_1^{(0)}(z)$ only. However, the fact that the starting polynomials are now related introduces another recurrence relation among the matrix polynomials $U_i^{(n)}(z)$.

Proposition 4.18:

If $U_l^{(n)}(z)$, $U_l^{(n+1)}(z)$ and $U_{l+1}^{(n)}(z)$ are well defined, then there exists a nonsingular rxr matrix $E_l^{(n)}$ such that:

$$U_l^{(n+1)}(z) - U_l^{(n)}(z) = E_l^{(n)} U_{l+1}^{(n)}(z) \tag{4.29}$$

Proof:

Let us compute the difference $U_l^{(n+1)}(z) - U_l^{(n)}(z)$. We have:

$$U_l^{(n+1)}(z) - U_l^{(n)}(z) = \Phi_{l,1}^{(n+1)} G_l^{(n+1)}(z) + \ldots + \Phi_{l,l}^{(n+1)}(z) - \Phi_{l,1}^{(n)} G_l^{(n)}(z) - \ldots - G_{l,l}^{(n)}(z)$$

$$= \Phi_{l,1}^{(n+1)} G_1^{(n+1)}(z) + \ldots + \Phi_{l,l}^{(n+1)} G_1^{(n+l)}(z) - \Phi_{l,1}^{(n)} G_1^{(n)}(z) - \ldots - \Phi_{l,l}^{(n)} G_1^{(n+l-1)}(z)$$

$$= -\Phi_{l,1}^{(n)} G_1^{(n)}(z) + (\Phi_{l,1}^{(n+1)} - \Phi_{l,2}^{(n)}) G_1^{(n+1)}(z) + \ldots + (\Phi_{l,l-1}^{(n+1)} - \Phi_{l,l}^{(n)}) G_1^{(n+l-1)}(z) + \Phi_{l,l}^{(n+1)} G_1^{(n+l)}(z) = [B_{l,1}^{(n+1)} - B_{l,1}^{(n)}] x^l + \ldots$$

So, let

$$\vec{X} = (-\Phi_{l,1}^{(n)}, (\Phi_{l,1}^{(n+1)} - \Phi_{l,2}^{(n)}), \ldots, (\Phi_{l,l-1}^{(n+1)} - \Phi_{l,l}^{(n)}), \Phi_{l,l}^{(n+1)})$$

We can write:

$$\vec{X}(M_{l+1}^{(n)} + N_{l+1}^{(n)}) = (0, \ldots, 0, B_{l,1}^{(n+1)} - B_{l,1}^{(n)})$$

However, $U_{l+1}^{(n)}(z)$ is well defined, so $\vec{Y} = (\Phi_{l+1,1}^{(n)}, \ldots, \Phi_{l+1,l+1}^{(n)})$ satisfies:

$$\vec{Y}(M_{l+1}^{(n)} + N_{l+1}^{(n)}) = (0, \ldots, 0, I)$$

so $\vec{X} = (B_{l,1}^{(n+1)} - B_{l,1}^{(n)})\vec{Y}$ and both $\vec{X}$ and $\vec{Y}$ are of rank r. So $E_l^{(n)} = B_{l,1}^{(n+1)} - B_{l,1}^{(n)}$ is of same rank and hence nonsingular.

(Q.e.d.)

In this section, we have made the assumption that all the matrix polynomials $U_i^{(n)}(z)$, i=1,...,m-1, n=1,2,... are well defined. In this particular case, we can combine (4.24) and (4.29) to obtain the Rhombus rules (4.1).

Proposition 4.19:

Let all the table $U_i^{(n)}(z)$ be well defined, then:

$$Q_i^{(n+1)} + E_{i-1}^{(n+1)} = Q_i^{(n)} + E_i^{(n)}$$

$$Q_i^{(n+1)} E_i^{(n+1)} = E_i^{(n)} Q_{i+1}^{(n)}$$

$$i = 1,\ldots,m-1 \;\; ; \;\; n = 1,2,\ldots$$

Proof:

Since we make the assumption that the whole table defining the matrix polynomials $U_i^{(n)}(z)$ is well defined, then we can write from (4.29):

$$z^{-1}U_i^{(n+1)}(z) - z^{-1}U_i^{(n)}(z) = z^{-1}E_i^{(n)}U_{i+1}^{(n)}(z)$$

Using (4.24), we obtain:

$$Q_i^{(n+1)}U_i^{(n+2)}(z) - Q_i^{(n)}U_i^{(n+1)}(z) = E_i^{(n)}Q_{i+1}^{(n)}U_{i+1}^{(n+1)}(z) + U_{i-1}^{(n+1)}(z) - U_{i-1}^{(n+2)}(z) + E_i^{(n)}U_i^{(n+1)}(z)$$

Now, from (4.29), we have:

$$U_{i-1}^{(n+1)}(z) - U_{i-1}^{(n+2)}(z) = -E_{i-1}^{(n+1)}U_i^{(n+1)}(z)$$

$$U_i^{(n+2)}(z) - U_i^{(n+1)}(z) = E_i^{(n+1)}U_{i+1}^{(n+1)}(z)$$

We obtain the following identity:

$$[Q_i^{(n+1)} + E_{i-1}^{(n+1)} - Q_i^{(n)} - E_i^{(n)}]U_i^{(n+1)}(z) = [E_i^{(n)}Q_{i+1}^{(n)} - Q_i^{(n+1)}E_i^{(n+1)}]U_{i+1}^{(n+1)}(z)$$

The above expression is an identity between two matrix polynomials of different degree. So, each term between brackets must be identically zero.

(Q.e.d.)

Using the above iterations, we can construct the Q.D. tableau (4.2). If we look back at corollary 4.17, we can see that if some conditions are satisfied (the ones that are stated in corollary 4.17), then the i^{th} Q column of the tableau converges. In this particular case, (4.28) shows that both $E_{i-1}^{(n)}$ and $E_i^{(n)}$ converge to zero. If a particular column $E_i^{(n)}$ does not converge to zero, we can extract factors by generating the matrix polynomials $V_{i,p}^{(n)}(z)$ and use the more general result of theorem 4.16.

4.10 An Existence Theorem for the Q.D. Algorithm.

The Q.D. Tableau (4.2) as defined by the rhombus rules (4.1) cannot be generated unless all the matrices Q_i and E_i are nonsingular. This condition is satisfied if all matrix polynomials $U_i^{(n)}(z)$ are well defined. In this section, we are going to show that is possible if and only if certain block Hankel matrices are nonsingular.

Theorem 4.20:

The Q.D. algorithm exists for all n and all k if and only if the following block Hankel matrices are nonsingular:

$$\mathbf{H}_k^{(n)} = \begin{pmatrix} C_n & C_{n+1} & \cdots & C_{n+k-1} \\ C_{n+1} & C_{n+2} & \cdots & C_{n+k} \\ . & . & \cdots & . \\ C_{n+k-1} & C_{n+k} & \cdots & C_{n+2k-2} \end{pmatrix}$$

for n=0,1,... and k=1,2,...,m and C_n *are the coefficients of the formal power series:*

$$V_1^{(0)}(z) = G_1^{(0)}(z)F^{-1}(z) = C_0 + C_1 z + C_2 z^2 + \dots$$

Proof:

We have proved (4.24) and (4.29) under the conditions that all the matrix polynomials $U_k^{(n)}(z)$ are well defined, i.e. $M_k^{(n)} + N_k^{(n)}$ is nonsingular and $\Phi_{k,k}^{(n)}$ is also nonsingular.

To obtain the Q.D. algorithm, we started the block power method (4.15) with the m-1 degree matrix polynomial $G_1^{(0)}(z)$ and we used $G_l^{(0)}(z) = G_1^{(l-1)}(z)$.

Let us define the following matrix rational fraction:

$$V_l^{(n)}(z) = G_l^{(n)}(z)F^{-1}(z)$$

We have seen in the proof of proposition 4.11 (equation (4.20)) that:

$$V_l^{(n+1)}(z) = G_l^{(n+1)}(z)F^{-1}(z) = z^{-1}[V_l^{(n)}(z) - V_l^{(n)}(0)]$$

We can also write:

$$G_l^{(n)}(z) = G_1^{(n+l-1)}(z) = V_l^{(n)}(z)F(z) = V_1^{(n+l-1)}(z)F(z)$$

with $V_1^{(0)}(z) = C_0 + C_1 z + C_2 z^2 + \ldots$. Using theorem 4.13, we have:

$$U_k^{(n)} = \Phi_{k,1}^{(n)} G_1^{(n)}(z) + \Phi_{k,2}^{(n)} G_1^{(n+1)}(z) + \ldots + \Phi_{k,k}^{(n)} G_1^{(n+k-1)}(z) = z^{k-1}[I + \ldots]$$

so

$$M_k^{(n)} + N_k^{(n)} = \begin{pmatrix} G_{1,0}^{(n)} & G_{1,1}^{(n)} & \ldots & G_{1,k-1}^{(n)} \\ . & . & \ldots & . \\ G_{1,0}^{(n+k-1)} & G_{1,1}^{(n+k-1)} & \ldots & G_{1,k-1}^{(n+k-1)} \end{pmatrix}$$

Thus from equation (4.8), we can write:

$$M_k^{(n)} + N_k^{(n)} = \begin{pmatrix} C_n & C_{n+1} & \ldots & C_{n+k-1} \\ C_{n+1} & C_{n+2} & \ldots & C_{n+k} \\ . & . & \ldots & . \\ C_{n+k-1} & C_{n+k} & \ldots & C_{n+2k-2} \end{pmatrix} \begin{pmatrix} I & A_1 & A_2 & \ldots & A_k \\ 0 & I & A_1 & \ldots & A_{k-1} \\ . & . & . & \ldots & . \\ 0 & 0 & 0 & \ldots & I \end{pmatrix}$$

$$M_k^{(n)} + N_k^{(n)} = \mathbf{H}_k^{(n)} F_k$$

F_k being nonsingular, for a given n and k, the matrix $M_k^{(n)} + N_k^{(n)}$ is nonsingular if and only if $\mathbf{H}_k^{(n)}$ is nonsingular. So, in order to have the whole table of $U_k^{(n)}(z)$, $\mathbf{H}_k^{(n)}$ has to be nonsingular for all k and n.

In this case, we can have an explicit expression for $\Phi_{k,k}^{(n)}$. We have:

$$\vec{\Phi}_k^{(n)} = (\Phi_{k,1}^{(n)}, \Phi_{k,2}^{(n)}, \ldots, \Phi_{k,k}^{(n)}) = (0, \ldots, 0, I) F_k^{-1} [\mathbf{H}_k^{(n)}]^{-1}$$

So, $\vec{\Phi}_k^{(n)}$ is the last block row of $F_k^{-1}[\mathbf{H}_k^{(n)}]^{-1}$. F_k^{-1} is a block triangular matrix with identity matrices on the diagonal, $\Phi_{k,k}^{(n)}$ must be the last block element of $[\mathbf{H}_k^{(n)}]^{-1}$.

Let us express $\mathbf{H}_k^{(n)}$ as a function of $\mathbf{H}_{k-1}^{(n)}$.

$$\mathbf{H}_k^{(n)} = \begin{pmatrix} \mathbf{H}_{k-1}^{(n)} & A_{12} \\ A_{21} & C_{n+2k-2} \end{pmatrix}$$

$$A_{12} = \begin{pmatrix} C_{n+k-1} \\ C_{k+k} \\ \cdot \\ \cdot \\ C_{n+2k-1} \end{pmatrix} \qquad A_{21} = (C_{n+k-1}, C_{n+k}, \ldots, C_{n+2k-1})$$

Since $\mathbf{H}_k^{(n)}$, $\mathbf{H}_{k-1}^{(n)}$ and $\mathbf{H}_1^{(n+2k-2)} = C_{n+2k-2}$ are nonsingular, we obtain:

$$\Phi_{k,k}^{(n)} = \left[\mathbf{H}_1^{(n+2k-2)} - A_{21}[\mathbf{H}_{k-1}^{(n)}]^{-1} A_{12}\right]^{-1}$$

So we see that $\Phi_{k,k}^{(n)}$ is nonsingular for all k and n and thus $Q_k^{(n)}$ and $E_k^{(n)}$ are nonsingular for all k and n.

(Q.e.d.).

Since $\Phi_{k,k}^{(n)}$ can be expressed as a function of C_n, we can also give an expression for $Q_k^{(n)}$ and $E_k^{(n)}$. From proposition 4.14, we can write:

$$\Phi_{k,k}^{(n)} = Q_k^{(n)} \Phi_{k,k}^{(n+1)} \;\Rightarrow\; Q_k^{(n)} = \Phi_{k,k}^{(n)} [\Phi_{k,k}^{(n+1)}]^{-1}$$

and from proposition 4.18, we have:

$$\Phi_{k,k}^{(n+1)} = E_k^{(n)} \Phi_{k+1,k+1}^{(n)} \Rightarrow E_k^{(n)} = \Phi_{k,k}^{(n+1)} [\Phi_{k+1,k+1}^{(n)}]^{-1}$$

Remark:

Since $H_1^{(n)} = C_n$ we find that $\Phi_{1,1}^{(n)} = C_n^{-1}$ and $\Phi_{1,1}^{(n+1)} = C_{n+1}^{-1}$ so:

$$Q_1^{(n)} = C_n^{-1} C_{n+1} \tag{4.30}$$

and this is the left Bernoulli's iteration.

Equation (4.30) provides us with an initial column for the Q.D. iteration. If we use also the fact that $E_0^{(n)} = 0$, we can generate the Q.D. tableau by columns:

Column Generation:

$$E_k^{(n)} = Q_k^{(n+1)} - Q_k^{(n)} + E_{k-1}^{(n+1)} \tag{4.31a}$$

$$Q_{k+1}^{(n)} = [E_k^{(n)}]^{-1} Q_k^{(n+1)} E_k^{(n+1)} \tag{4.31b}$$

The derivation of the above relations was done using comonic matrix polynomials, however we can use directly the Q.D. tableau for monic matrix polynomials. We should keep in mind that the spectrum of a comonic matrix polynomial is composed of the inverse of the latent roots of the corresponding monic matrix polynomial.

So, if all $E_k^{(n)}$ converge toward zero then, for a monic matrix polynomial $A(\lambda)$ of degree m and order r, we have the following result:

$$A(\lambda) = (\lambda I - Q_1)(\lambda I - Q_2)\ldots(\lambda I - Q_m)$$

where the following dominance relation exists:

$$Q_1 > Q_2 > \ldots > Q_m > 0$$

and the matrices Q_k are the limits of the columns $Q_k^{(n)}$ when n becomes very large. This is just a consequence of corollary 4.17. If it happens that a particular column $E_k^{(n)}$ does not converge, we can use the more general result of theorem 4.16. In this particular case, we extract a set of matrix polynomials using the relations (4.28) which can be rewritten for monic matrix polynomials as:

$$G_{l,0}^{(n)}(\lambda) = I$$

$$G_{l,p+1}^{(n)} = \lambda G_{l,p}^{(n)}(\lambda) - Q_{l-p}^{(n+p)} G_{l,p}^{(n+1)}(\lambda) \qquad (4.32)$$

Example:

Let us assume that $A(\lambda)$ is a fourth degree monic matrix polynomial, that the Q.D. scheme exists and that $\lim_{n\to\infty} E_1^{(n)} = 0, \lim_{n\to\infty} E_2^{(n)} = 0$, while $E_3^{(n)}$ does not converge. In this case we have a right factor of degree two that can be extracted from $A(\lambda)$.

$$G_{4,0}^{(n)}(\lambda) = I$$

$$G_{4,1}^{(n)}(\lambda) = \lambda I - Q_4^{(n)}$$

$$G_{4,2}^{(n)}(\lambda) = \lambda^2 I - \lambda Q_4^{(n)} - Q_3^{(n+1)}[\lambda I - Q_4^{(n+1)}]$$

$$G_{4,2}^{(n)}(\lambda) = \lambda^2 I - \lambda[Q_4^{(n)} + Q_3^{(n+1)}] + Q_3^{(n+1)} Q_4^{(n+1)}$$

In this case, we have the following factorization:

$$A(\lambda) = (\lambda I - Q_1)(\lambda I - Q_2)(\lambda^2 I - \lambda C + D)$$

along with:

$$C = \lim_{n\to\infty} Q_4^{(n)} + Q_3^{(n+1)} \quad ; \quad D = \lim_{n\to\infty} Q_3^{(n+1)} Q_4^{(n+1)}$$

We also have the following dominance relation:

$$Q_1 > Q_2$$

$$\min_{\lambda\in\sigma(Q_2)} |\lambda| > \max_{\lambda\in\sigma(\lambda^2 I - \lambda C + D)} |\lambda|$$

However, the column generation of the Q.D. tableau is numerically unstable. This can be understood from equation (4.31a). If the Q.D. scheme converges, then in (4.31a) we will be subtracting numbers that are almost equal. This will lead to a catastrophic cancellation of significant digits. And since the matrices $E_i^{(n)}$ converge to zero, we will be adding a large error to a small number. A more stable way for generating the Q.D. scheme will be studied in the next chapter.

Chapter 5

The Q.D. Algorithm: Block Matrix Methods

5.1 Introduction.

In the previous chapter, we have introduced the Q.D. algorithm using power series. The derived algorithm was shown to be a generalization of Bernoulli's iteration. In chapter 3, we have seen an equivalence between Bernoulli's method and the block power method. In this chapter, we are going to present a block matrix method which is a generalization of the block power method: The block L.R. method. This technique is a generalization to block matrices of Rustishauser's L.R. algorithm [41]. The interest of this presentation is that it provides us with an alternate way to start the Q.D. algorithm, the row generation, which is more stable numerically.

In this chapter, we will present the Q.D. algorithm for the more restrictive case of a complete factorization of a monic matrix polynomial. We will present only the right Q.D. keeping in mind that the left Q.D. algorithm can be derived by transposition.

Note:

All the block matrices that will be used in this chapter have rxr square matrices as block elements.

The matrix polynomial that we will consider in this chapter is:

$$A(\lambda) = \lambda^m I + A_1 \lambda^{m-1} + \ldots + A_{m-1}\lambda + A_m \tag{5.1}$$

$$A(\lambda) = (\lambda I - Q_m)(\lambda I - Q_{m-1}) \ldots (\lambda I - Q_1) \tag{5.2}$$

5.2 The Block L.R. Algorithm.

In chapter 2, we have seen that monic matrix polynomials have standard triples which define their spectral properties. A important set of standard triple is the set of block companion matrices (2.13) and (2.14). However, in our case, since $A(\lambda)$ can be factored as shown in (5.2), we can use theorem 2.12 and use the standard triple (2.37). The problem that we have to solve in this section is to find a transformation (or a sequence of transformations) that will lead us from a block companion matrix to a block bidiagonal matrix T_1 as defined by (2.37). The use of block companion matrices is obvious since their elements are the coefficients of the matrix polynomial.

Let us first consider the block lower companion form:

$$P_1 = [I, 0, \dots, 0] \qquad Y_1 = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ I \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & I & 0 & \dots & 0 & 0 \\ 0 & 0 & I & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & I \\ -A_m & -A_{m-1} & -A_{m-2} & \dots & -A_2 & -A_1 \end{bmatrix} \tag{5.3}$$

In reference [32], it is shown that the block companion matrix (5.3) is similar to a matrix in block Jacobi's form:

$$M_1 = \begin{bmatrix} M_{1,1} & I & 0 & \dots & 0 & 0 \\ M_{2,1} & M_{2,2} & I & \dots & 0 & 0 \\ 0 & M_{3,2} & M_{3,3} & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & M_{m,m-1} & M_{m,m} \end{bmatrix} \tag{5.4}$$

Being block tridiagonal, M_1 can be decomposed (under certain conditions) into a product of two block bidiagonal matrices L_1 and R_1. L_1 is a lower block triangular matrix with identity matrices on the main diagonal and R_1 is an upper block triangular matrix. By using a "block LR" algorithm, we obtain a sequence of similar matrices M_n:

$$M_n = L_n R_n \qquad M_{n+1} = R_n L_n \tag{5.5}$$

$$L_n = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 \\ E_1^{(n)} & I & 0 & \dots & 0 & 0 \\ 0 & E_2^{(n)} & I & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & E_{m-1}^{(n)} & I \end{bmatrix} \tag{5.6}$$

$$R_n = \begin{bmatrix} Q_1^{(n)} & I & 0 & \dots & 0 & 0 \\ 0 & Q_2^{(n)} & I & \dots & 0 & 0 \\ 0 & 0 & Q_3^{(n)} & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & Q_m^{(n)} \end{bmatrix} \tag{5.7}$$

By identifying $M_{n+1} = R_n L_n$ and $M_{n+1} = L_{n+1} R_{n+1}$ we obtain the following "rhombus" rules:

$$\begin{gathered} Q_i^{(n+1)} + E_{i-1}^{(n+1)} = Q_i^{(n)} + E_i^{(n)} \\ E_i^{(n+1)} Q_i^{(n+1)} = Q_{i+1}^{(n)} E_i^{(n)} \\ E_0^{(n)} = E_m^{(n)} = 0 \\ i = 1, \dots, m-1 \quad ; \quad n = 1, 2, \dots \end{gathered} \tag{5.8}$$

It is clear from the expression of L_n that if the matrices $E^{(n)}$ converge to zero, then the block companion matrix C_1 will be similar to the following matrix:

$$M = \begin{bmatrix} Q_1 & I & 0 & \dots & 0 & 0 \\ 0 & Q_2 & I & \dots & 0 & 0 \\ 0 & 0 & Q_3 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & Q_m \end{bmatrix} \tag{5.9}$$

The following theorem shows that under certain conditions, the "block LR" algorithm converges.

Theorem 5.1:

Let $M_1 = X \Lambda X^{-1}$ *where:*

$$\Lambda = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ 0 & 0 & \dots & R_m \end{bmatrix}$$

If the following conditions are satisfied:

a) dominance relation between R_k*:* $R_1 > R_2 > \dots > R_m > 0$

b) $X^{-1} = Y$ *has a block LR factorization* $L_y R_y$

c) X *has a block LR factorization* $L_x R_x$

then the block LR algorithm just defined converges (i.e. $L_n \to I$*).*

Proof:

The proof is similar to the one in reference [30].

We have $M_k = L_k R_k$ and $M_{k+1} = R_k L_k$ thus:

$$M_{k+1} = L_k^{-1} L_{k-1}^{-1} \dots L_1^{-1} M_1 L_1 \dots L_{k-1} L_k = E_k^{-1} M_1 E_k \tag{5.10}$$

Defining $H_k = R_k R_{k-1} \dots R_1$ and using (5.10), we have:

$$E_k H_k = M_1^k \tag{5.11}$$

E_k being a product of block lower triangular matrices having identity matrices on the main diagonal, will also have the same form. So $E_k H_k$ is the block LR factorization of M_1^k and we will express M_1^k as a product of a block lower and a block upper triangular matrix. We have:

$$M_1^k = X\Lambda^k X^{-1}$$

$$M_1^k = X\Lambda^k L_y R_y$$

$$M_1^k = X(\Lambda^k L_y \Lambda^{-k})(\Lambda^k R_y)$$

Since the main block diagonal of L_y is composed of identity matrices, we can write:

$\Lambda^k L_y \Lambda^{-k} = I + B_k$ where B_k is strictly block lower triangular and the (i,j) block of B_k is:

$R_i^k L_{ij} R_j^{-k}$ for i > j and 0 for i ≤ j.

Using condition a) and lemma 3.1, we have:

$$\lim_{k\to\infty}(B_k) = 0 \tag{5.12}$$

Using condition c), we have also:

$$M_1^k = L_x R_x (I + B_k)(\Lambda^k R_y)$$

$$M_1^k = L_x (I + R_x B_k R_x^{-1}) R_x (\Lambda^k R_y)$$

since $B_k \to 0$, then:

$I + R_x B_k R_x^{-1} \to I$ and it will eventually have an LR factorization for k large enough. Thus:

$I + R_x B_k R_x^{-1} = \hat{L}_k \hat{R}_k$ and both $\hat{L}_k$ and $\hat{R}_k$ converge to the identity matrix. So:

$$M_1^k = (L_x \hat{L}_k)(\hat{R}_k R_x \Lambda^k R_y)$$

We see that we have factored M_1^k into a product of a block lower triangular matrix and a block upper triangular matrix. Then:

$$E_k = L_x \hat{L}_k \text{ and } H_k = \hat{R}_k R_x \Lambda^k R_y$$

using (5.10), we finally have:

$$M_{k+1} = \hat{L}_k^{-1} L_x^{-1} M_1 L_x \hat{L}_k$$

and as $k \to \infty$, the limit is:

$$M = L_x^{-1} M_1 L_x = R_x \Lambda R_x^{-1}$$

Thus the sequence $\{M_k\}$ converges to a block upper triangular matrix.

(Q.e.d.).

5.3 The Right Q.D. Algorithm.

in theorem 5.1, we have made the implicit assumption that an LR factorization exists at each step. If such factorization cannot be made, it will lead to a breakdown of the algorithm. Furthermore, this theorem is too general for our purpose. It is not directly related to the matrix polynomial. In this section, we are going to rephrase theorem 5.1 in terms of the matrix polynomial and its solvents. We need first the following lemma.

Lemma 5.2:

The nonsingular block matrix A of size mrxmr has a unique block LDU[1] *factorization if and only if all leading block minors $A_L^{[k]}$ of size krxkr, k=1,...,m are nonsingular.*

Proof:

A particular factorization can be found by using a block gaussian decomposition.

Let

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1,m-1} & A_{1m} \\ \cdot & \cdots & \cdot & \cdot \\ A_{m-1,1} & \cdots & A_{m-1,m-1} & A_{m-1,m} \\ A_{m1} & \cdots & A_{m,m-1} & A_{m,m} \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

where $B_{22} = A_{mm}$ and $B_{11} = A_L^{[m-1]}$.

We can factorize A in the following form:

$$A = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ X & I \end{pmatrix} \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} I & Y \\ 0 & I \end{pmatrix}$$

In order to have a unique solution X and Y, $B_{11} = A_L^{[m-1]}$ has to be nonsingular. We can now repeat the same process on B_{11}, which can have the same decomposition if $A_L^{[m-2]}$ is nonsingular. We reiterate this procedure until we arrive at k=1.

(Q.e.d.).

---

1 L is a unit block lower triangular matrix, U is a unit block upper triangular matrix and D is a block diagonal matrix.

There is a converse lemma which will be stated without proof.

Lemma 5.3:

The nonsingular block matrix A of size mrxmr has a unique block UDL factorization if and only if all trailing block minors $A_T^{[k]}$ *of size krxkr, k=1,...,m are nonsingular.*

Lemma 5.2 and 5.3 can be used now to rephrase theorem 5.1 in terms of solvents of $A(\lambda)$.

Theorem 5.4:

The right Q.D. algorithm defined by (5.8) converges under the following sufficient conditions:

a) $\exists$ *m solvents* $R_1, R_2, \dots, R_m$ *such that:* $R_1 > R_2 > \dots > R_m > 0$

b) the following block Vandermonde matrices are nonsingular:

$V(R_1, \dots, R_k)$; $V(R_{m-k+1}, \dots, R_m)$; $k = 1, \dots, m$

Proof:

In order to show the convergence of the algorithm, we have to satisfy to conditions a), b) and c) of theorem 5.1. Let C_1 be the block lower companion matrix of $A(\lambda)$. If $V(R_1, \dots, R_m)$ is nonsingular, then (from theorem 2.10):

$$[V(R_1, \dots, R_m)]^{-1} C_1 V(R_1, \dots, R_m) = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ 0 & 0 & \dots & R_m \end{bmatrix}$$

So condition a) of theorem 5.1 is satisfied. Condition c) means that $X = V(R_1,\dots,R_m)$ has a block LR factorization. According to lemma 5.2, this is possible if all leading principal block minors $X_L^{[k]}$ of X are nonsingular. To satisfy condition b), $X^{-1} = Y$ must have a block LR factorization. This means that X must have a block RL decomposition. X can also be decomposed into $X = U_1 D_1 L_1$ if all the trailing principal block minors $X_T^{[k]}$ are nonsingular according to lemma 5.3. Thus we must have

$X_L^{[k]} = V(R_1,\dots,R_m)$ nonsingular for k=1,...,m and

$X_T^{[k]} = V(R_{m-k+1},\dots,R_m) BlockDiagonal(R_{m-k+1}^k,\dots,R_m^k)$

nonsingular for k=1,...,m.

(Q.e.d.)

Theorem 5.4 provides us with a convergence proof which is a generalization of theorem 3.3. The reader should remark that if the conditions stated in theorem 5.4 are satisfied then Bernoulli's iteration converges to the dominant right solvent R_1.

5.4 The Row Generation of the Q.D. Tableau.

We have see in the previous chapter that we can generate the Q.D. tableau by columns starting from the column produced from Bernoulli's iteration. In section 5.2, the initial block tridiagonal matrix (generated from the matrix polynomial by the algorithm in reference [32]) corresponds to a generation of the tableau by diagonals. In this section, we are going to see that we can generate the Q.D. tableau by rows.

The main problem that we have to solve is to find a transformation from a block companion matrix to a block tridiagonal matrix. We are going to use block LR decompositions starting from a block companion matrix. So let us consider the following standard pair (the block left companion form):

$$P_3 = [I, 0, \dots, 0]$$

$$C_3 = \begin{bmatrix} -A_1 & I & 0 & \dots & 0 \\ -A_2 & 0 & I & \dots & 0 \\ . & . & . & \dots & . \\ -A_{m-1} & 0 & 0 & \dots & I \\ -A_m & 0 & 0 & \dots & 0 \end{bmatrix} \tag{5.13}$$

C_3 is similar to C_2 defined by (2.14). The transformation matrix is the following permutation matrix:

$$P = \begin{bmatrix} 0 & 0 & \dots & O & I \\ 0 & 0 & \dots & I & 0 \\ . & . & \dots & . & . \\ I & 0 & \dots & 0 & 0 \end{bmatrix}$$

The transformation we seek is a sequence of LR decompositions. Let:

$$C_3 = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ X & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

where:

$$C_{11} = \begin{bmatrix} -A_1 & I & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ -A_{m-1} & 0 & \dots & 0 \end{bmatrix} \quad C_{12} = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ I \end{bmatrix}$$

$$C_{12} = [-A_m, 0, \dots, 0] \quad C_{22} = [0]$$

We want to have C=0. Let:

$$X = [X_1, X_2, \dots, X_{m-1}]$$

we obtain the following set of equations:

$$-X_1 A_1 - X_2 A_2 - \dots - X_{m-1} A_{m-1} = -A_m$$

$$X_1 = X_2 = \dots = X_{m-2} = 0$$

$$X_{m-1} + D = 0$$

Finally, we obtain the following decomposition for C_3:

$$C_3 = \begin{bmatrix} I & 0 & \dots & 0 & 0 & 0 \\ 0 & I & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 0 & I & 0 \\ 0 & 0 & \dots & 0 & A_m A_{m-1}^{-1} & I \end{bmatrix} \begin{bmatrix} -A_1 & I & \dots & 0 & 0 \\ -A_2 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot \\ -A_{m-1} & 0 & \dots & 0 & I \\ 0 & 0 & \dots & 0 & -A_m A_{m-1}^{-1} \end{bmatrix}$$

It is understood that we have made the implicit assumption that A_{m-1} is not singular. C_3 has been decomposed into a product of two matrices:

$$C_3 = L_{-(m-2)} R_{-(m-2)}$$

let:

$$C_3^* = R_{-(m-2)} L_{-(m-2)}$$

C_3^* is similar to C_3 and the transformation does not modify P_3. We can continue this process on the block C_{11}. So, if all the coefficient A_k of the matrix polynomial are nonsingular, we can start the LR algorithm from C_3 and we obtain the following decomposition of C_3:

$$C_3 = L_{-(m-2)}L_{-(m-3)}\dots L_0 R_0$$

where:

$$R_0 = \begin{bmatrix} -A_1 & I & 0 & \dots & 0 & 0 \\ 0 & -A_2A_1^{-1} & I & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & -A_{m-1}A_{m-2}^{-1} & I \\ 0 & 0 & 0 & \dots & 0 & -A_mA_{m-1}^{-1} \end{bmatrix}$$

$$L_0 = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 \\ A_2A_1^{-1} & I & 0 & \dots & 0 & 0 \\ 0 & 0 & I & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & I \end{bmatrix}$$

$$L_{-1} = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 \\ 0 & I & 0 & \dots & 0 & 0 \\ 0 & A_3A_2^{-1} & I & \dots & O & O \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & I \end{bmatrix}$$

etc... This set of matrices have the same format as in equation (5.6). So, it means that we can start the Q.D. algorithm with:

$$E_1^{(0)} = A_2A_1^{-1} \;;\; E_2^{(-1)} = A_3A_2^{-1} \;;\; \dots \;;\; E_{m-1}^{(-m+2)} = A_mA_{m-1}^{-1} \qquad (5.14)$$

and we deduce that:

$$Q_1^{(0)} = -A_1 \ ; \ Q_2^{(-1)} = 0 \ ; \ \dots \ ; \ Q_m^{(-m+1)} = 0 \tag{5.15}$$

Equations (5.14) and (5.15) provide us with the first two rows of the Q.D. tableau (one row of Q's and one row of E's). So, we can solve the rhombus rules (5.8) for the bottom element (called the south element by Henrici [17]). We obtain the row generation of the Q.D. algorithm:

$$Q_i^{(n+1)} = Q_i^{(n)} + E_i^{(n)} - E_{i-1}^{(n+1)} \tag{5.16a}$$

$$E_i^{(n+1)} = Q_{i+1}^{(n)} E_i^{(n)} [Q_i^{(n+1)}]^{-1} \tag{5.16b}$$

As a conclusion, we can state that given a monic matrix polynomial $A(\lambda)$ as in (5.1) with all its coefficients nonsingular, we can generate the following Q.D. tableau:

$$\begin{array}{cccccccc}
 & -A_1 & & 0 & & 0 & & \dots \\
0 & & A_2 A_1^{-1} & & A_3 A_2^{-1} & & & \dots \\
 & Q_1^{(1)} & & Q_2^{(0)} & & . & & \dots \\
0 & & E_1^{(1)} & & E_2^{(0)} & & & \dots \\
 & Q_1^{(2)} & & Q_2^{(1)} & & . & & \dots \\
0 & & E_1^{(2)} & & E_2^{(1)} & & & \dots \\
 & Q_1^{(3)} & & Q_2^{(2)} & & . & & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & & \dots
\end{array}$$

We have proved theorem 5.4 starting from the lower companion pair C_1, P_1 and the Q.D. tableau has been generated from C_3, P_3. However, C_3 and P_3 are given by:

$$C_3 = PC_2P \quad P_3 = PP_2$$

and from Gohberg et Al. [10], it is shown that:

$$C_2 = BC_1B^{-1} \quad P_2 = P_1B^{-1}$$

where:

$$B = \begin{bmatrix} A_{m-1} & A_{m-2} & \dots & A_1 & I \\ A_{m-2} & A_{m-3} & \dots & I & 0 \\ . & . & \dots & . & . \\ A_1 & I & \dots & 0 & 0 \\ I & 0 & \dots & 0 & 0 \end{bmatrix}$$

So:

$$C_3 = PBC_1B^{-1}P$$

$$PB = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ A_1 & I & 0 & \dots & 0 \\ A_2 & A_1 & I & \dots & 0 \\ . & . & . & \dots & . \\ A_{m-1} & A_{m-2} & A_{m-3} & \dots & I \end{bmatrix}$$

In theorem 5.4, we can see that the transformation matrix is a product of PB with the block Vandermonde matrix. So the conditions of theorem 5.4 remain unchanged because all the principal block minors of PB are nonsingular.

Chapter 6

Local Methods

6.1 Introduction.

We have developed in chapters four and five a global method for the factorization of monic matrix polynomials. However, the Q.D. algorithm is only linearly convergent. It can have a very slow convergence. This is why we will use it only as a first step for a two stage algorithm. It will be followed by a fast converging method derived from Newton's method [6,23,34]: Broyden's method [6]. The proposed algorithm enjoys superlinear convergence but needs an initial approximation that is quite close to the solution (see definition 3.1).

6.2 General Definitions.

In this chapter, we are going to change the statement of the problem. Instead of looking for general factors, we are going to look only for right linear ones. In this case, we can invoke the remainder theorem (corollary 2.2) and use the equivalence given by relation (2.22). So, for the monic matrix polynomial $A(\lambda)$ of degree m and order r, we have:

$$A_R(X)=0 \quad iff \quad A(\lambda)=Q(\lambda)(\lambda I-X) \tag{6.1}$$

where:

$$A_R(X)=X^m+A_1X^{m-1}+\ldots+A_m \tag{6.2}$$

and $A_R(X)$ is the right evaluation of $A(\lambda)$ at X according to definition 2.3. $\boldsymbol{A_R}$ is a nonlinear operator that maps the space of square rxr matrices onto itself. Since the space of complex rxr square matrices is a Banach space under any matrix norm, we can use powerful results from functional analysis [22,43]. This space is also a finite dimensional space and as such the equation $A_R(X)=0$ is a set of r^2 nonlinear equations with r^2 unknowns. In reference [6], Dennis et Al. provide the general theory for solving this type of problems using Newton and secant methods. From functional analysis, we have the following definition for a derivative [43].

Definition 6.1:

Let B_1 and B_2 be Banach spaces and F a nonlinear operator from B_1 to B_2. If there exists a linear operator L from B_1 to B_2 such that:

$$\| F(X+H)-F(X)-LH \| = o(\|H\|)$$
$$H,X\in B_1 \qquad F(X),LH\in B_2$$

then LH is called the Frechet differential of F at X and is written $dF(X,H)$. L is called the Frechet derivative of F at X and is written as $\frac{dF}{dX}(X)$.

The Frechet derivative is also called strong derivative by Kantorovitch et Al. in [22]. This concept of derivative is used by Kratz et Al. in [23] to prove the convergence of Newton's method. We can relate the Frechet derivative concept and the Jacobian of a vector valued function of a vector by introducing the "vec" operator [13,26].

Definition 6.2:

Let X be an rxr matrix of complex numbers:

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1r} \\ . & \cdots & . \\ x_{r1} & \cdots & x_{rr} \end{pmatrix}$$

then $x = vec(X)$ is a r^2 column vector consisting of the columns of X written one after the other:

$$x = vec(X) = \begin{pmatrix} x_{11} \\ . \\ x_{r1} \\ x_{12} \\ . \\ x_{r2} \\ . \\ . \\ x_{rr} \end{pmatrix}$$

The vec operator can be used because we are working in a finite dimensional space. We obtain the following result:

$$vec(dF(X,H)) = J(x)vec(H) \qquad (6.3)$$

where F is a nonlinear operator from the space of rxr square matrices to itself, dF(X,H) is the Frechet differential of F at X, x=vec(X) and J(x) is the Jacobian of the r^2 column vector valued function vec(F(X)).

The vec operator can be used also to rephrase the initial problem.

The problem that we have to solve is to find an rxr matrix X such that $A_R(X)=0$. Let x = vec(X) and $f(x)$ = vec($A_R(X)$), then the problem that we have to solve becomes:

Find $x \in C^{r^2}$ such that $f(x)$ = 0, i.e. solve a set of r^2 nonlinear equations with r^2 unknowns.

At this point, we can derive the expression of the Frechet differential of the right evaluation of the matrix polynomial and as a consequence of (6.3) the expression of the Jacobian matrix of the function $f(x)$.

In reference [23] and in reference [34], we can find the following expression for the Frechet differential of A_R evaluated at X:

$$dA_R(X,H) = \sum_{k=1}^{m} B_k(X) H X^{m-k} \qquad (6.4)$$

where:

$$B_k(X) = \sum_{j=0}^{k-1} A_j X^{k-j-1} \qquad (6.5)$$

Using the properties of the vec operator (see reference [13,26]), we obtain the following expression for the Jacobian of $f(x)$:

$$J(x) = \sum_{k=1}^{m} (X^T)^{m-k} \otimes B_k(X) \tag{6.6}$$

As a general remark, we can see from the expression of the Jacobian of $f(x)$ that it is quite expensive to evaluate and furthermore that it is an $r^2 x r^2$ matrix.

The numerical methods that we are going to use will produce a sequence of matrices (or vectors if we use the vec operator). This sequence will ultimately converge to the solution. The speed of convergence is an important factor in the selection of a particular numerical procedure. So, we present in this section some important definitions.

Definition 6.3: (order of convergence)

*Let **B** be a Banach space and $x_* \in B$, $x_k \in B$, $k=0,1,2,\ldots$ then:*

the sequence $\{x_k\} = \{x_0, x_1, x_2, \ldots\}$ is said to converge to x_* if:

$$\lim_{k \to \infty} |x_k - x_*| = 0$$

If $\exists c \in]0,1[$ *and an integer* $N \geq 0$ *such that:*

$$\forall k \geq N \quad |x_{k+1} - x_*| \leq c|x_k - x_*|$$

then $\{x_k\}$ is said to be q-linearly convergent to x_*.

If for some sequence of positive numbers $\{c_k\}$ that converges to zero, we have:

$$|x_{k+1}-x_*| \le c_k|x_k-x_*|$$

then $\{x_k\}$ is said to converge q-superlinearly to x_*.

If $\exists p>1,\ c>0,\ p,c\in R$ and an integer $N>0$ such that:

$$\lim_{k\to\infty} x_k = x_* \quad \text{and} \quad \forall k\ge N \ |x_{k+1}-x_*|\le c|x_k-x_*|^p$$

then $\{x_k\}$ is said to converge to x_* with order at least p. If p = 2 or p = 3, the convergence is said to be q-quadratic or q-cubic respectively.

In order to present the methods, we need also the following notations.

Definition 6.4:

*Let g be a mapping from a Banach space **B** to itself. We say that g is Lipschitz continuous in $W\subset B$ with constant γ if for every $x,y\in W$, we have:*

$$|g(x)-g(y)| \le \gamma|x-y|$$

As a notation, we use: $g\in Lip_\gamma(W)$.

*We define also the neighborhood of a point x in **B** by:*

$$N(x,\rho) = \{y\in B \mid |y-x|<\rho\}$$

6.3 Newton's Method.

All the published local methods for finding the solvents of a matrix polynomial use Newton's method. This method enjoys q-quadratic convergence for solvents for which the Jacobian is nonsingular. The most general convergence theorem is given by Kantorovitch in reference [22]. Kratz et Al. present a particular proof of convergence for the special case of right evaluation of matrix polynomials in reference [23]. However, in our work, we are going to present Dennis et Al. theorem [6] because we are going to use it for the next section.

The Algorithm:

Given $f(x)$, an n-dimensional vector valued function of an n-dimensional vector x, with Jacobian J(x) (it is an nxn matrix) and an initial vector x_0, at each iteration, we solve:

$$J(x_k)s_k = -f(x_k) \tag{6.7a}$$

$$x_{k+1} = x_k + s_k \tag{6.7b}$$

The convergence is given by the following theorem:

Theorem 6.1:

Let $f(x): R^n \rightarrow R^n$ *be continuously differentiable in an open convex set* $D \subset R^n$. *Assume* $\exists x_* \in R^n$ *and* $\rho, \beta > 0$ *such that* $N(x_*, \rho) \subset D$, $f(x_*) = 0$, $J^{-1}(x_*)$ *exists with* $|J^{-1}(x_*)| \leq \beta$ *while* $J \in Lip_\gamma(N(x_*, \rho))$, *then there exists* $\epsilon > 0$ *such that for all* $x_0 \in N(x_*, \epsilon)$, *the sequence generated by:*

$$x_{k+1} = x_k - J^{-1}(x_k)f(x_k) \quad k = 0,1,\dots$$

is well defined, converges to x_* and obeys:

$$|x_{k+1} - x_*| \leq \beta\gamma |x_k - x_*|^2 \quad k = 0,1,\dots$$

Proof: see Dennis et Al. reference [6].

Theorem 6.1 simply states that if the Jacobian is nonsingular at the solution, then Newton's method converges quadratically to the solution if the initial approximation x_0 is close enough from the solution. The Lipschitz continuity of the Jacobian of $f(x) = \text{vec}(A_R(X))$ can be proved by noticing that the elements of this Jacobian are polynomials with r^2 unknowns and as such they are continuously differentiable. So the above theorem is immediately applicable to right evaluation of matrix polynomial with the following adjustments:

$n = r^2$, $f(x) = \text{vec}(A_R(X))$, $x = \text{vec}(X)$, $J(x)$ evaluated by (6.6).

6.4 Broyden's Method.

We can see that the use of Newton's method means that we have to evaluate the Jacobian $J(x)$ and the function $f(x)$, then we have to solve a system of r^2 linear equation at each step. So Newton's method has a quite high computational cost. Broyden's method is a generalization of the secant method to the multivariable case. It is shown in [6] that it has only a superlinear convergence rate. However, it is much less expensive in computations for each step.

The Algorithm:

Given $f(x)$, an n-dimensional vector valued function of an n-dimensional vector x, an initial approximation x_0 and an nxn matrix A_0, at each iteration, we compute:

$$x_{k+1} = x_k - A_k^{-1} f(x_k) \tag{6.8a}$$

$$A_{k+1} = A_k + \frac{(y_k - A_k s_k) s_k^T}{s_k^T s_k} \tag{6.8b}$$

$$y_k = f(x_{k+1}) - f(x_k)$$

$$s_k = x_{k+1} - x_k$$

The convergence is given by the following theorem:

Theorem 6.2:

Let all the hypothesis of theorem 6.1 hold. Assume also $\exists \epsilon, \delta > 0$ *such that, if* $\| x_0 - x_* \|_2 \le \epsilon$ [1] *and* $\| A_0 - J(x_*) \|_2 \le \delta$ [2]*, then the sequence generated by Broyden's algorithm is well defined and converges q-superlinearly to* x_**.*

Proof: see Dennis et Al. reference [6].

Theorem 6.2 states that we have to provide not only a good approximation of the solution but also a good approximation of the Jacobian evaluated at the solution. Its applicability to

---

1 $\| \cdot \|_2$ is the Euclidean norm.
2 This is the induced matrix norm.

right evaluation of matrix polynomials is the same as the previous theorem. We can remark also that the evaluation of the Jacobian is avoided in Broyden's method (aside from initializing the algorithm). Step (6.8a) implies the solution of an $r^2 x r^2$ system of linear equations. This computation can be avoided if we calculate directly the inverse of the matrix A_k at each step. This can be accomplished by using the Sherman-Morrison-Woodbury formula as stated in [6].

Proposition 6.3:

Let $u, v \in R^n$ and let the square nxn matrix A be nonsingular. Then $A + uv^T$ is nonsingular if and only if:

$$\sigma = 1 + v^T A^{-1} u \neq 0$$

Furthermore:

$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{\sigma} A^{-1} u v^T A^{-1} \tag{6.9}$$

The use of (6.9) in (6.8b) provides directly the following update for the inverse of the matrix A_k:

$$A_{k+1}^{-1} = A_k^{-1} + \frac{(s_k - A_k^{-1} y_k) s_k^T A_k^{-1}}{s_k^T A_k^{-1} y_k} \tag{6.10}$$

The use of (6.10) requires $O(n^2)$ operations ($(O(r^4))$) then the update (6.8a) requires only an additional $O(n^2)$ ($(O(r^4)$) operations since it is only a matrix vector multiplication.

Chapter 7

Implementation and Numerical Results

7.1 Data Structure.

In order to test the algorithms described in chapters four, five and six, we have developed a set of computer programs. The language used is VAX-11 FORTRAN which is a dialect of FORTRAN-77.

The first problem that we had to solve was to provide an appropriate data structure to represent matrix polynomials. We have opted for a three-dimensional array. We made use of the fact that FORTRAN stores arrays in column major order. So to a matrix polynomial $A(\lambda)$ of degree m and order r, we associate the array A(1:r,1:r,1:m) [1] (The coefficient A_0 is not represented because we are using monic matrix polynomials). The last index in A(I,J,K) is used to point to a particular matrix coefficient while the first two ones point to a particular element of this matrix.

We have also made use of the fact that FORTRAN passes data to subroutines by reference.

---

1 This represents an rxrxm array.

Example: A_2 in $A(\lambda)$ is passed as A(1,1,2) to a particular subroutine that will process it.

7.2 The Q.D. Algorithm.

The Q.D. algorithm basic iteration uses two three-dimensional arrays as input (Q and E) and produces two other ones as output. We have selected the row generation algorithm of chapter five. Our implementation uses also an in-place computation: the output is produced in the same arrays that are passed as input.

The row generation of the Q.D. tableau is selected for the following two reasons: numerical stability as discussed at the end of chapter four and the fact that rows have finite size while columns do not. So Q and E are declared as Q(1:r,1:r,1:m) and E(1:r,1:r,0:m).

We have seen that there exist two Q.D. algorithms: one that factorizes the matrix polynomial from the right and one that factorizes it from the left. So, we have provided two different subroutines: QDRF and QDLF for the right and left factorization respectively. Those two subroutines are straightforward applications of the formulas (5.16).

right factorization:

$$Q_i^{(n+1)} = Q_i^{(n)} + E_i^{(n)} - E_{i-1}^{(n+1)}$$

$$E_i^{(n+1)} = Q_{i+1}^{(n)} E_i^{(n)} [Q_i^{(n+1)}]^{-1} \tag{7.1}$$

left factorization:

$$Q_i^{(n+1)} = Q_i^{(n)} + E_i^{(n)} - E_{i-1}^{(n+1)}$$

$$E_i^{(n+1)} = [Q_i^{(n+1)}]^{-1} E_i^{(n)} Q_{i+1}^{(n)} \qquad (7.2)$$

n is the iteration counter. In order to appreciate the convergence of a particular algorithm, we have coded a real valued function called AMAXNORM which produces the modulus of the largest element of a two-dimensional array.

Cost of a Q.D. iteration:

We have used LU factorization in order to compute the solution of a linear matrix equation AX=B or XA=B. (A, X and B are rxr square matrices). So, from formulas (7.1) or (7.2), for a matrix polynomial of degree m and order r, we have $O(2mr^2)$ additions and $O(\frac{8}{3}(m-1)r^3)$ multiplications for each iteration (see reference [6,18,20]). For most machines, it is the multiplication time that dominates the computation time.

7.3 Broyden's Method

Broyden's method has also been coded as a straightforward application of formulas (6.8a) and (6.10). The initial value of the matrix A_0 in those formulas is taken to be the Jacobian evaluated at the initial approximation. The function f(x) is of course computed as $vec(A_R(X))$ and the variable x is equal to vec(X). So, we have provided a subroutine that transforms an rxr square matrix into a r^2 vector (one-dimensional array). The right evaluation of a matrix polynomial $A_R(X)$ is computed by Horner's algorithm (see equations (2.23)).

Cost of a Broyden's method iteration:

For a matrix polynomial of degree m and order r, the cost of one iteration seems to be mostly dependent on the order r. The degree m appears in the cost only in the evaluation of f(x) (computation of $A_R(X)$). So, from chapter six, we have $O(kr^4)$ multiplications for the evaluation of (6.8a) and (6.10). To this number, we have to add $O((m-1)r^3)$ multiplications for the evaluation of f(x).

7.4 The Complete System.

The complete program starts with the Q.D. algorithm. It is then followed by a refinement of the right factor by Broyden's method. After deflation, Broyden's method is again applied using the next Q output from the Q.D. algorithm and the process is repeated until we are left with a linear term. Of course, this process can be applied only to polynomial matrices that satisfy the conditions of theorem 5.4 (i.e. complete right and left factorization and complete dominance relation between solvents). In cases where those conditions are not satisfied, we extract second order matrix polynomials using equations (4.32).

We can see that, for a complete factorization, we have to run the Q.D. algorithm for N iterations then we have to use m-1 times Broyden's method and perform m-1 deflations. It is quite difficult to find an optimum value for N. We must not forget that Broyden's method needs a good approximation of the solution in order to converge.

7.5 Numerical Results.

To provide a reliable test of the proposed algorithms, a quite large number of matrix polynomials have been used. Some of them have been constructed especially to show a particular property of the algorithms while others have been generated randomly.

The first matrix polynomial has been used in reference [8] and in reference [14]. This polynomial is not very informative on the matrix Q.D. algorithm because its coefficients commute. So basically, it behaves as a scalar polynomial.

$$A_1(\lambda) = \lambda^3 I + \lambda^2\begin{pmatrix} -6 & 6 \\ -3 & -15 \end{pmatrix} + \lambda\begin{pmatrix} 2 & -42 \\ 21 & 65 \end{pmatrix} + \begin{pmatrix} 18 & 66 \\ -33 & -81 \end{pmatrix}$$

After 30 iterations of the right Q.D. algorithm, we obtain $|E_1^{(30)}| = 6.053\,10^{-6}$ and $|E_2^{(30)}| = 1.863\,10^{-9}$ (The norm used is the one computed by the function AMAXNORM). We obtain the following factors:

$$Q_1^{(30)} = \begin{pmatrix} 3.99999 & -2.00002 \\ 1.00001 & 7.00002 \end{pmatrix}$$

$$Q_2^{(30)} = \begin{pmatrix} 2.00001 & -1.99998 \\ 0.99991 & 4.99998 \end{pmatrix}$$

$$Q_3^{(30)} = \begin{pmatrix} 0.18\,10^{-8} & -2.00000 \\ 1.00000 & 3.00000 \end{pmatrix}$$

Because of commutativity, the left Q.D. algorithm will produce essentially the same factorization. The only difference is attributable to round-off error. In this case, we obtain the following approximate factorization:

$$A_1(\lambda) = (\lambda I - Q_3^{(30)})(\lambda I - Q_2^{(30)})(\lambda I - Q_1^{(30)})$$

The next example (from [34]) is more informative:

$$A_2(\lambda) = \lambda^3 I + \lambda^2 \begin{pmatrix} 4 & 2 \\ -2 & 7 \end{pmatrix} + \lambda \begin{pmatrix} 12 & 11 \\ -2 & 28 \end{pmatrix} + \begin{pmatrix} 19 & 14 \\ 16 & 36 \end{pmatrix}$$

Using the right Q.D. algorithm, we obtain the following tableau:

$\lvert E_0^{(n)} \rvert$	$\lvert E_1^{(n)} \rvert$	$\lvert E_2^{(n)} \rvert$	$\lvert E_3^{(n)} \rvert$	n
0	6.6875	1.865	0	1
0	11.15	1.26	0	2
0	8.656	0.137	0	3
.	.	.	.	.
.	.	.	.	.
0	$4.995\,10^{-8}$	0.729	0	27
0	$2.863\,10^{-8}$	1.03	0	28
0	$1.265\,10^{-8}$	0.88	0	29
0	$3.766\,10^{-9}$	0.26	0	30

It is apparent from the above tableau that the $E_2^{(n)}$ column does not converge. So, we extract the following matrices:

$$Q_1^{(29)} = \begin{pmatrix} 5.40678 & -5.47458 \\ 13.4915 & -9.40678 \end{pmatrix}$$

$$Q_2^{(29)} = \begin{pmatrix} -7.09819 & 3.50457 \\ -10.1949 & 4.46236 \end{pmatrix}$$

$$Q_3^{(29)} = \begin{pmatrix} -2.30859 & -0.29998\,10^{-1} \\ -1.29662 & -2.05558 \end{pmatrix}$$

and:

$$Q_1^{(30)} = \begin{pmatrix} 5.40678 & -5.47458 \\ 13.4915 & -9.40678 \end{pmatrix}$$

$$Q_2^{(30)} = \begin{pmatrix} -7.35051 & 3.61065 \\ -11.0796 & 4.83318 \end{pmatrix}$$

$$Q_3^{(30)} = \begin{pmatrix} -2.05627 & -0.136071 \\ -0.411944 & -2.42640 \end{pmatrix}$$

It is manifest from the results that $A_2(\lambda)$ can be factored only as a product of a second degree polynomial and a linear one:

$$A_2(\lambda) = (\lambda^2 I - C_1\lambda + D_1)(\lambda I - Q_1^{(30)})$$

where C_1 and D_1 are evaluated according to (4.32) transposed.

$$C_1 = Q_2^{(29)} + Q_3^{(29)}$$
$$D_1 = Q_3^{(30)} Q_2^{(29)}$$

Giving:

$$A_{21}(\lambda) = \lambda^2 I - C_1\lambda + D_1 = \lambda^2 I + \lambda\begin{pmatrix} 9.40678 & -3.47458 \\ 11.4915 & -2.40678 \end{pmatrix} + \begin{pmatrix} 15.9831 & -7.81356 \\ 27.6610 & -12.2712 \end{pmatrix}$$

We can check the convergence of the algorithm by comparing the spectra of the factors with the spectrum of $A_2(\lambda)$. Using the QR algorithm [41] on the block companion matrix of $A_2(\lambda)$, we obtain the following spectrum:

$$\sigma(A_2) = \{-2.0 + j4.3589, -2.0 - j4.3589, -1.5 + j1.65831, -1.5 - j1.65831, -2.0, -2.0\}$$

and the factors have the following spectra:

$$\sigma(Q_1^{(30)}) = \{-2.0 + j4.3589, -2.0 - j4.3589\}$$

$$\sigma(A_{21}) = \{-1.5 + j1.65831, -1.5 - j1.65831, -1.99994, -2.00006\}$$

We remark a total agreement between the spectrum of the matrix polynomial $A_2(\lambda)$ and the one of its factors as computed by the right Q.D. algorithm.

Similarly, the use of the left Q.D. algorithm on the same matrix polynomial shows the same type of convergence ($|E_1^{(n)}|$ converges to zero while $|E_2^{(n)}|$ does not). So, after 30 iterations of the left Q.D. algorithm, we obtain:

$$Q_1^{(29)} = \begin{pmatrix} -1 & -5 \\ 4 & -3 \end{pmatrix}$$

$$Q_2^{(29)} = \begin{pmatrix} -0.642995 & 3.02245 \\ -0.917371 & -1.99283 \end{pmatrix}$$

$$Q_3^{(29)} = \begin{pmatrix} -2.35701 & -0.224474\,10^{-1} \\ -1.08263 & -2.0017 \end{pmatrix}$$

and:

$$Q_1^{(30)} = \begin{pmatrix} -1 & -5 \\ 4 & -3 \end{pmatrix}$$

$$Q_2^{(30)} = \begin{pmatrix} -0.509438 & 3.3275 \\ -1.13930 & -2.00789 \end{pmatrix}$$

$$Q_3^{(30)} = \begin{pmatrix} -2.49056 & -0.327524\,10^{-1} \\ -0.860699 & -1.99211 \end{pmatrix}$$

We make use of (2.32) to obtain:

$$A_2(\lambda) = (\lambda I - Q_1^{(30)})(\lambda^2 I - C_2\lambda + D_2)$$

where:

$$C_2 = Q_2^{(29)} + Q_3^{(29)}$$

$$D_2 = Q_2^{(29)} Q_3^{(30)}$$

$$A_{22}(\lambda) = \lambda^2 I - \lambda C_2 + D_2 = \lambda^2 I + \lambda \begin{pmatrix} 3 & -3 \\ 2 & -4 \end{pmatrix} + \begin{pmatrix} -1 & -6 \\ 4 & 4 \end{pmatrix}$$

Let us now consider the combined application of the Q.D. algorithm and Broyden's method. The considered matrix polynomial is:

$$A_3(\lambda) = \lambda^3 I + \lambda^2 \begin{pmatrix} -6 & -3 \\ -1 & -6 \end{pmatrix} + \lambda \begin{pmatrix} 12 & 11 \\ 4 & 13 \end{pmatrix} + \begin{pmatrix} -9 & -12 \\ -3 & -8 \end{pmatrix}$$

15 iterations of the right Q.D. algorithm produce the following matrices:

$$Q_1^{(15)} = \begin{pmatrix} 2.99752 & 2.02064 \\ -0.185730\,10^{-1} & 3.15478 \end{pmatrix}$$

$$Q_2^{(15)} = \begin{pmatrix} 2.00182 & -0.206580\,10^{-1} \\ 1.01851 & 1.84523 \end{pmatrix}$$

$$Q_3^{(15)} = \begin{pmatrix} 1.00066 & 1.00002 \\ 0.599079\,10^{-4} & 0.999999 \end{pmatrix}$$

Using $Q_1^{(15)}$ as initial approximation for a right solvent, Broyden's method needs 13 function evaluations to produce:

$$Q_1 = \begin{pmatrix} 3 & 2 \\ -0.181033\,10^{-9} & 3 \end{pmatrix}$$

We deflate $\lambda I - Q_1$ from the right of $A_3(\lambda)$ to produce:

$$A_{31}(\lambda) = \lambda^2 I + \lambda \begin{pmatrix} -3 & -1 \\ -1 & -3 \end{pmatrix} + \begin{pmatrix} 3 & 2 \\ 1 & 2 \end{pmatrix}$$

We use now $Q_2^{(15)}$ as initial guess for the right solvent of $A_{31}(\lambda)$. In this case, Broyden's method produced the following result after 14 function evaluations:

$$Q_2 = \begin{pmatrix} 2 & 0.339978\,10^{-10} \\ 1 & 2 \end{pmatrix}$$

The last factor is of course obtain by deflation from $A_{31}(\lambda)$.

$$Q_3 = \begin{pmatrix} 1 & 1 \\ 0.273713\,10^{-9} & 1 \end{pmatrix}$$

For this particular matrix polynomial, we have thus obtained the following factorization:

$$A_3(\lambda) = (\lambda I - Q_3)(\lambda I - Q_2)(\lambda I - Q_1)$$

$$= \left(\lambda I - \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\right)\left(\lambda I - \begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}\right)\left(\lambda I - \begin{pmatrix} 3 & 2 \\ 0 & 3 \end{pmatrix}\right)$$

We have considered as zero any number less than 10^{-9}.

We will obtain an alternate factorization if we use the left Q.D. algorithm. After 20 iterations of the left Q.D. algorithm, we obtain:

$$Q_1^{(20)} = \begin{pmatrix} -3.13121 & -9.25156 \\ 4.08525 & 9.16344 \end{pmatrix}$$

$$Q_2^{(20)} = \begin{pmatrix} 4.13207 & 4.25326 \\ -1.08570 & -0.164324 \end{pmatrix}$$

$$Q_3^{(20)} = \begin{pmatrix} 4.99914 & 7.99830 \\ -1.99955 & -2.99911 \end{pmatrix}$$

We should point out that now $Q_3^{(20)}$ is the approximation of the right solvent. So, we use it as an initial value for Broyden's method. We obtain the following matrix after 7 function evaluations:

$$Q_3 = \begin{pmatrix} 5 & 8 \\ -2 & -3 \end{pmatrix}$$

Deflating $\lambda I - Q_3$ from the right of $A_3(\lambda)$, we obtain:

$$A_{31L}(\lambda) = \lambda^2 I + \lambda \begin{pmatrix} -1 & 5 \\ -3 & -9 \end{pmatrix} + \begin{pmatrix} -3 & -12 \\ 7 & 16 \end{pmatrix}$$

Using the above matrix polynomial and starting from $Q_2^{(20)}$, Broyden's method need's 10 functions evaluations to produce:

$$Q_2 = \begin{pmatrix} 4 & 4 \\ -1 & -0.929954\,10^{-10} \end{pmatrix}$$

The dominant left solvent is obtained by deflation from $A_{31L}(\lambda)$.

$$Q_1 = \begin{pmatrix} 3 & 9 \\ -4 & -9 \end{pmatrix}$$

So, we have obtained the following factorization for $A_3(\lambda)$:

$$A_3(\lambda) = (\lambda I - Q_1)(\lambda I - Q_2)(\lambda I - Q_3)$$
$$= \left(\lambda I - \begin{pmatrix} -3 & -9 \\ 4 & 9 \end{pmatrix}\right)\left(\lambda I - \begin{pmatrix} 4 & 4 \\ -1 & 0 \end{pmatrix}\right)\left(\lambda I - \begin{pmatrix} 5 & 8 \\ -2 & -3 \end{pmatrix}\right)$$

The next example is the same matrix polynomial as the one given in example in chapter 2, but with a shift in order to avoid singular coefficients (we can thus start the algorithm).

$$A_4(\lambda)=\lambda^3 I+\lambda^2\begin{pmatrix}6 & 1.41421\\1.41421 & 6\end{pmatrix}+\lambda\begin{pmatrix}12 & 4.65685\\6.65685 & 12\end{pmatrix}+\begin{pmatrix}8 & 3.65685\\7.65685 & 8\end{pmatrix}$$

It has the following spectrum:

$$\sigma(A_4)=\{-1,-1,-2,-2,-3,-3\}$$

10 iterations of the left Q.D. algorithm produce the following tableau:

$\lvert E_0^{(n)}\rvert$	$\lvert E_1^{(n)}\rvert$	$\lvert E_2^{(n)}\rvert$	$\lvert E_3^{(n)}\rvert$	n
0	0.6	0.3	0	1
0	0.3	0.24	0	2
.	.	.	.	.
.	.	.	.	.
0	$0.4 10^{-1}$	$0.17 10^{-1}$	0	10

giving:

$$Q_1^{(10)}=\begin{pmatrix}-5.56925 & -1.53491\\0.203550 & -2.40125\end{pmatrix}$$

$$Q_2^{(10)}=\begin{pmatrix}-2.10532 & -0.156707\\0.49565 10^{-1} & -1.91053\end{pmatrix}$$

$$Q_3^{(10)}=\begin{pmatrix}-0.325425 & 0.277404\\-1.66733 & -1.68822\end{pmatrix}$$

$Q_3^{(10)}$ is an approximation of a right solvent. Broyden's method needs 10 function evaluations to refine it to:

$$Q_3=\begin{pmatrix}-0.292893 & 0.292893\\-1.70711 & -1.70711\end{pmatrix}$$

and $|A_R(Q_3)| = O(10^{-11})$. After deflation, we obtain:

$$A_{41}(\lambda) = \lambda^2 I + \lambda\begin{pmatrix} 5.70711 & 1.70711 \\ -0.292893 & 4.29289 \end{pmatrix} + \begin{pmatrix} 7.41421 & 3.41421 \\ -0.585786 & 4.58579 \end{pmatrix}$$

Broyden's methods needs 11 function evaluation to refine $Q_2^{(10)}$ to:

$$Q_2 = \begin{pmatrix} -2 & 0.2\,10^{-9} \\ -0.3\,10^{-10} & -2 \end{pmatrix}$$

and the right evaluation of $A_{41}(\lambda)$ at Q_2 produces a matrix whose elements are no greater than 10^{-9}. Another step of deflation produces the dominant left solvent:

$$Q_1 = \begin{pmatrix} -3.70711 & -1.70711 \\ 0.292893 & -2.292893 \end{pmatrix}$$

The next polynomial has been constructed to have a triple latent root. This should prevent the Q.D. algorithm from converging.

$$A_5(\lambda) = \lambda^3 I + \lambda^2\begin{pmatrix} -6 & -4 \\ -2 & -5 \end{pmatrix} + \lambda\begin{pmatrix} 13 & 13 \\ 8 & 13 \end{pmatrix} + \begin{pmatrix} -12 & -15 \\ -6 & -9 \end{pmatrix}$$
$$= \left(\lambda I - \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\right)\left(\lambda I - \begin{pmatrix} 2 & 0 \\ 2 & 1 \end{pmatrix}\right)\left(\lambda I - \begin{pmatrix} 3 & 3 \\ 0 & 3 \end{pmatrix}\right)$$

The constructed spectrum can be read directly from the factored form of the matrix polynomial. However, the computed spectrum (using the QR algorithm) does not contain any triple root:

$$\sigma(A_5) = \{3.0 + j1.88\,10^{-7}, 3.0 - j1.88\,10^{-7}, 2, 1, 1 + j10^{-8}, 1 - j10^{-8}\}$$

After 11 iterations, the right Q.D. algorithm produces:

$$Q_1^{(11)} = \begin{pmatrix} 2.99993 & 3.01106 \\ -0.728529\,10^{-3} & 3.02694 \end{pmatrix}$$

$$Q_2^{(11)} = \begin{pmatrix} 2.24182 & -0.133381 \\ 2.02353 & 0.961442 \end{pmatrix}$$

$$Q_3^{(11)} = \begin{pmatrix} 0.758246 & 1.12232 \\ -0.228005\,10^{-1} & 1.01162 \end{pmatrix}$$

We can remark that $Q_1^{(11)}$ is a good approximation of the right solvent. However, $Q_2^{(11)}$ and $Q_3^{(11)}$ are both quite far from the correct factors. Starting from $Q_1^{(11)}$ as initial approximation, Broyden's method needs 10 function evaluations to produce:

$$Q_1 = \begin{pmatrix} 3 & 3 \\ 0.7\,10^{-10} & 3 \end{pmatrix}$$

After deflation, we obtain the following matrix polynomial:

$$A_{S1}(\lambda) = \lambda^2 I + \lambda \begin{pmatrix} -3 & -1 \\ -2 & -2 \end{pmatrix} + \begin{pmatrix} 4 & 1 \\ 2 & 1 \end{pmatrix}$$

Using $Q_2^{(11)}$ as initial approximation, Broyden's method applied to $A_{S1}(\lambda)$ needs the impressive number of 62 function evaluations to produce:

$$Q_2 = \begin{pmatrix} 1.99945 & 0.27475\,10^{-3} \\ 2 & 1 \end{pmatrix}$$

Newton's method needs 30 iterations to produce the same result. So, we can see that for this particular case, Broyden's method, even though slow, is still more economical than Newton's method.

The left solvent is produced by another step of deflation:

$$Q_3 = \begin{pmatrix} 1.00055 & 0.999725 \\ -0.301955\,10^{-6} & 1 \end{pmatrix}$$

The spectra of the factors are:

$$\sigma(Q_1) = \{3, 2.9999\}$$

$$\sigma(Q_2) = \{2, 0.99945\}$$

$$\sigma(Q_3) = \{1.00027 + j4.8\,10^{-4}, 1.00027 - j4.8\,10^{-4}\}$$

We remark a close agreement between computed spectra.

It appears from the previous example that if the multiplicity of a latent root exceeds the dimension of the matrix polynomial, then the local methods (Broyden and Newton) have slow convergence. So this case affects adversely both local and global methods (The Q.D. algorithm can converge because round off will create the needed dominance relation).

The last example is a fifth degree matrix polynomial that has been generated randomly.

$$A_6(\lambda) = \lambda^5 I + \lambda^4 \begin{pmatrix} 1 & 2 \\ 12 & 8 \end{pmatrix} + \lambda^3 \begin{pmatrix} 20 & 3 \\ 5 & 81 \end{pmatrix} +$$

$$\lambda^2 \begin{pmatrix} 1 & 10 \\ 10 & 2 \end{pmatrix} + \lambda \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix} + \begin{pmatrix} 100 & 1 \\ 1 & 100 \end{pmatrix}$$

After 21 iterations of the right Q.D. algorithm, we obtain the following tableau:

$\lvert E_0^{(n)}\rvert$	$\lvert E_1^{(n)}\rvert$	$\lvert E_2^{(n)}\rvert$	$\lvert E_3^{(n)}\rvert$	$\lvert E_4^{(n)}\rvert$	$\lvert E_5^{(n)}\rvert$	n
0	17.98	$2.4\,10^{-2}$	35.3	0.29	0	1
0	11.9	0.2	35.3	0.3	0	2
0	17.8	$2\,10^{-2}$	1.05	9.5	0	3
.	.	.	.	.	.	.
.	.	.	.	.	.	.
0	$6.34\,10^{-4}$	$2.9\,10^{-10}$	39.2	5.7	0	21

The above tableau shows that the first two Q columns converge, so we can extract two linear factors at the right of $A_6(\lambda)$ and that there will be a third degree factor on the left.

The output of the Q.D. algorithm is:

$$Q_1^{(21)} = \begin{pmatrix} 4.26379 & -2.82957 \\ 43.0452 & -13.4859 \end{pmatrix}$$

$$Q_2^{(21)} = \begin{pmatrix} -5.46907 & 0.974266 \\ -54.4052 & 5.50082 \end{pmatrix}$$

$Q_3^{(21)}$, $Q_4^{(21)}$ and $Q_5^{(21)}$ are not used because the corresponding E columns did not converge.

Using $Q_1^{(21)}$ as initial approximation, Broyden's algorithm used 8 function evaluations to produce:

$$Q_1 = \begin{pmatrix} 4.26427 & -2.82974 \\ 43.0455 & -13.4861 \end{pmatrix}$$

After deflation, a new application of Broyden's method produces the following matrix with 8 function evaluations:

$$Q_2 = \begin{pmatrix} -5.46955 & 0.974437 \\ -54.4055 & 5.50098 \end{pmatrix}$$

Another step of deflation produces the third degree left factor:

$$A_{61}(\lambda) = \lambda^3 I + \lambda^2 \begin{pmatrix} -0.205281 & 0.144696 \\ 0.639988 & 0.0149194 \end{pmatrix} + \lambda \begin{pmatrix} -0.0177626 & -0.110659 \\ -0.733192 & -0.073443 \end{pmatrix} + \begin{pmatrix} 5.40769 & -0.145837 \\ -0.270976 & 1.26159 \end{pmatrix}$$

We have obtained the following factorization:

$$A_6(\lambda) = A_{61}(\lambda)(\lambda I - Q_2)(\lambda I - Q_1)$$

The smallest latent roots of $A_6(\lambda)$ are the latent roots of $A_{61}(\lambda)$ while the dominant ones go with the linear terms. In this particular case, we could not obtain a complete factorization.

Chapter 8

Conclusion

As already expressed in the introduction (chapter one), matrix polynomial have become an important mathematical tool for the analysis and design of linear time invariant systems [1,5,21,24,31,32,35,40]. The spectral data of the "denominator" of the matrix transfer function determines the dynamic properties of the system under study.

The work done in this thesis provides tools for the analysis of the structure of complex time invariant linear systems. The matrix Q.D. algorithm enjoys practically the same properties as the scalar algorithm. We tried to present a quite thorough theory of the algorithm (as complete as we could).

In the development of the matrix Q.D. algorithm, some important theoretical results have been produced as by-products. We can cite theorem 2.1 which could be used for the analysis of the stability of matrix polynomials, theorem

2.7 which characterizes solvents of matrix polynomials and theorem 2.12 which provides the standard triple of a matrix polynomial having a complete factorization.

In chapter four, aside from the main objective which was the convergence of the Q.D. algorithm, proposition 4.3 provides us with an important tool for the decomposition of matrix transfer functions into incomplete partial fraction expansion.

In chapter five, theorem 5.1 is a convergence theorem for a numerical method that is more general than the matrix Q.D. algorithm.

finally, we have shown that numerical techniques that are commonly used for solving nonlinear equations can be applied with advantage for finding the solvents of a matrix polynomial.

Looking back at the different numerical examples presented, we can also view our numerical methods as tools for investigation of the structure of a linear system.

If some E column in the Q.D. tableau converges, it implies that there exists a factorization of the matrix polynomial that splits the spectrum into a dominant set and a dominated one. If the system under consideration is a digital system, we know that the largest modulus latent roots have the preponderant effect on the dynamic properties of the system. In such case, the Q.D. algorithm can become a tool for system reduction (using the dominant mode concept).

Suggestions for Further Research.

(1) In the scalar case (i.e. SISO systems), the Q.D. algorithm is directly related to continued fraction expansions [16]. These expansions have been used to provide criteria for studying the stability of scalar polynomials [5]. An interesting idea would be to investigate relations between the matrix Q.D. algorithm and matrix continued fractions. This could lead to the discovery of criteria for the study of stability of matrix polynomials.

(2) In chapter five, we have investigated the convergence of a block LR algorithm. However, the proposed algorithm has a slow convergence because of the lack of shift factor. In the scalar case, the use of shift factors produces an important acceleration of the LR algorithm [41]. We should explore the use of matrix shift factors.

Another acceleration of convergence can be accomplished by using an inverse power method. The research in this case should be oriented toward the lines of theorem 2.7 and formula (2.24).

(3) The Q.D. algorithm as used in our thesis converges to factors of a matrix polynomial. By using the transformations defined in [33], we can derive the solvents. However, it would be convenient to have a global algorithm that converges directly to all solvents.

REFERENCES

[1] S. M. Ahn, *Stability of a Matrix Polynomial in Discrete Systems,* IEEE trans. on Auto. Contr., Vol. AC-27, pp. 1122-1124, Oct. 1982.

[2] E. H. Bareiss, *The Numerical Solution of Polynomial Equations and the Resultant Procedures,* in Ralston and Wilf, **Mathematical Methods for Digital Computers**, Vol. 2, John Wiley, 1967.

[3] H. W. Bode, *Relations between Attenuation and Phase in Feedback Amplifier Design,* Bell System Tech. J., Vol. 19, pp. 421-454, July 1940.

[4] C. A. Bavely, G. W. Stewart, *An Algorithm for Computing Reducing Subspaces by Block Diagonalization,* SIAM J. Numer. Anal., Vol. 16, pp. 359-367, 1979.

[5] C. T. Chen, **Linear System Theory and Design**, Holt, Reinhart and Winston, 1984.

[6] J. E. Dennis, R. B. Schnabel, **Numerical Methods for Unconstrained Optimization and Nonlinear Equations,** Prentice Hall, 1983.

[7] J. E. Dennis, J. F. Traub, R. P. Weber, *The Algebraic Theory of Matrix Polynomials,* SIAM J. Numer. Anal., Vol. 13, pp. 831-845, 1976.

[8] J. E. Dennis, J. F. Traub, R. P. Weber, *Algorithms for Solvents of Matrix Polynomials,* SIAM J. Numer. Anal., Vol. 15, pp. 523-533, 1978.

[9] J. J. DiStefano, A. R. Stubberud, I. J. Williams, **Theory and Problems of Feedback and Control Systems**, Mc. Graw Hill, 1967.

[10] I. Gohberg, P. Lancaster, L. Rodman, **Matrix Polynomials,** Academic Press, 1982.

[11] I. Gohberg, P. Lancaster, L. Rodman, *Spectral Analysis of Matrix Polynomials: I. Canonical Forms and Divisors,* Linear Algebra Appl., 20, pp. 1-44, 1978.

[12] I. Gohberg, P. Lancaster, L. Rodman, *Spectral Analysis of Matrix Polynomials: II. The Resolvent Form and Spectral Divisors,* Linear Algebra Appl., 21, 1978.

[13] A. Graham, **Kronecker Products and Matrix Calculus with Applications**, Ellis horwood Ltd., 1981.

[14] K. Hariche, *Interpolation Theory in the Structural Analysis of λ-Matrices,* Ph.D. Dissertation, University of Houston, 1987.

[15] P. Henrici, **Applied and Computational Complex Analysis**, Vol. 1, John Wiley, 1974.

[16] P. Henrici, **Applied and Computational Complex Analysis**, Vol. 2, John Wiley, 1974.

[17] P. Henrici, **Elements of Numerical Analysis**, John Wiley, 1964.

[18] P. Henrici, *The Quotient-Difference Algorithm,* Nat. Bur. Standards Appl. Math. Series, Vol. 49, pp. 23-46, 1958.

[19] F. B. Hildebrand, **Introduction to Numerical Analysis**, Mc. Graw Hill, 1974.

[20] J. A. Jensen, J. H. Rowland, **Methods of Computation**, Scott, Foresman and Co.,1975.

[21] T. Kailath, **Linear Systems**, Prentice Hall, 1980.

[22] L. V. Kantorovich, G. P. Akilov, **Functional Analysis**, 2nd Ed., Pergamon Press, 1982.

[23] W. Kratz, E. Stickel, *Numerical Solution of Matrix Polynomial Equations by Newton's Method,* IMA J. Numer. Anal. (G.B.), Vol. 7, pp. 355-369, 1987.

[24] V. Kucera, **Discrete Linear Control: The Polynomial Equation Approach**, John Wiley, 1979.

[25] P. Lancaster, *Algorithms for Lambda Matrices,* Numer. Math., Vol. 6, pp. 388-394, 1964.

[26] P. Lancaster, M. Timenetski, **The Theory of Matrices**, 2nd Ed., Academic Press, 1985.

[27] C. L. Lawson, R. J. Hanson, **Solving Least Square Problems**, Prentice Hall, 1974.

[28] M. Marcus, **Introduction to Modern Algebra**, Marcel Dekker Inc., 1978.

[29] H. Nyquist, *Regeneration Theory*, Bell Syst. Tech. J., Vol. 11, pp. 126-147, January 1932.

[30] B. N. Parlett, *The LU and QR Algorithms*, in Ralston and Wilf, **Mathematical Methods for Digital Computers**, Vol. 2, John Wiley, 1967.

[31] P. Resende, E. Kaskurewicz, *A Sufficient Condition for the Stability of Matrix Polynomials*, IEEE trans. on. Auto. Contr., Vol. AC-34, pp. 539-541, May 1989.

[32] L. S. Shieh, S. Sacheti, *A Matrix in the Block Schwarz Form and the Stability of Matrix Polynomials*, Int. J. Control, Vol. 27, pp. 245-259, 1978.

[33] L. S. Shieh, Y. T. Tsay, *Transformations of Solvents and Spectral Factors of Matrix Polynomials and their Applications*, Int. J. Control, Vol. 34, pp. 813-823, 1978.

[34] L. S. Shieh, Y. T. Tsay, N. P. Coleman, *Algorithms for Solvents and Spectral Factors of Matrix Polynomials*, Int. J. Systems Sci., Vol. 12, n°11, pp. 1301-1316, 1981.

[35] M. K. Solak, *Divisors of Polynomial Matrices: Theory and Applications*, IEEE trans. on Auto. Contr., Vol. AC-32, pp. 916-919, Oct. 1987.

[36] G. W. Stewart, **Introduction to Matrix Computations**, Academic Press, 1973.

[37] G. W. Stewart, *On a Companion Operator for Analytic Functions*, Numer. Math., Vol. 18, pp. 26-43, 1971.

[38] J. F. Traub, *A class of Globally Convergent Iteration Functions for The Solution of Polynomial Equations*, Math. Comp., Vol. 20, pp. 113-138, 1966.

[39] J. S. Tsai, L. S. Shieh, T. T. C. Shen, *Block Power Method for Computing Solvents and Spectral Factors of Matrix Polynomials*, Comput. Math. Appl. (U.K.), Vol. 16, n°9, pp. 683-699, 1988.

[40] Y. T. Tsay, L. S. Shieh, R. E. Yates, S. Barnett, *Block Partial Fraction Expansion of a Rational Matrix*, Linear and Multilinear Algebra, Vol. 11, pp. 225-241, 1982.

[41] J. H. Wilkinson, **The Algebraic Eigenvalue Problem**, University Press, Oxford, 1965.

[42] H. S. Wilf, *The Numerical Solution of Polynomial Equations*, in Ralston and Wilf, **Mathematical Methods for Digital Computers**, Vol. 1, John Wiley, 1960.

[43] A. Wouk, **A course of Applied Functional Analysis**, John Wiley, 1979.

[44] L. A. Zadeh, C. A. Desoer, **Linear System Theory: A State Space Approach**, Mc Graw Hill, 1963.

[45] L. A. Zadeh, E. Polack, **System Theory**, Mc. Graw Hill, 1969.

A N N E X E

Liste et composition du jury en vue de la soutenance de la thése de magister en Ingénerie des Systémes Electroniques.
Par Mr DAHIMENE ABDELHAKIM.

<u>PRESIDENT</u> / - Dr. R. TOUMI (Professeur à l'U.S.T.HB)

<u>RAPPORTEUR</u> / - Dr.K. HARICHE (chargé de rehcerhce, I.N.EL.E.C)

<u>EXAMINATEURS</u> / - Dr O. TOKHI (Professeur à l'Université de SHEFEIEl G.B)
- Dr B. CHANANE (chargé de cours à l'I.N.E.L.E.C)
- Dr M. DJEDDI (chargée de cours à l'I.N.E.L.E.C)