NUMERICAL ASPECTS OF REALIZATION
ALGORITHMS IN LINEAR SYSTEMS THEORY

PROEFSCHRIFT

ter verkrijging van de graad van doctor in de technische wetenschappen aan de Technische Hogeschool Eindhoven, op gezag van de rector magnificus, prof. dr. ir. G. Vossers, voor een commissie aangewezen door het college van dekanen in het openbaar te verdedigen op dinsdag 1 juli 1975 te 16.00 uur

door

Lieuwe Sytse de Jong

geboren te Leeuwarden

© 1975 by L. S. de Jong, Geldrop, The Netherlands
DIT PROEFSCHRIFT IS GOEDGEKEURD
DOOR DE PROMOTOREN

PROF. DR. G. W. VELTKAMP
EN
PROF. DR. IR. M. L. J. HAUTUS
Aan mijn Vrouwe
en Kinderen
CONTENTS

Some notations and conventions ........................................ 9

Chapter 0.  Introduction
  0.1.  problem statement ............................................ 13
  0.2.  problem history .............................................. 13
  0.3.  objectives ................................................... 21
  0.4.  fundamental notions and theorems in linear systems theory 21

Chapter 1.  Preliminary on numerical stability
  1.1.  definition of numerical stability ............................... 33
  1.2.  techniques for proving or disproving numerical stability 38

Chapter 2.  Preliminary on numerical rank
  2.1.  introduction .................................................. 49
  2.2.  the ε-stable rank ............................................. 55
  2.3.  the critical values .......................................... 59

Chapter 3.  Numerically unstable, recursive, minimal realization algorithms
  3.1.  the algorithm of Rissanen ................................. 68
  3.2.  the algorithm of Massey/Hoslempf ......................... 74
  3.3.  a new algorithm ............................................ 77
  3.4.  the algorithms of 3.1 and 3.2 as decomposition algorithms 79
  3.5.  the numerical stability of the algorithms .............. 85
  3.6.  the approach to be followed in chapter 4 .............. 93

Chapter 4.  A numerically stable, recursive, minimal realization algorithm
  4.1.  the algorithm ............................................... 95
  4.2.  the numerical stability .................................... 111
  4.3.  assessment of the algorithm ................................ 130

Chapter 5.  Filtering and approximation
  5.1.  filtering .................................................... 135
  5.2.  approximation ............................................... 140

References ............................................................. 157
**SOME NOTATIONS AND CONVENTIONS**

Matrices are denoted by capital letters, vectors by small letters.

\( e_i \) : the vector \((0, \ldots, 0, 1, 0, \ldots, 0)\)  
\( i \) : the unit matrix  
\( 0 \) : the matrix resp. vector whose elements are zeroes  
\((A)_{i,j} \) : the \((i,j)\)-th element of the matrix \(A\)  
\((x)_i \) : the \((i)\)-th element of the vector \(x\)  
\((A_1, A_2)\) : partitionings of the matrix \(A\)

\[ \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \]

\( A^T \) : the transposed matrix of \(A\)  
\( A^{-1} \) : inverse of \(A\); defined only if \(A\) is a square regular matrix  
\( A^+ \) : pseudo inverse of \(A\); it is the unique matrix satisfying  
\( AA^+ A = A, A^+ A A^+ = A^+ \) \( (A^+)^T = A^+ A, (AA^+)^T = AA^+ \)

\( r(A) \) : rank of \(A\)

\( \mathbf{X} \) : submultiplicative (i.e., \( \|AB\| \leq \|A\| \|B\| \)) matrix norm or vector norm

\[ \|x\|_2 := \left( \sum_i (x_i^2) \right)^{\frac{1}{2}} \]

\[ \|x\|_\infty := \max_i |(x)_i| \]

\[ \|A\|_2 := \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} \]

\[ \|A\|_\infty := \max_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_i (\sum_j |(A)_{i,j}|) \]

\[ \|A\|_F := \left( \sum_{i,j} (A)_{i,j}^2 \right)^{\frac{1}{2}} \]
$C(A)$ : condition number of $A$ defined by $C(A) = \|AA^T\|$.

$|A|$ : the matrix defined by $(|A|)_{i,j} = |(A)_{i,j}|$.

A matrix $A$ is said to have

- **diagonal form** if $A = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$,

- **tridiagonal form** if $A = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$,

- **upper triangular form** if $A = \begin{bmatrix} \mathbf{1} & & \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$,

- **unit upper triangular form** if $A = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$,

- **upper trapezoidal form** if $A = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$.

A matrix $A$ is said to be an

- **orthogonal matrix** if $A^T A = A A^T = I$,

- **permutation matrix** if $A$ can be transformed into $I$ by row interchanges,

- **(orthogonal) projector** if $A^2 = A$ and $A = A^T$,

- **elementary transformation matrix** if $A = I + u e_i e_j^T$ for some non-zero scalar $u$ and some pair $(i,j)$ with $i \neq j$; $u$ is called a multiplier; if $|u| < 1$ then $A$ is said to be a stabilized elementary transformation matrix.
If for some matrices $A$ and $B$ we have $AB = B$, then this is called a factorization of $A$; if $B$ is a square regular matrix, then $AB = B$ is called a decomposition of $A$.

$\mathbb{R}$ denotes the set of reals.

$M_{k,\ell}$ denotes the set of $k \times \ell$ matrices.

$M_{k,\ell}^r$ denotes the set of $k \times \ell$ matrices with rank equal to $r$.

$\int f(x)$ or $f(.)$ denotes the function $f$.

In this thesis algorithms are defined using the algorithmic language Algol 60. However, the neater Algol 68 construct

```
while <boolean expression> do <statement>
```

is used instead of the Algol 60 equivalent

```
for dummy := 0 while <boolean expression> do <statement>.
```

In any chapter theorems, lemma's and definitions are numbered $1, 2, \ldots$.
If we refer to theorem 4 (say) in some chapter, this means theorem 4 of the same chapter. If we refer to theorem 0.4 (say), this means theorem 4 of chapter 0.

Other notations are clarified in the text.
0. **INTRODUCTION**

0.1. **Problem statement**

Let $F$ be an arbitrary field. For any pair of natural numbers $(k,t)$ we denote by $M_{k,t}(F)$ the set of $k \times t$ matrices with elements from $F$.

Given a pair of natural numbers $(p,q)$ and an impulse response sequence

$$\{s_i \mid s_i \in M_{p,q}(F)\}_{i=1}^{\infty},$$

we consider the problem to determine a natural number $n$ and a triple

$$\langle A, B, C \mid A \in M_{n,p}(F), B \in M_{n,q}(F), C \in M_{p,n}(F) \rangle_n$$

such that

$$s_i = CA^{i-1}B \quad (i = 1, 2, \ldots)$$

and

$n$ is minimal.

This problem is referred to as the (complete) **minimal realization problem**. If it admits a solution, the sequence $\{s_i\}_{i=1}^{\infty}$ is said to be realizable; the solution is called a (complete) minimal realization triple of order $n$.

In this thesis we shall assume that $p = q = 1$ and that $F = \mathbb{R}$. We investigate the numerical stability of existing and new algorithms with which a minimal realization triple may be found in a recursive way.

0.2. **Problem history**

0.2.1.

The problem is classical in the form $p = q = 1$ and $F = \mathbb{R}$. Two variants in which it occurs in literature are

(i) given the sequence $\{s_i\}_{i=1}^{\infty} = \{cA^{i-1}b\}_{i=1}^{\infty}$, to determine the eigenvalues of $A$,

(ii) given the Taylor expansion at the origin of a rational function $f(z)$, to determine the poles of $f(z)$.

For a detailed survey on these variants, we refer to the excellent articles of Henrici and Parlett, [7,8] and [18]. We shall briefly review the main points.
The poles of the rational function

(1) \[ f(z) = c^T (I-zA)^{-1} b \]

are the reciprocals of the eigenvalues of \( A \). At the origin \( f(z) \) has the Taylor expansion

(2) \[ f(z) = \sum_{i=1}^m c^T A_i b z^{i-1} = \sum_{i=1}^m a_i z^{i-1} . \]

This shows the equivalence of the variants (i) and (ii).

If \( A \) has a single dominant eigenvalue \( \lambda_1 \), then (König [14, 1884])

(3) \[ \lim_{k \to \infty} \frac{a_{k+1}}{a_k} = \lambda_1 . \]

One of the commonest methods for computing \( \lambda_1 \), the so-called power method, is based on this result.

In 1892 Hadamard [5] solved the general problem to determine all eigenvalues of \( A \) with the aid of Hankel determinants as defined by

\[ H_0^k := 1, \quad H_k^k := \det \begin{bmatrix} h_1 & \cdots & h_{k+1} \\ \vdots & \ddots & \vdots \\ h_{k-k+1} & \cdots & h_{k+2k-2} \end{bmatrix} (k = 1, 2, \ldots; \lambda = 1, 2, \ldots) . \]

He found that if \( f(z) \) has \( n \) poles \( \lambda_1^{-1} \) satisfying \( |\lambda_1| > \cdots > |\lambda_n| > L > 0 \), it holds that

(4) \[ H_k^k = C(\lambda_1 \cdots \lambda_n)^k (1 + O(\lambda_1^{-k})) , \quad k = n \]

for all \( k \geq n \) with \( C \) a non-zero constant independent of \( t \). Hence, if \( z \) is large enough, we have \( H_k^k \neq 0 \) for all \( k \leq n \). This shows that the \( k \)-th eigenvalue of \( A \) satisfies

(5) \[ \lambda_k = \lim_{k \to \infty} \frac{H_{k+1}^k}{H_k^k} . \]

14
In 1931 Aitken\cite{2} proved that
\begin{equation}
(H_k^{\xi+1})_k^2 - (H_{k-1}^{\xi+2})_k = 0
\end{equation}

Using this relation the Hankel determinants can be evaluated recursively provided that each Hankel determinant of order less than \(n+1\) is unequal to zero. This may be done either by using (6) in the form
\begin{equation}
H_{k+1}^{\xi+1} = \frac{H_k^{\xi+1} - (H_k^{\xi+1})_k}{H_k^{\xi+1}}
\end{equation}
\(\xi = 2,3,\ldots; k = 1,2,3,\ldots\)

starting out with \(H_0^{\xi}\) and \(H_1^{\xi}\)\((k = 1,2,3,\ldots)\), or by using (6) in the form
\begin{equation}
H_{k+1}^{\xi+1} = \frac{H_k^{\xi+1} + (H_k^{\xi+1})_k}{H_k^{\xi+1}}
\end{equation}
\(\xi = 2,3,\ldots; k = 1,2,3,\ldots\)

starting out with \(H_0^{\xi}\) and \(H_1^{\xi}\)\((k = 0,1,2,\ldots)\).

The first method is numerically unstable (both terms in the numerator tend to the same limit); the second method has the disadvantage that \(H_k^{\xi}\) and \(H_k^{\xi+1}\)
are non-trivial determinants.

In 1954 Rutishauser\cite{20} showed how the eigenvalues may also be computed by defining
\begin{equation}
q_k^{\xi+1} = \frac{H_k^{\xi+1}}{H_k^{\xi+1}} \quad (\xi = 1,2,\ldots; k = 1,2,\ldots)
\end{equation}

and the auxiliary quantities
\begin{equation}
\sigma_0^{\xi} = 0, \quad \sigma_k^{\xi} = \frac{H_{k-1}^{\xi+1} - H_k^{\xi+1}}{H_k^{\xi+1}} \quad (\xi = 1,2,\ldots; k = 1,2,\ldots)
\end{equation}

He shows that
\begin{equation}
q_k^{\xi} = q_k^{\xi+1} + q_k^{\xi+1} \quad (\xi = 1,2,\ldots; k = 1,2,3,\ldots)
\end{equation}

and
\begin{equation}
q_k^{\xi+1} q_k^{\xi} = q_k^{\xi+1} q_k^{\xi} \quad (\xi = 1,2,\ldots; k = 0,1,2,\ldots)
\end{equation}
Since the $e_i$ are known and the $q_i$ can easily be found, all $c_i$ and $\delta_i$ can be computed by using (9) and (10) in the form

$$c_i = q_i - e_i \quad \text{and} \quad \delta_i = \frac{q_{i+1} - q_i}{e_i},$$

which is a numerically unstable method (since $e_i = q_i - q_i$ and the terms at the right-hand side tend to the same limit).

However, when the $c_i$ and $\delta_i$ are known, all $e_i$ and $q_i$ can also be computed by using (9) and (10) in the form

$$e_i = c_i - e_i \quad \text{and} \quad q_i = \frac{c_{i+1} - c_i}{e_i} (i = 1, 2, \ldots; k = 1, 2, \ldots),$$

the so-called Q-D (quotient-difference) algorithm.

In 1958 Henrici [7] showed that the matrix $A$ is similar to the tridiagonal matrices $J_i$ where $J_i = L_i R_i$ $(i = 1, 2, \ldots)$ and

$$L_i = \begin{bmatrix}
1 & 0 & & \\
\vdots & c_i & 1 & \\
\vdots & 0 & \ddots & \ddots \\
0 & \cdots & \cdots & c_{n-1}
\end{bmatrix} \quad \text{and} \quad R_i = \begin{bmatrix}
q_1 & 1 & 0 & \\
q_2 & q_1 & 1 & \\
\vdots & \vdots & \ddots & \ddots \\
0 & \cdots & \cdots & q_n
\end{bmatrix}.$$
The Padé Approximation problem [17] is a third classical variant of the realization problem. Let \( f(z) \) have the Taylor expansion (2). Let \( \mathbb{R}_t/k \) denote the class of rational functions that are ratios of polynomials in \( z \) with degrees (at most) \( t \) and \( k \).

The Padé Approximation problem is to determine for various values of \( k \) and \( t \), a member \( P_{tk} \) of \( \mathbb{R}_t/k \) whose Taylor expansion coincides with (2) as far as the term in \( z^t \).

Let us consider pairs \((k,t)\) such that \( t \geq k \). If

\[
P_{tk}(z) = \frac{a_0 + a_1 z + \ldots + a_t z^t}{1 + b_1 z + \ldots + b_k z^k}
\]

then \( \{s_i\}_{i=1}^{k+t} \) has the realization triple

\[
\begin{bmatrix}
0 & 0 & 0 \\
1 & 0 & 0 \\
& 1 & 0 \\
&& \ddots & \ddots \\
&&& 1 & 0 \\
&&&& \ddots & \ddots & \ddots \\
&&&&& 0 & 1 \\
&&&&&& -b_k \\
&&&&&&& \ddots \\
&&&&&&&& -b_1
\end{bmatrix}
, \quad (s_1, \ldots, s_k, s_{k+1}, \ldots, s_{k+t})_k.
\]

This triple is minimal, whenever \( P_{tk}(z) \) is unique, if \( t \geq k-1 \) and \( k = n \), then \( P_{tk}(z) = f(z) \) and (11) is a complete minimal realization triple of \( \{s_i\}_{i=1}^{n} \) (compare Kalman [12], Silverman [24]).

Consequently, a minimal realization of \( \{s_i\}_{i=1}^{n} \) can be given if \( P_{nk}(z) \) is known. There are a number of methods known in the literature to compute \( P_{nk}(z) \) recursively. We shall mention one method that is essentially contained in Hinzlci [7] and Handscomb [6]. To the Taylor expansion (2) there corresponds at most one continued fraction of the form

\[
f(z) = \frac{c_1}{1 + \frac{c_2}{1 + \frac{c_3}{1 + \ldots}}}
\]

so that the Taylor expansion at the origin of the \( n \)-th convergent agrees with (2) up to and including the term in \( z^n \) (\( n = 1, 2, \ldots \)).
If (12) exists then we have
\[
c_i = s_i, \quad c_{2i} = -q_i \quad (i = 1, 2, \ldots), \quad c_{2i+1} = -q_i \quad (i = 1, 2, \ldots)
\]
(see Perron [19]); it also holds that the Padé approximants
\[
P_{00}(z), \quad P_{01}(z), \quad P_{11}(z), \quad P_{12}(z), \ldots
\]
are the successive convergents of the continued fraction (12) (this method may be applied if all Hankel determinants \(H_k^n(k \leq n)\) are not zero; the method is numerically unstable).

0.2.2.
Since the 1960's the realization problem 0.1 has received a great deal of consideration in mathematical system theory. Particularly the names of Kalman, Ho, Youla, Silverman and Rissanen should be mentioned. There the problem may arise as follows.

A constant, linear, discrete-time system is a (physical) entity that accepts input and emits outputs. It is described by the (difference) equations
\[
\begin{align*}
\dot{x}(t+1) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t).
\end{align*}
\]
(13)

The parameter \(t\) symbolizes the time and assumes values from \(Z = \{\ldots, -1, 0, 1, 2, \ldots\}\).

For all \(t \in Z\), the vectors \(u(t), x(t)\) and \(y(t)\) are elements respectively from \(F^n, F^m\) and \(F^n\), where \(F\) is a field. These vectors are called input-vector, state-vector and output-vector.

\(A, B\) and \(C\) are (constant) matrices with elements from the field \(F\).

The equations (13) give a so-called internal description of the system. Via \(\zeta\)-transformation it is seen that the input-output behaviour of the system is described by
\[
\hat{y}(\zeta) = G(\zeta)B(\zeta)
\]
(14)

where \(\hat{y}(\zeta) = \sum_{i=-\infty}^{\infty} y(i)\zeta^{-i}\), \(G(\zeta) = \sum_{i=-\infty}^{\infty} u(i)\zeta^{-i}\) (formally) and where the transfer function \(G(\zeta)\) is defined by
\[
G(t) = C(tI - A)^{-1}B = \sum_{i=1}^{n} C A^{i-1} B e^{i t}.
\]

The equation (14) is a so-called \emph{external} description of the system.

The minimal realization problem 0.1 can now be reformulated as:

- given an external description of a system, to determine an internal description such that the order of the system (the dimension of the state vector \( x \)) is minimal.

Let \( t = t_0 \) an impulse be applied as input (so, if \( e_k \) denotes the \( k \)-th unit vector, \( u(t) = e_k \) with \( 1 \leq k \leq p \)) and let the system be relaxed at \( t = t_0 \) (so \( x(t) = 0 \)). Then the output is \( y(t) = \sum_{i=1}^{n} C A^{i-1} B e^{i t} \).

This is the reason that the sequence \( \{ S_i \}_{i=1}^{\infty} \) is called the impulse response. Performing a physical experiment, one may measure the impulse response at times \( t_0+1, t_0+2, \ldots \). Henceforth, we shall see that a complete minimal realization triple can be found if the first \( 2n \) elements of the impulse response are known.

The first algorithms for solving the minimal realization problem were independently proposed by Ho and Kalman [9], Silverman [23], and Youla and Tissi [25] in the mid 1960's. These algorithms have in common that a minimal realization triple is constructed from a decomposition of some Hankel block \( H_{n,k} \) that is associated with \( \{ S_i \}_{i=1}^{\infty} \):

\[
H_{n,k} = \begin{bmatrix}
S_1 & \cdots & S_k \\
\vdots & \ddots & \vdots \\
S_{k-1} & \cdots & S_{k+n-1}
\end{bmatrix}.
\]

It is a precondition that \( k \) and \( n \) should be large enough. In fact, for the realization algorithm to be possible, it is necessary and sufficient that

\[
k \geq \min(t \mid r(H_{n+1,k}) \neq r(H_{n,k})) \quad \text{for all } j \geq 1,
\]

\[
k \geq \min(t \mid r(H_{n+1,k}) \neq r(H_{n,k})) \quad \text{for all } i \geq 1,
\]

and that in (16) or (17) we have strict inequality (compare [24]).
If the impulse response \( \{6_r \}_{r=1}^\infty \) is realizable, then both minima exist. The minimum in (16) is called the observability index of the system. The minimum in (17) is called the controllability index of the system. In case \( p = q = 1 \), the observability index and the controllability index are both equal to the order of a minimal realization. Hence, it is then necessary and sufficient that \( k \geq n \), \( l \geq n \), and that \( k \) or \( l \) is greater than \( n \).

More recently, a new approach to the computation of a minimal realization was introduced by Rissanen [22]. His algorithm computes partial minimal realizations of successively growing, finite parts of the impulse response. The new realizations are obtained by updating the old ones (with few computations). Also, with this algorithm it is necessary that upper bounds for the observability index and the controllability index are a priori known, although for another reason: the finiteness of the computational process. In case \( p = q = 1 \), it is necessary that an upper bound \( N \) for \( n \) is known.

In fact, the algorithm of Rissanen is a method for recursively decomposing growing Hankel matrices \( H_{n,k} \). The decomposition of \( H_{n,k} \) has the property that a partial minimal realization of \( \{6_r \}_{r=1}^{k-1} \) can be directly obtained from it.

At the end of the computational process, \( k \) satisfies (16) with inequality and \( l \) satisfies (17) and then — as we shall see furtheron — a partial minimal realization is a complete minimal realization.

In our opinion the advantage of a recursive approach is that it seems to be more appropriate for solving the partial minimal realization problem (which will not be discussed in this thesis) and that it is suitable for finding approximate minimal realizations.

Pioneering the work of Rissanen, Massey gave an efficient recursive solution for the case \( p = q = 1 \) [163]. He uses an algorithm, which was developed by Berlekamp for the decoding of BCH codes [3]. This algorithm is not based on a Hankel matrix approach. In 1974, Dickinson, Morf and Kailath gave the generalization of the algorithm of Massey/Berlekamp to the multiple-input, multiple-output case [42].
0.3. Objectives

In case $F$ is an infinite field, the known recursive algorithms are numerically unstable in the sense that has become customary in numerical analysis. The inexact minimal realization, as it is supplied by any of these algorithms on a machine with finite arithmetic, might not be any "numerical neighbour" of the exact minimal realization and might not correspond with any "numerical neighbour" of the given impulse response.

The notions of numerical stability and numerical rank are discussed in the chapters 1 and 2. In chapter 3, we shall investigate the algorithms of Bissansen, Massey/Berlekamp and a new algorithm, which is somewhat faster. It will be shown that these algorithms are numerically unstable. A numerically stable, recursive, realization algorithm will be proposed in chapter 4. A serious drawback of this algorithm is that it requires $O(n^3)$ operations instead of the $O(n^2)$ operations, which for instance the algorithm of Bissansen needs. This, however, seems to be inevitable and is the price to be paid for numerical stability. Finally, in chapter 5 the minimal realization problem is considered in case one has only noisy impulse responses to work on.

In this thesis we consider only single-input, single-output systems. Although the multiple-input, multiple-output case is certainly not a trivial generalization, we believe that the discussion gives ample information how to attack the general case.

0.4. Fundamental notions and theorems in linear systems theory

In this section we shall prove a number of theorems (some of which are new), which will be useful in the following chapters.

We consider realizable impulse responses $\{s_i\}_{i=1}^\infty$ with elements $s_i \in \mathbb{R}$ ($i = 1, 2, \ldots$). We assume that at least one $s_i \neq 0$.

The infinite matrix $A$ defined by

\[(A)_{i,j} = s_{i-j-1}, \quad (i = 1, 2, \ldots; j = 1, 2, \ldots)\]

is called the Hankel matrix associated with $\{s_i\}_{i=1}^\infty$. 

21
A matrix $R_{k,j}$ defined by
\begin{equation}
(R_{k,j})_{i,j} := s_{i-k,j-1} \quad (i = 1, 2, \ldots, k; j = 1, 2, \ldots, )
\end{equation}
is called a finite Hankel matrix or Hankel block associated with $s_{j}^{k+2-1}$.
Note that there are $n$ different Hankel blocks associated with $s_{j}^{k+2-1}$.
The following theorem can also be found in Kalman [13], Silverman [24] and Rissanen [22], be it in various formulations.

**Theorem 1.**
Let $s_{j}^{m}$ be an impulse response, if
\begin{equation}
r(R_{n,j}) = r(R_{n+1,n+j}) = n \quad (j = 0, 1, \ldots)
\end{equation}
then there exists a vector $x$ such that
\begin{equation}
(x_{1}^{T}, 1) d_{n+1,n+j} = 0^{T} \quad (j = 0, 1, 2, \ldots)
\end{equation}
and
\begin{equation}
\begin{pmatrix}
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
-1 & \cdots & -1 & 0 & \cdots & 0 \\
\end{pmatrix}
\begin{pmatrix}
(s_{1}, s_{2}, \ldots, s_{n})^{T} \\
\vdots \\
\vdots \\
\vdots \\
\end{pmatrix}
\begin{pmatrix}
1 \\
0 \\
\vdots \\
-1 \\
\end{pmatrix}
= 0_{n}
\end{equation}
is a complete minimal realization of $s_{j}^{m}$ of order $n$.
Conversely, if $s_{j}^{m}$ admits a complete minimal realization of order $n$, then (20) holds.

**Proof.**
The first part of the proof is not difficult and therefore omitted.
Let $(A_{k}, b, c)_{n}$ be a complete minimal realization of $s_{j}^{m}$.
We define for $k \geq 1$
\begin{equation}
R_{k} :=
\begin{pmatrix}
\phi_{T} \\
\phi_{T}^{2} \\
\vdots \\
\phi_{T}^{k} \\
\end{pmatrix}
\text{ and } Q_{k} := (b_{1}^{T} A_{k}^{T} b_{1} \quad \ldots \quad A_{k}^{k-1} b_{1})_{T}.
\end{equation}
Because \( A \) is an \((n \times n)\) matrix, we have

\[
\tau(R_k) = \tau(Q_k), \quad \tau(Q_k) = \tau(\Omega_k) \quad \text{(} k = n, n-1, \ldots \text{)}.
\]

Furthermore, it is readily shown that

\[
H_{k,2} = R_k \Omega_k \quad \text{for any pair} \ (k,2) .
\]

Let \( H_{k+1,n} \) be the submatrix of \( H_{n+1,n} \) such that \( \tau(H_{k+1,n}) = \tau(H_{k,n}) = k \) \((k \leq n)\). Then there exist numbers \( a_1, \ldots, a_k \) such that

\[
\begin{bmatrix}
    s_{k+1} \\
    \vdots \\
    s_n \\
    s_{n+1} \\
    \vdots \\
    s_{n+k}
\end{bmatrix} = a_1 \begin{bmatrix}
    s_1 \\
    \vdots \\
    s_{k+1}
\end{bmatrix} + \cdots + a_k \begin{bmatrix}
    s_{k+1} \\
    \vdots \\
    s_{n+1}
\end{bmatrix}.
\]

From this it follows readily that \( \tau(H_{k,n}) = \tau(H_{k,n}) = k \).

Furthermore, we have \( H_{k+1,n} = R_{k+1} \Omega_n \) and \( H_{k+1,n+1} = R_{k+1} Q_{n+1} \) \((j = 0, 1, 2, \ldots)\). Because, when \( j \geq 0 \), \( \Omega_n \) and \( Q_{n+1} \) have the same rank and \( \Omega_n \) is a submatrix of \( Q_{n+1} \), we obtain \( \tau(H_{k+1,n}) = \tau(H_{k+1,n+1}) = k \) \((j = 0, 1, 2, \ldots)\).

Summarizing, we have:

\[
\tau(H_{k,2}) = \tau(H_{k+1,k+1}) = k \quad \text{(} j = 0, 1, 2, \ldots \text{)} .
\]

Applying the first part of the theorem it shows that \( \{s_j\}_{j=1}^m \) admits a minimal realization of order \( k \). Since \( k \leq n \) and it is given that \( (A, B, C) \) is a minimal realization, we must have \( k = n \).

It follows from theorem 1 that \( \tau(H_{k,2}) = n \) for all \( k \) and \( \Theta \).

Theorem 2 is due to Kalman [10].

Theorem 2:

Any complete minimal realization is unique up to a similarity transformation; if \((A, B, C)\) and \((\bar{A}, \bar{B}, \bar{C})\) are both minimal realizations for \( \{s_j\}_{j=1}^m \), then a regular matrix \( \Theta \) exists such that

\[
(A, B, C) = (\Theta \bar{A} \Theta^{-1}, \Theta \bar{B}, \Theta \bar{C}) .
\]
Proof.
Let $\mathbb{R}, Q_n$, and, similarly, $\tilde{R}, \tilde{Q}_n$, be defined by (23) for $k \geq 1$. We have
$$R_n \mathbb{R} = R_n Q_n = \tilde{R} \tilde{Q}_n.$$ 

$R_n$, $Q_n$, $\tilde{R}$, and $\tilde{Q}_n$ are $(n \times n)$ matrices. Because $R_n \mathbb{R}$ is an $(n \times n)$ matrix of rank $n$, also $R_n$, $Q_n$, $\tilde{R}$, and $\tilde{Q}_n$ have rank $n$. So, we may define:
$$T := R_n^{-1} \mathbb{R} = Q_n^{-1} \tilde{Q}_n^{-1}. $$

$(A, b, c)_n$ as well as $(\tilde{A}, \tilde{b}, \tilde{c})_n$ are realizations of $(s_i)_{i=1}^n$. Hence, we have
$$(s_1, \ldots, s_n) = c^T Q_n = \tilde{c}^T \tilde{Q}_n$$
and therefore
$$c^T = \tilde{c}^T \mathbb{T}^{-1}.$$ 

Similarly, we find
$$b = \mathbb{T} \tilde{b}.$$ 

Finally, we have
$$A = R_n^{-1} \mathbb{R} A Q_n Q_n^{-1} = R_n^{-1} \begin{bmatrix} s_1 & \cdots & s_n \end{bmatrix} = \tilde{Q}_n^{-1} = \tilde{Q}_n^{-1} \tilde{A} \tilde{Q}_n \tilde{Q}_n^{-1} = \tilde{T} \tilde{A} \tilde{T}^{-1}. $$

Remark.
From theorem 2 it follows that of any minimal realization triple $(A, b, c)_n$ of $(s_i)_{i=1}^n$ the realization matrix $A$ has a unique characteristic polynomial. In the following, we shall for brevity speak of the characteristic polynomial of $(s_i)_{i=1}^n$.

Theorem 3.
Let $(s_i)_{i=1}^n$ have a minimal realization of order $n$.
$$P(\lambda) = \lambda^n + \lambda^{n-1} + \cdots + \lambda_1$$ is the characteristic polynomial of $(s_i)_{i=1}^n$ if and only if
$$(\lambda_1, \ldots, \lambda_n) R_{n-1, n}^{-1} \mathbb{R} = 0^T \quad (j = 0, \lambda, \ldots). $$
Proof.

(22) is a minimal realization triple of \((s_i)_{i=1}^\infty\); the realization matrix has \(P(\lambda)\) as its characteristic polynomial.

Next, we shall prove some theorems concerning finite sequences \((s_i)_{i=1}^n\).

Theorem 6.

(i) Let \(m \geq 2n\).

\[
\{s_i\}_{i=1}^n \quad \text{has a partial minimal realization of order} \quad n \quad \text{if and only if}
\]

\[
(24) \quad r(H_{n+k,n+j}) = n \quad (j = 0, 1, \ldots, m-2n),
\]

(ii) Let \(m < 2n\).

\[
\{s_i\}_{i=1}^n \quad \text{has a partial minimal realization of order} \quad n \quad \text{if and only if}
\]

for any choice of \(s_i\) from \(n+1, n\),

\[
r(H_{n,k}) = r(H_{n+1,n}) = n.
\]

Proof.

(i) If \(\{s_i\}_{i=1}^n\) has a partial minimal realization of order \(n\), then \(\{s_i\}_{i=1}^m\) can be extended to \(\{s_i\}_{i=1}^n\) such that \(\{s_i\}_{i=1}^n\) has a complete minimal realization of order \(n\). Applying theorem 1, we obtain (24).

On the other hand, if (24) holds, then \(\{s_i\}_{i=1}^n\) can be extended to \(\{s_i\}_{i=1}^m\) such that (20) holds. Applying theorem 1 shows that \(\{s_i\}_{i=1}^m\) has a realization of order \(n\). This realization should be minimal, since otherwise one would have a conflict with (24).

(ii) Let \(\{s_i\}_{i=1}^n\) have a minimal realization of order \(n\). Because \(\{s_i\}_{i=1}^m\) has a realization of order \(m\) (if \(\{s_i\}_{i=1}^m\) is extended with merely zeros, then \(r(H_{n,m}) = r(H_{n+1,m+1}) = m\) for all \(j \geq 0\)), we have \(m \geq n\).

So we may consider the Hankel block

\[
H_{n,n+1,n} = \begin{bmatrix}
2 & \cdots & n \\
\vdots & \ddots & \vdots \\
0 & \cdots & 2n
\end{bmatrix}
\]

\[
\{s_i\}_{i=1}^n \quad \text{has an extension to} \quad \{s_i\}_{i=1}^m \quad \text{such that} \quad \{s_i\}_{i=1}^n \quad \text{has a minimal realization of order} \quad n. \text{ Theorem 1 shows that} \quad r(H_{n,n}) = n \quad \text{and, hence,}
\]

\[
r(H_{n+1,n+1}, n) = n - n + 1.
\]
Let \( s_{n+1} \) be arbitrary. If \( m - n + 1 = n \) then \( r(H_{m-n+1,n}) = n \). If \( m - c + 1 < n \) and \( r(H_{m-n+2,n}) = m - n + 1 \), then there exist constants \( a_1, \ldots, a_{m-n+1} \) such that
\[
\begin{bmatrix}
  s_{m-n+2} \\
  \vdots \\
  s_{n+1}
\end{bmatrix} = s_1 \begin{bmatrix} 1 \end{bmatrix} + \cdots + a_{m-n+1} \begin{bmatrix} s_{m-n+1} \end{bmatrix}.
\]

This shows, as \( m-n+2 \leq n \), that the last column of \( H_{m-n+1,n} \) is linearly dependent on the foregoing columns of \( H_{m-n+1,n} \). However, in that case it is possible to extend \( \{ s_i \}_{i=1}^m \) such that \( r(H_{n,n}) = r(H_{m-n+1,n}) \) for all \( j > 0 \), which implies that \( \{ s_i \}_{i=1}^m \) should have a minimal realization of order less than \( n \). So, we have \( r(H_{m-n+2,n}) = m - n + 2 \).

Continuing like this, it follows that for any choice of \( \{ s_i \}_{i=1}^{2n} \)
\[
r(H_{n,n}) = r(H_{n+1,n}) = n.
\]
The converse is similarly proved as in case (i).

Theorem 5.
If \((A, b, c)^n\) is a partial minimal realization of \( \{ s_i \}_{i=1}^m \), then also
\[
(TA^{-1}, Tb, T^Tc)^n \quad \text{for all regular } T.
\]

If \( m < 2n \), then any partial minimal realization is unique up to a similarity transformation.

Proof.
Let \( T \) be a regular matrix. For all \( i > 0 \) we have
\[
s_i = c T^{i-1} b = c T^{-1} (TA^{-1})^{i-1} Tb,
\]
which proves the first part. The second part is proved as in theorem 2.

The following three theorems are stated without proof.

Theorem 6.
Let \( \{ s_i \}_{i=1}^m \) have a complete minimal realization \((A, b, c)^n\) and let \( \{ s_i \}_{i=1}^m \) have a partial minimal realization \((A, b, c)^n\). If \( m < 2n \), then \((A, b, c)^n\) is similar to \((A, b, c)^n\).
Theorem 7.
Let \( \{a_i\}_{i=1}^n \) have a partial minimal realization of order \( n \) and let \( m \geq n \).
Then \( F(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1 \) is the characteristic polynomial of a partial minimal realization of \( \{a_i\}_{i=1}^n \) if and only if
\[
(\pi_1, \ldots, \pi_n, 1) R_{n+1,a} = \mathbf{0}^T \quad (t = 1, 2, \ldots, n-1).
\]

Theorem 8.
Let \( F(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1 \) be the characteristic polynomial of a partial minimal realization of \( \{a_i\}_{i=1}^n \).
A minimal realization of \( \{a_i\}_{i=1}^n \) is given by (22).

Theorem 9.
Let \( \{a_i\}_{i=1}^n \) have a minimal realization of order \( n \).
If for some pair \((k, \ell)\) it holds that
(i) \( r(H_{k, \ell}) = r(H_{k+1, \ell}) = k \)
then \( r(H_{k, \ell}) = k \) and \( n \geq k \).

(ii) \( r(H_{k, \ell}) = r(H_{k+1, \ell}) = k \)
and
\( r(H_{k+1, \ell+1}) = k + 1 \)
then \( r(H_{k+1, \ell+1}) = k + 1 \) and \( n \geq \ell + 1 \),

(iii) \( r(H_{k, \ell}) < r(H_{k, \ell+1}) \)
then \( n > k \).

Proof.
(i) Let \((\pi_1, \ldots, \pi_n, 1)\) be the unique vector such that \((\pi_1, \ldots, \pi_n, 1) R_{k+1,a} = \mathbf{0}^T \).
Then (22) is a partial realization for \( \{a_i\}_{i=1}^k \) of order \( k \).

Let \( k' \leq k \) be the order of a partial minimal realization.
According to theorem 4 we must have
\[
r(H_{k', \ell}) = k' \quad \text{and} \quad r(H_{k'+1, \ell}) = r(H_{k'+1, \ell+1}) = k'.
\]
The last relation shows that \( k' = k \). Hence we have \( r(H_{k, k}) = k \).
(ii) Let \( k+1 \leq s \leq l+1 \). We consider the \( r \)-th row of \( H_{k+1,l+1} \) \((s_1, \ldots, s_{l+2})^T \).

If this row is linearly dependent on the foregoing rows of \( H_{k+1,l+1} \), then, as \( s \leq l+1 \), the last column of \( H_{k+1,l+1} \) \((s_1, \ldots, s_{l+2})^T \) is linearly dependent on the foregoing columns of \( H_{k+1,l+1} \), which is not so.

Therefore, we must have \( r(H_{k+1,l+1}) = 2 \cdot 1 \).

(iii) If \( n < k \) then we should have \( r(H_{k+1,2}) \) that

\[ r(H_{k+1,2}) = r(H_{k+1,2}) \]

This is not so, consequently, we have \( n > k \).

Theorem 10.
If \( \{s_i\}_{i=1}^m \) contains a non-zero element, then the following algorithm determines the order of a partial minimal realization of \( \{s_i\}_{i=1}^m \):

\begin{verbatim}
1 \( k := 1 \); while \( s_k = 0 \) do \( \ell := k + 1 \);
2 \( k := \ell \);
3 while \( k+1 < m \) do if \( r(H_{k+1,k+1}) = k+1 \)
4 then begin \( k := \ell + 1 \); \( \ell := \ell + 1 \) end
5 else \( \ell := \ell + 1 \);
6 \( n := k \);
\end{verbatim}

Proof.
After execution of lines 1 and 2, we have \( k = \ell \) and \( s_1 = s_2 = \ldots = s_{\ell-1} = 0 \), but \( s_\ell \neq 0 \). So

\[
H_{\ell+1,\ell} = H_{\ell+1,\ell} = \begin{pmatrix}
0 & s_1 \\
\vdots & \ddots \\
0 & \ldots & s_\ell \\
\end{pmatrix}
\]

with \( s_\ell \neq 0 \).

Consequently, we then have

\[
\begin{align*}
\text{(25)} & \quad r(H_{\ell+1,\ell}) = r(H_{\ell+1,\ell}) = k.
\end{align*}
\]
If \( k + 4 \geq m \), then, obviously, (25) still holds just before line 6 is executed.
If \( k + 4 < m \), then, if \( r(H) \sim 1 \), again that (25) holds; otherwise, if \( r(H) \sim k - 1 \), then we have after \( k := k + 1 \) and \( k := 1 \) again that (25) holds as a consequence of theorem 9 part (ii).

It is obvious that the algorithm is finite.

Consequently, we have at the end of the computational process:

\[
k = m, \quad r(H) = r(H + 1) = k
\]

for any choice of \( \{s_i\}_{i=1}^{k+4} \).

Applying theorem 4, it shows that \( \{s_i\}_{i=1}^{n} \) has a partial minimal realization of order \( k \).

The first and second part of theorem 9 are new, as well as theorem 10.

The following theorem is due to Hassey [16].

**Theorem 11.**

Let the degree \( q \) of the characteristic polynomial \( P_n(\lambda) \) of \( \{s_i\}_{i=1}^{n} \) be non-zero. We have

(i) if \( n < 2q \), then degree \( (P_j(\lambda)) = q \) for \( n \leq j \leq 2q \),

(ii) if \( n \geq 2q \), then either degree \( (P_{m+1}(\lambda)) = q \) and \( P_{\lambda+1}(\lambda) = P_{2q}(\lambda) \)

or degree \( (P_{m+1}(\lambda)) = m - q + 1 \).

**Proof.**

The degree of \( P_n(\lambda) \) is the order of a minimal realization of \( \{s_i\}_{i=1}^{n} \).

If \( n < 2q \), then we have for any choice of \( s_{n+1}, \ldots, s_{2q} \) (theorem 4)

\[
r(H) = r(H + 1) = q.
\]

This implies that \( \{s_i\}_{i=1}^{n} \) has a minimal realization of order \( q \).

If \( n \geq 2q \), then \( \{s_i\}_{i=1}^{n} \) has a unique minimal realization of order \( q \) (theorems 3). This implies that \( P_n(\lambda) = P_{2q}(\lambda) \).

If \( P_{m+1}(\lambda) = P_{m+1}(\lambda) \), then, obviously, degree \( (P_{m+1}(\lambda)) = q \) and \( P_{m+1}(\lambda) = P_{2q}(\lambda) \).

If \( P_{m+1}(\lambda) \neq P_{m+1}(\lambda) \), then we have, according to theorem 4,

\[
r(H) = r(H + 1, n - q) = q.
\]
but
\[ r(q^*; n+1-i) = q + 1. \]

Consequently, \( r(M_{n+1-i}^{q}; n+1-i-q) = m+1-q \) (theorem 9 part (ii)). This implies that degree \( (P_{m+1}(\lambda)) = m+1-q \) (theorem 4). \( \square \)

**Corollary.**
If \( P_{m+1}(\lambda) \neq P_n(\lambda) \) then degree \( (P_{m+1}(\lambda)) = \max(q, m+q+1) \). \( \square \)

**Theorem 12.**
If \( m > \text{degree}(P_n(\lambda)) > 1 \), then we have \( k > \text{degree}(P_k(\lambda)) \) for all \( k \geq n \).

**Proof.**
Applying the corollary of theorem 11, it follows that
\[ \text{degree}(P_{m+1}(\lambda)) = \max(\text{degree}(P_n(\lambda)), m+1-\text{degree}(P_n(\lambda))) < m+1. \] \( \square \)

In conclusion to this chapter we introduce norms for Hankel blocks.

Given a pair of natural numbers \((k, l)\) one readily verifies that
\[ H_{k, l} := \{ H_{k, l}^{s} \mid H_{k, l}^{s} \text{ is a Hankel block} \} \]
is a linear vector space over \( \mathbb{R} \). In this vector space norms are defined by
\begin{equation}
\|H_{k, l}\|_{2} := \left( \sum_{i=1}^{k+l-1} s_{i}^{2} \right)^{\frac{1}{2}}
\end{equation}

and
\begin{equation}
\|H_{k, l}\|_{\infty} := \max_{1 \leq k+l-1} \{|s_{i}|\}. \tag{27}
\end{equation}

Any usual matrix norm is also a norm in \( H_{k, l} \).

The following relations exist between \( \| \cdot \|_{2}, \| \cdot \|_{E} \) and \( \| \cdot \|_{\infty} \).

If \( \alpha := \min(k, l) \), then
\begin{equation}
\alpha^{-1} \|H_{k, l}\|_{2} \leq \|H_{k, l}\|_{E} \leq \|H_{k, l}\|_{\infty} \leq \alpha \|H_{k, l}\|_{2} \tag{28}.
\end{equation}

Because
\[ \|H_{k, l}\|_{2} = \|H_{k, l}\|_{E} \leq \alpha \|H_{k, l}\|_{2}, \]

30
it follows that

$$a^{-1/2} \| H_0, u_s \|_2 \leq \| H_0, z \|_2 \leq a^{1/2} \| H_0, 0 \|_2.$$  

The following relation exists between \( \| H_{0,s} \|_2 \) and \( \| H_{0,c} \|_2 \):

$$\| H_{0,s} \|_2 \leq \| H_{0,c} \|_2 \leq \| H_{0,s} \|_2.$$


1. PRELIMINARY ON NUMERICAL STABILITY

1.1. Definition of numerical stability

A numerical problem is characterized by a set of possible inputs (the input set), a set of possible outputs (the output set) and a mapping from the input set into the output set. The input set as well as the output set is a multidimensional set of real numbers. We consider numerical problems for which the mapping is unique, in other words, numerical problems with a unique solution. We consider digital computers with floating point arithmetic and relative machine precision $\eta$. We always assume that $\eta > 0$.

Dealing with a numerical problem and an algorithm for solving it, we shall use the following notation:

**Notation 1.**

$\mathcal{D}$ denotes the input set.

$\mathcal{R}$ denotes the output set.

The exact solutions are obtained by applying the mapping

$f: \mathcal{D} \to \mathcal{R}$.

The implementation of the algorithm on some machine is the mapping

$f_\text{approx}: \mathcal{D} \to \mathcal{R}$.

Let $d \in \mathcal{D}$. $f_\text{approx}(d)$ is a function of the relative machine precision $\eta$. For example, $f_\text{approx}(a+b) = (a+b)(1+\epsilon_1)$, $f_\text{approx}(a-b) = ab(1+\epsilon_2)$ with $|\epsilon_i| \leq \eta$ for $i = 1, 2$. The values of $\epsilon_1$ and $\epsilon_2$ depend on the arithmetic of the machine and on the values of $a$ and $b$. However, it is not unrealistic to assume that values of $|\epsilon_1|$ and $|\epsilon_2|$ that are greater than $\eta$ are as frequent as values of $|\epsilon_1|$ and $|\epsilon_2|$ that are smaller than $\eta$. See further Wilkinson [9, p.112].

We assume that an essential property of an algorithm is that for all $d \in \mathcal{D}$ no breakdown in the computational process occurs (i.e., division by zero, taking the square root of a negative number and so on), provided that all computations are performed with a relative precision $\eta < \eta_0(d)$.

Furthermore, we assume that in $\mathcal{D}$ a metric $\mu(d_1, d_2)$ can be defined. A metric $\mu(d_1, d_2)$ is a mapping from $\mathcal{D} \times \mathcal{D}$ into $\mathbb{R}$ that has the following properties:

For all $d_1 \in \mathcal{D}$, all $d_2 \in \mathcal{D}$ and all $d_3 \in \mathcal{D}$,
\[ i) \quad \mu(d_1, d_2) = \mu(d_2, d_1); \\
ii) \quad d_1 \neq d_2 \Rightarrow \mu(d_1, d_2) > 0, \quad d_1 = d_2 \Rightarrow \mu(d_1, d_2) = 0; \\
iii) \quad \mu(d_1, d_2) \leq \mu(d_1, d_3) + \mu(d_3, d_2). \]

Let the input that consists entirely of zeroes be symbolized by 0. We shall assume that, if 0 \( \notin \mathcal{D} \), the metric \( \mu(d_1, d_2) \) can be extended to a metric in \( \mathcal{D} \cup \{0\} \).

Next, let us introduce in \( \mathcal{D} \) the notion of \( \mathcal{N}_n \)-neighbourhood or numerical neighbourhood.

**Definition 2.**

Let \( d \in \mathcal{D} \setminus \{0\} \) and let \( C > 0 \),

\[ \mathcal{N}(d, C) := \{ x \mid x \in \mathcal{D}, \mu(d, x) < C\mu(d, 0) \} \]

is called a \( \mathcal{N}_n \)-neighbourhood or numerical neighbourhood of \( d \).

If \( x \in \mathcal{N}(d, C) \), then we shall say that \( x \) is a \( \mathcal{N}_n \)-neighbour or a numerical neighbour of \( d \).

We shall assume that also in \( \mathbb{R} \cup \{0\} \) a metric \( \mu'(r_1, r_2) \), \( r_i \in \mathbb{R} \) \((i = 1, 2)\) is defined and that in \( \mathbb{R} \) similarly as in \( \mathcal{D} \) the notion of numerical neighbourhood: \( \mathcal{N}'(r, C) \), \( r \in \mathbb{R} \setminus \{0\} \) is introduced.

If we apply the notation and notions introduced so far to the numerical problem of a \((k \times k)\) system of linear equations \( Ax = b \), the input set is

\[ \mathcal{D} = \mathcal{N}_{A, b}^c (\mathbb{R}) \times \mathbb{R}^k \]

with metric

\[ \mu(d_1, d_2) = \| (A_1 \mid b_1) - (A_2 \mid b_2) \| \] where \( d_1 = (A_1, b_1) \in \mathcal{D} \) \((i = 1, 2)\).

Here is \( \| \cdot \| \) a submultiplicative matrix norm. Furthermore, the output set is

\[ \mathbb{R} = \mathbb{R}^k \]

with metric

\[ \mu'(r_1, r_2) = \| r_1 - r_2 \| \] where \( r_1 \in \mathbb{R} \) \((i = 1, 2)\).

Here is \( \| \cdot \| \) a usual vector norm. The mapping
\( f: \mathcal{D} \rightarrow \mathbb{R} \)

is defined by
\[
f = \bigvee_{(a,b) \in \mathcal{D}} a^T b.
\]

Finally, a numerical neighbourhood of \((a_0, b_0) \in \mathcal{D}\) is
\[
\mathcal{N}(a_0, b_0) = \{(a,b) \in \mathcal{D}; \frac{\|a - b\|}{\|a_0 - b_0\|} < C_n\}
\]
and a numerical neighbourhood of \(r_0 \in \mathcal{D}\) is
\[
\mathcal{N}(r_0) = \{x \in \mathbb{R}; \frac{\|x - r_0\|}{\|r_0\|} < C_n\}.
\]

**Definition 3.**
An algorithm is said to be
backward stable on \(\mathcal{D}\) if
\[
1. \quad \exists C_1 > 0 \forall e \in \mathbb{R} \forall n \in \mathbb{N} \exists d \in \mathcal{D}(d, C_1 n) \left[ \|\mathbf{f}(d) - \mathbf{f}(d')\| < C_1\| e \| \right],
\]
forward stable on \(\mathcal{D}\) if
\[
2. \quad \exists C_2 > 0 \forall e \in \mathbb{R} \forall n \in \mathbb{N} \exists d \in \mathcal{D}(d, C_2 n) \left[ \|\mathbf{f}(d) - \mathbf{f}(d')\| < C_2\| e \| \right],
\]
numerically stable on \(\mathcal{D}\) if it is backward or forward stable on \(\mathcal{D}\),
numerically unstable on \(\mathcal{D}\) if it is not numerically stable on \(\mathcal{D}\).

If an algorithm is backward stable on \(\mathcal{D}\), then the output \(\mathbf{f}(d)\), obtained on some machine with \(n < n_0\) (\(n_0\) depends on \(d \in \mathcal{D}\)), corresponds exactly to a \(C_1 n\)-neighbour of the input \(d\), with \(C_1\) independent of \(n\) and independent of the individual input element of \(\mathcal{D}\). So, if (1) holds, then we can obtain an output that corresponds exactly with an input \(d'\) as close as we like to the input \(d\) by increasing the relative machine precision (making \(n\) smaller).

If an algorithm is forward stable on \(\mathcal{D}\), then the output \(\mathbf{f}(d)\), obtained on some machine with \(n < n_0\) (\(n_0\) depends on \(d \in \mathcal{D}\)), is a \(C_2 n\)-neighbour of the exact output, with \(C_2\) independent of \(n\) and independent of the individual input element of \(\mathcal{D}\). So, if (2) holds, then we can approximate the exact output as close as we like by increasing the machine precision.
If we have two forward stable algorithms for the same problem, then we shall say that the algorithm that admits the smaller $C_2$ is more stable than the other; similarly for two backward stable algorithms.

We cannot compare a backward stable algorithm with a forward stable algorithm.

The logical negation of numerical stability on $\mathbb{D}$ is:

$$(3) \quad \forall \varepsilon > 0 \quad \exists \delta > 0 \quad \forall \eta > 0 \quad \exists y \in \Omega(d, \varepsilon, \eta) \left| f(y)(d) \neq f'(y) \right|$$

and

$$(4) \quad \forall \varepsilon > 0 \quad \exists \delta > 0 \quad \forall \eta > 0 \quad \exists y \in \Omega(d, \varepsilon, \eta) \left| f(y)(d) \neq f'(y) \right|$$

In order to prove that an algorithm is numerically stable on $\mathbb{D}$, it is sufficient to prove either (1) or (2). In order to prove that an algorithm is numerically unstable on $\mathbb{D}$ we must prove (3) as well as (4).

The reason that (1) as well as (2) is used as a criterion for numerical stability, is that it is possible that, for instance, the obtained output, even though it is a numerical neighbour of the exact output, does not correspond with any possible input; it is also possible that - due to the (forward) instability inherent to the numerical problem itself - (1) holds, but (2) is not.

In section 1.1 we shall give examples that (1) holds, but (2) does not, and that (2) holds, but (1) does not.

Let us compare the concept of numerical stability as it is defined here with the concept as it is defined in [2], Babuška et al. [1] and Stru [6].

In [2] the notion of a "guterig" Roehamprozess is introduced: a computational process is "guterig" for a set of possible inputs, if - independent of the individual input - round-off errors of the order of $\varepsilon$ in the computations and round-off errors of the order of $\eta$ in the input data have a comparable effect on the output data. Consequently, in a "guterig" computational process, the sensitivity of the exact solution to errors in the input data is comparable with the sensitivity of the numerical solution to errors in the computations.

Babuška's concept of numerical stability, in short, is the following: an algorithm for a numerical problem is numerically stable, if - independent of
the individual input data - round-off errors of the order of \( n \) in the computations have an effect on the output data that is bounded.

It is quite clear that - if an algorithm is backward stable - the computational process governed by the algorithm is gutartig in the sense of Bauer; it is also clear - if an algorithm is forward stable - that the algorithm is numerically stable in the sense of Babuška. Under certain conditions on \( \hat{f} \), the converse is also true: a gutartig computational process is governed by a backward stable algorithm; an algorithm that is numerically stable in the sense of Babuška, is forward stable.

The concept of numerical stability of definition 3 unifies the merits of the concepts of numerical stability of Bauer and Babuška.

Finally, let us compare our concept of numerical stability with the concept of Steer [6].

Steer argues that, independent of the particular algorithm, a round-off error of the order of \( n \) in the input data and a round-off error of the order of \( n \) in the output data is inevitable. Consequently, the best one can expect is that \( \epsilon(\hat{f}(d)) \lesssim \epsilon(\hat{f}(d'),r) \), with \( d' \in B(d,n) \). If the round-off errors of the order of \( n \) have an effect on the final solution that is comparable with the effect of the inevitable errors (they are "harmles") then - following Bauer - Steer says that the computational process is "gutartig". Formally, this would lead to the following definition:

An algorithm is numerically stable on \( D \), if

\[
(5) \quad \exists_{C_1>0} \exists_{C_2>0} \forall_{\epsilon>0} \exists_{n>0} \forall_{d \in D} \exists_{d' \in B(d,n)} \left[ \epsilon(\hat{f}(d)) \lesssim \epsilon(\hat{f}(d'),n) \right] .
\]

One readily verifies that \((5)\) is implied by either \((1)\) or \((2)\), but not conversely, unless the mapping \( \hat{f} \) satisfies certain conditions (which are normally satisfied).

We would prefer \((5)\) as definition of numerical stability, were it not that the use of \((5)\) may lead to rather complicated and tedious examinations, if one tries to prove that an algorithm is numerically unstable on \( D \).
1.2. Techniques for proving or disproving numerical stability

The natural way to show that (1) holds for an algorithm is by backward error analysis. With backward error analysis one shows that the obtained output corresponds exactly with a slightly perturbed input. Backward error analysis has firstly been introduced by Von Neumann and Goldstine in 1947 [4], although presented as a forward error analysis, in 1954 been explicitly named and used by Givens [3] and, thereafter, intensively been exploited and propagated in many publications by Wilkinson.

The natural way to show that (2) holds is by forward error analysis. With forward error analysis one shows that the obtained output is a slightly perturbed exact output.

We shall give a few examples of how to work with the concept of numerical stability of definition 3.

1. We consider the computation of $1 + \frac{1}{x-1}$, $x > 1$.

The input space is $\mathcal{D} := \{x \mid x \in \mathbb{R}, x > 1\}$.

The output space is $\mathcal{R} := \{y \mid y \in \mathbb{R}, y > 0\}$.

For the metric in $\mathcal{D} \cup \{0\}$ and $\mathcal{R} \cup \{0\}$ we take the usual metric in $\mathbb{R}$.

The mapping $f$ from $\mathcal{D}$ into $\mathcal{R}$ is defined by $f(x) = 1 + \frac{1}{x-1}$, $x \in \mathcal{D}$.

Let $d \in \mathcal{D}$. If $\eta$ is chosen small enough, then the algorithm can be applied to $d \in \mathcal{D}$ without a breakdown (division by 0) in the computational process. It then holds that

$$ E(d)(x) = \left(1 + \frac{1}{x-1}\right)\left(1 + \epsilon_1\right)\left(1 + \epsilon_2\right) $$

with $|\epsilon_1| \leq 2\eta$ and $|\epsilon_2| \leq \eta$.

Firstly, we give a backward error analysis; let $f(y) = f(t)(x)$. One readily verifies that

$$ y = \frac{x-1}{(1 + \epsilon_1)(1 + \epsilon_2) + \epsilon_2(x-1)} - 1 $$

and that

$$ \frac{|y - x|}{|x|} = \frac{|x-1|}{x} \left(\epsilon_1 + \epsilon_2 + \epsilon_2 \epsilon_2 + \epsilon_2(x-1)\right). $$

From definition 3, it follows that – if the algorithm is backward stable – we should have that

30
for all \( n \) smaller than some \( n_0(x) \), with \( c_1 \) independent of \( x \) and \( n \).

We have

\[
\frac{|x - x_1|}{|x|} \leq c_1 \left( \frac{\varepsilon_1 \cdot \varepsilon_2 \cdot \varepsilon_2}{x \cdot (1 + \varepsilon_1)(1 + \varepsilon_2) + \varepsilon_2^2 (x - 1)} \right).
\]

The factor \( \frac{x - 1}{x} \) at the right-hand side can be uniformly bounded on \( \mathcal{D} \).

The numerator of the second factor gives no trouble; if \( n_0 \) is chosen small enough, it is greater than \( \frac{1}{2} \) (say). However, the term \( \frac{\varepsilon_2^2}{n_0} \) of the numerator causes trouble. The coefficient \( \frac{\varepsilon_2^2}{n_0} \) lies somewhere in the interval \([-1, 1]\), which implies that unless \( \frac{\varepsilon_2^2}{n_0} = O(x^{-1}) \), \( x = \infty \) no upper bound for the numerator can be given that is independent of \( x \). Since \( x = \infty \) it is as probable that \( \frac{\varepsilon_2}{n_0} \approx \frac{\varepsilon_3}{n_0} \) as that \( \frac{\varepsilon_2}{n_0} < \frac{1}{2} \), \( \frac{\varepsilon_2^2}{n_0} \neq O(x^{-1}) \) if \( x = \infty \) and, hence, the algorithm is not backward stable on \( \mathcal{D} \).

A forward error analysis,

\[
\frac{\|k(\delta)(y) - f(x)\|}{\|f(x)\|} \leq \frac{\left| 1 + \frac{\varepsilon_1 \cdot \varepsilon_2 \cdot \varepsilon_2}{(1 + \varepsilon_1)(1 + \varepsilon_2) - (1 + \varepsilon_2^2)(x - 1)} \right|}{1 + \frac{\varepsilon_1 \cdot \varepsilon_1 \cdot \varepsilon_2}{1 + \varepsilon_2^2}} \leq \frac{\varepsilon_1 \cdot \varepsilon_2 \cdot \varepsilon_1 \cdot \varepsilon_2}{1 + \varepsilon_2^2} \leq 5\eta.
\]

So, the algorithm is forward stable.

According to definition 3, the algorithm is numerically stable on \( \mathcal{D} \). We gave here an example of an algorithm that is forward stable, but not backward stable. If we restrict \( \mathcal{D} \) to the subset \( \mathcal{D}(M) := \{ x \in \mathcal{D} \mid x \leq M \} \), \( M > 0 \), then on \( \mathcal{D}(M) \) the algorithm is backward stable as well.

If we consider a \((k \times k)\) system of linear equations \( Ax = b \),

\[
\mathcal{D} = \text{span}(\mathcal{W}) \times \mathbb{R}^k; \quad R = \mathbb{R}^k; \quad f = v \cdot (A \delta x)_{\mathcal{D}} A^{-1} b,
\]

For the metric in \( \mathcal{D} \) we take
\[(A_i; b_i) - (A_2; b_2)\|_\infty \text{ where } (A_i, b_i) \in \mathcal{D} (i = 1, 2).\]

And for the matrix in \( \mathbb{R} \) we take
\[\|r_i\|_\infty \text{ where } r_i \in \mathbb{R} (i = 1, 2).\]

An algorithm for solving the system is the following:

1. Make a decomposition for \( A \) with Gaussian elimination and partial pivoting:
   \[PA = LU,\]
   with \( P \) a permutation matrix, \( L \) unit lower triangular and \( U \) upper triangular.
2. Solve the triangular system \( Ly = Pb.\)
3. Solve the triangular system \( Ux = y.\)

Let \( (A, b) \in \mathcal{D}. \) It can be shown that, if the algorithm is applied to \( (A, b) \in \mathcal{D}\) on some machine, the computational process does not break down provided that the relative machine precision \( \eta \leq \eta_0(A, b).\)

In the following, \( L, U, y \) and \( x \) denote the \( L, U, y \) and \( x \) as they are computed by the machine.

We shall assume that \( \eta_0 \) is so small that we may account for higher order terms in \( \eta \) by multiplying the first order terms by \( 1.1. \) We have (Wilkinson [8, 9]):

1. \[PA = P_{LU} = LU \text{ with } \|E_{LU}\|_\infty \leq 1.1 k|A| |U| \eta ;\]
2. \[(L + \delta L)y = Pb \text{ with } |\delta L| \leq 1.1 k|L| \eta ;\]
3. \[(U + \delta U)x = y \text{ with } |\delta U| \leq 1.1 k|U| \eta .\]

It follows, formally, that \( x = (A + E)^{-1}b = f(A + E, b) \) with
\[E = P^{-1}(E_{LU} + \delta L + L\delta U + \delta LU)\]
and
\[\frac{\|E(0)\|_\infty}{\|A(0)\|_\infty} \leq 1.4 k \eta \frac{\|L\|_\infty \|U\|_\infty}{\|A\|_\infty}.\]

The decomposition for \( A \) is obtained with partial pivoting, so \( IL_i \leq k \) and
\[\|U\|_\infty \approx 2^{k-1} \eta \] (Wilkinson [8, 9]).
Hence,

\[
\frac{\|E(\hat{\mathbf{a}})\|_2}{\|\hat{\mathbf{a}} - \mathbf{a}\|_2} \leq 3.4 \cdot 2^{k-h-1} \eta_0.
\]

If \( \eta_0 \) is chosen so small that \( 3.4 \cdot 2^{k-h-1} \eta_0 < 1 \), then, indeed, \((A + E)^{-1}\) exists. Applying definition \( \delta \), with \( C_1 = 3.4 \cdot 2^{k-h-1} \), it follows that the algorithm is backward stable on \( D \).

It can be shown that the algorithm is not forward stable on \( D \).  

Next, we shall give two examples of numerically unstable algorithms. The proofs that they are unstable are founded upon the assumption that different round-off errors are independent of each other and have no relation to the data in which the errors are committed. For instance, if a matrix \( A \) is rounded to a matrix \( \tilde{A} \) of machine numbers, this assumption means that the elements of \( \delta A = \tilde{A} - A \) have no relation to each other and that \( \delta A \) has no other relation to \( A \) than \( \|\delta A\| \leq \eta_0 \|A\| \).

With these assumptions it is legitimate to say that an algorithm is numerically unstable, if it can be shown that one single, artificial round-off error in the computations may have the effect that the solution is not a numerical neighbour of the exact solution and, simultaneously, the solution does not correspond exactly with a numerical neighbour of the input.

III. We consider again a system of linear equations \( A\mathbf{x} = \mathbf{b} \) and, essentially, the same algorithm as in II for solving it, except that now we assume that during the construction of the decomposition no pivoting technique whatsoever is employed. In order that this algorithm is possible, we restrict \( D \) to the subset

\[
D' := \{ (A, \mathbf{b}) \mid (A, \mathbf{b}) \in D; \text{ all principal minors of } A \text{ are } \neq 0 \}.
\]

In order that the algorithm is numerically stable on \( D' \) we either must prove that it is backward stable on \( D' \):

\[
\exists C_0 > 0 \quad \forall \mathbf{d} \in D' \quad \exists \eta_0 > 0 \quad \exists \mathbf{d}' \in D'(d, C_0) \quad \left( \|E(\mathbf{f})(\mathbf{d}) = \mathbf{f}(\mathbf{d}')\| \geq C_1 \eta_0 \right),
\]

which is equivalent with:

\[
\exists C_0 > 0 \quad \forall \mathbf{d} \in D' \quad \exists \eta_0 > 0 \quad \left( \min_{\mathbf{d}'} \frac{\|\mathbf{d} - \mathbf{d}'\|}{\|\mathbf{d}\|} \|E(\mathbf{f})(\mathbf{d}) = \mathbf{f}(\mathbf{d}')\| < C_1 \eta_0 \right).
\]
or, we must prove that it is forward stable on $\mathbb{D}$:

\[(8) \quad \exists \omega > 0 \quad \forall x \in \mathbb{D}^n \quad \exists \varepsilon > 0 \quad \forall \varepsilon < \varepsilon_0 \left( \frac{\|f(x) - f(x)\|}{\|f(x)\|} < C_2 \varepsilon \right) \]

We consider the system

\[
\begin{pmatrix}
1 & 0 \\
1 & 1
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
= 
\begin{pmatrix}
1 \\
0
\end{pmatrix}

\text{with } |c| \ll 1.
\]

With exact arithmetic we have

\[
L = \begin{pmatrix}
1 & 0 \\
\varepsilon & 1
\end{pmatrix}, 
U = \begin{pmatrix}
\varepsilon & 1 \\
0 & 1 - \varepsilon
\end{pmatrix}, 
y = \begin{pmatrix}
1 \\
-\varepsilon
\end{pmatrix},
\]

\[
x = \begin{pmatrix}
\varepsilon^{-1}(1 + \varepsilon u_{22})^{-1} \\
-(\varepsilon u_{22})^{-1}
\end{pmatrix} = \begin{pmatrix}
(1 - \varepsilon)^{-1} \\
-(1 - \varepsilon)^{-1}
\end{pmatrix}.
\]

Let us investigate the effect of one single, artificial round-off error in $u_{22}$:

\[
\bar{u}_{22} = u_{22}(1 + \eta), \quad 0 < \eta < 1.
\]

We assume that $\eta$ is so small that the algorithm can be applied to this system of equations on a machine with relative precision $\eta$, without a breakdown in the computational process.

We have

\[
\tilde{L} = I, \quad \tilde{U} = \begin{pmatrix}
\varepsilon & 1 \\
0 & u_{22}
\end{pmatrix}, \quad \tilde{y} = y,
\]

\[
\tilde{x} = \begin{pmatrix}
\varepsilon^{-1}(1 + \varepsilon u_{22})^{-1} \\
-(\varepsilon u_{22})^{-1}
\end{pmatrix} = \begin{pmatrix}
(1 - \varepsilon)^{-1}((1 + \eta)^{-1}(1 + \eta - \varepsilon^{-1})) \\
-(1 - \varepsilon)^{-1}(1 + \eta)^{-1}
\end{pmatrix}.
\]

Let us first consider the error in $\tilde{x}$:

\[
\tilde{x} - x = \begin{pmatrix}
(1 - \varepsilon)^{-1}((1 + \eta)^{-1}(1 + \eta - \varepsilon^{-1}) - 1) \\
-(1 - \varepsilon)^{-1}(1 + \eta)^{-1} - 1
\end{pmatrix} = (1 - \varepsilon)^{-1} \begin{pmatrix}
\frac{\eta}{1 + \eta} \\
\frac{\eta}{1 + \eta}
\end{pmatrix}.
\]

42
Hence, since \(|c| << 1|\),
\[
\|\delta x - x\|_2 = \frac{1}{|c| - 1(1 + \eta)} |c^{-1}| .
\]

So,
\[
(9) \quad \frac{\|\delta x - x\|_2}{\|x\|_2} \leq \frac{|c^{-1}|}{(1 + \eta)} .
\]

It follows from (8) that, if the algorithm were forward stable, it should be possible to give an upper bound for the left-hand side of (9) that is independent of \(c\); however, since \(c\) may be arbitrarily small, this is not so, and we conclude that the algorithm is not forward stable.

Next, let us consider the backward error in \((A, b)\), Let
\[
S := \min \left\{ \frac{\| (A - A') (b - b') \|_2}{\| A (b - b') \|_2} \mid (A', b') \in \mathcal{E}', \quad \tilde{x} = (A')^{-1} b' \right\} .
\]

Since \(\| A \|_2 = \frac{1}{\sqrt{\rho}} \), and \(\| (A - A') (b - b') \|_2 < 3\), we have
\[
(10) \quad S \geq \frac{1}{3\sqrt{\rho}} \min \{ \| (A - A') (b - b') \|_2 \mid (A', b') \in \mathcal{E}', \quad \tilde{x} = (A')^{-1} b' \} .
\]

Let \((A', b') \in \mathcal{E}'\) minimize the right-hand side of (10).

Since \(\begin{bmatrix} x \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix}^\top\) is a projector, we have
\[
\| (A - A') (b - b') \|_2 \geq \| (A - A') \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2 \| \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2
\]
and, therefore,
\[
S \geq \frac{1}{3\sqrt{\rho}} \left\{ \| (A - A') \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2 \right\} \| \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2 = \frac{1}{3\sqrt{\rho}} |c^{-1}| \frac{n}{1 + \eta} \| \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2
\]

Using that
\[
\| \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2 \geq \sqrt{3} \| \begin{bmatrix} x \\ -1 \\ 0 \end{bmatrix} \|_2 = \sqrt{3} |c^{-1}| \| 1 - (1 + \eta) |c^{-1}| \|
\]

we obtain
\[
(11) \quad S \geq \frac{1}{3\sqrt{\rho}} \frac{|c^{-1}|}{(1 + \eta) \sqrt{3n}} .
\]
From (7) it follows that, if the algorithm were backward stable, it should be possible to give an upper bound for $S$ that is independent of $n$ and $\epsilon$. However, from (11) it follows that this is not even possible for a lower bound of $S$ and we conclude that the algorithm is not backward stable.

Summarizing, without partial pivoting, the algorithm is numerically unstable on $D'$.

IV. We consider an overdetermined system of linear equations $Ax = b$, with $A$ a $(k \times n)$ matrix with $r(A) = k$, and we suppose that $b$ belongs to the column space of $A$.

So,

$$
D = \{ (A, b) \mid (A, b) \in H^k_{\mathbb{R}}(\mathbb{R}) \times \mathbb{R}^n, (A; b) \in H^k_{\mathbb{R}, \mathbb{R}^n} \},
$$

$$
E = \mathbb{R}^k,
$$

is defined by

$$
\xi = \langle A, b \rangle_{D} \to A^*b.
$$

For the metric in $D \cup \{ 0 \}$ we take

$$
\| (A_1 - A_2; b_1 - b_2) \|_E \quad \text{where} \quad (A_1, b_1) \in D \cup \{ 0 \} (i = 1, 2).
$$

For the metric in $E \cup \{ 0 \}$ we take

$$
\| \xi_1 - \xi_2 \|_E \quad \text{where} \quad \xi_1 \in E \cup \{ 0 \} (i = 1, 2).
$$

We consider the algorithm:


(ii) Make a Cholesky decomposition of $A^*A$:

$$
A^*A = R^T R.
$$

(iii) Solve the triangular systems $R^Ty = c$ and $Rx = y$.

Let $(A, b) \in D$. We shall investigate the effect of artificial round-off errors in the computation of $A^*A$:

$$
\overline{A^*A} = A^*A + \xi(A^*A), \quad \| \xi(A^*A) \|_E = \eta\|A^*A\|_E, \quad \eta > 0.
$$
We assume that $\gamma$ is so small that the algorithm can be applied to $(A, b)$ on a machine with relative precision $\gamma$, without a breakdown in the computational process; and, moreover, that in the analysis below, we may account for higher order terms in $\gamma$ by multiplying first order terms by a factor $1.1$ in case we consider upper bounds, and by a factor $0.9$ in case we consider lower bounds.

We have

$$
(A^T A) + \Delta(A^T A) = R^T R, \quad R\delta(A^T A)\|_m = \gamma \| A^T A \|_m,
$$

$$y = (\delta)^{-1}c,
$$

$$k = (\delta)^{-1}y.$$

$x$ denotes the exact solution $(x = A^T b = (A^T A)^{-1}A^T b)$.

Let us first consider the error in $x$. We have

$$
(x - x) = \Delta(x) = 0.
$$

It follows that

$$
\frac{\|x - x\|_m}{\|x\|_m} = \frac{\|\Delta(x)\|_m}{\|x\|_m} \leq 1 + \|A^T A\|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m^{-1} \leq 0.9 \|A^T A\|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m^{-1}.
$$

Since $\Delta(A^T A)$ has no other relation to $A^T A$ than $\|\Delta(A^T A)\|_m = \gamma \| A^T A \|_m$, we should consider

$$
S := \sup \{ 0.9 \|A^T A\|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m^{-1} : \|A^T A\|_m = \gamma \| A^T A \|_m \}
$$

as a realistic lower bound for $\frac{\|x - x\|_m}{\|x\|_m}$ (compare Van der Sluis [3]).

Let $\Delta(A^T A)$ be so chosen that

$$
\|\Delta(A^T A)\|_m \leq 0.9 \|A^T A\|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m =
$$

$$= \|\Delta(A^T A)\|_m \|A^T A\|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m = \gamma \| A^T A \|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m.
$$

For this $\Delta(A^T A)$ it applies that

$$
\frac{\|x - x\|_m}{\|x\|_m} \geq 0.9 \| A^T A \|_m^{-1} \|\Delta(A^T A)\|_m^{-1} \|x\|_m.
$$

(13)
It follows from (6), that if the algorithm were forward stable it should be possible to give an upper bound for \( \frac{\|X - X_0\|_m}{\|I\|_m} \) that is independent of \((A,b)\). From (13) we see that this is not even possible for a lower bound of \( \frac{\|b - X\|_m}{\|I\|_m} \) and we conclude that the algorithm is not forward stable.

Next, let us investigate whether the algorithm is backward stable. We shall do this by giving a sensitivity analysis of the problem and by comparing the result with (13).

We assume that \(A\) is replaced by \((A + \Delta A)\) so that \(I\Delta A_{\mathcal{L}} < \|A^*\|_2^{-1}\) and that \(b\) is replaced by \((b + \Delta b)\).

\[
\Delta x := \nabla^* (b + \Delta b) - x = (A + \Delta A)^* (b + \Delta b - (A + \Delta A)x) = (x - A^* b) - \nabla^* (\Delta b - \Delta x),
\]

Hence,

\[
\|\Delta x\|_m \leq \|\Delta A\|_m \|\Delta (b;\Delta b)\|_m \|\nabla\|_{\mathcal{L}}^{-1}.
\]

Since for \((k \times k)\) matrices, \(\frac{1}{\|\cdot\|_2} \leq \|\cdot\|_m \leq \|\cdot\|_2 \leq \|\cdot\|_k\) and, furthermore, if \(\|\Delta A\|_2 < \|A^*\|_2^{-1}\),

\[
b \|(A + \Delta A)^*\|_2 \leq \frac{\|A^*\|_2}{1 - \|\Delta A\|_2 \|A^*\|_2}
\]

it applies that

\[
\frac{\|\Delta x\|_m}{\|\Delta (b;\Delta b)\|_m} \leq \frac{\|\Delta A\|_m}{\|\Delta (b;\Delta b)\|_m} \frac{\psi(A)}{\|A\|_m}
\]

provided that the denominator in (14) is positive.

If the algorithm were backward stable, then we should have that \(\bar{x}\) is the exact solution of a numerical neighbour \((A + \Delta A; b + \Delta b)\) of \((A,b)\), where
\[ \| (A; b) \|_\infty \leq C_1 \eta \]
with \( C_1 \) independent of \( \eta \) and independent of \( (A, b) \); hence - with (14) - we should have that
\[ \frac{\| x \|_\infty}{\| A \|_\infty} \leq 1.1 k C_1 C_1 \eta \frac{\| (A; b) \|_\infty}{\| A \|_\infty} . \]

If \( (A, b) \) was so chosen that \( \frac{\| (A; b) \|_\infty}{\| A \|_\infty} \leq 2 \) and \( \| x \|_\infty = 1 \), then we have the lower bound (13) as well as the upper bound
\[ \| x - x \|_\infty \leq 2.2 k C_1 C_1 \eta \]
with \( C_1 \) independent of \( (A, b) \).
However, since \( C_1 (A^T A) \leq \frac{1}{k} C_1^2 (A) \), this is only possible if \( C_1 \) does depend on \( (A, b) \), which is a contradiction.

Therefore, we conclude that the algorithm is not backward stable.

Summarizing, the algorithm is numerically unstable on \( D \).

We conclude this chapter with a few remarks.

1. If in example IV, \( D \) is restricted to \( D(N) := \{(A, b) \mid (A, b) \in D; C_1 (A) \leq N, N \geq 1\} \), then it can be shown that the algorithm is numerically stable on \( D(N) \) for all \( N > 0 \).

2. The kind of error analysis used in this section is a priori error analysis.
If with a priori analysis an upper bound for some error is derived - as for the backward error \( E \) in example II - then one should realize that the real error is, generally, due to the statistical distribution of round-off errors, much smaller than the upper bound. The important thing is that a uniform upper bound exists (Wilkinson [107]). However, if with an a priori error analysis a lower bound for some error is derived - as for the forward error in \( x \) of example IV - then the real error will, generally, be of the same order of magnitude, although the actual value of the lower bound does not predict the precise value of the real error.
3. In the definition of numerical stability it is possible that \( \gamma_0 \) strongly depends on \( d \in D \). It may be that \( \inf \{ \gamma_0(d) \mid d \in \Omega(d_0, \epsilon) \} = 0 \) for some \( d_0 \in D \) and all \( \epsilon > 0 \) is equal to zero. For the algorithms that we know this is not so, which suggests to include it as a demand in the definition of numerical stability. We have not done this to avoid too great a complexity.
2. PRELIMINARY ON NUMERICAL RANK

If a \( k \times k \) matrix \( A \) with \( k \leq \) \( k \) has \( r(A) < k \), any spherical neighbourhood of \( A \) contains matrices with rank greater than \( r(A) \). Therefore, the rank of a matrix is, numerically, a senseless notion. Yet, it is in many applications of interest to know the rank.

\( r(A) \) has also the property that a neighbourhood of \( A \) exists in which \( r(A) \) is the minimum rank. This suggests that the notion of "smallest rank in a certain neighbourhood" might be an acceptable substitute for \( r(A) \).

This idea will be elaborated and lead to the definition of the "\( \varepsilon \) -stable rank". The \( \varepsilon \) -stable rank depends on an a priori chosen positive \( \varepsilon \) and constitutes a lower bound for \( r(A) \). If \( \| \cdot \|_2 \) is used as matrix norm, the \( \varepsilon \) -stable rank is best understood as follows:

Let the singular values of \( A \) be denoted by \( \sigma_i, 1 \leq i \leq k \), and let

\[
0 < \sigma_1 \leq \ldots \leq \sigma_k \quad \text{if } A \text{ has a cluster of singular values close to zero, i.e.,}
\]

\[
\{ \sigma_k, \ldots, \sigma_p+1 \} \quad \text{such that} \quad \sigma_k \leq \varepsilon \sigma_{k-1} \leq \ldots \leq \sigma_p \leq \varepsilon \sigma_{p+1} \leq \sigma_{p+2} \leq \varepsilon,
\]

\[
\text{but } \sigma_p - \sigma_{p+1} > \varepsilon; \quad \sigma_p > 2\varepsilon,
\]

then, neglecting the cluster, the \( \varepsilon \) -stable rank is taken equal to \( p \).

If \( A \) satisfies a certain linear condition, it can be shown that the \( \varepsilon \) -stable rank is not influenced by perturbations in \( A \) of the order of \( \varepsilon \).

It will be discussed how the \( \varepsilon \) -stable rank can be found if various matrix norms are used. Special attention will be given to the problem: decide whether a matrix has, numerically, full rank.

2.1. Introduction

Throughout, we shall assume that we are dealing with \( k \times k \) (\( k \leq \) \( k \) ) matrices, which will be denoted by capital letters. The spherical neighbourhood \( (A + E \mid \|E\| < \varepsilon) \) will be denoted by \( U(A, \varepsilon) \), its closure by \( \overline{U}(A, \varepsilon) \).

\( \| \cdot \| \) denotes a submultiplicative matrix norm.

**Theorem 1.**

There exists \( \delta > 0 \) such that \( r(B) = \varepsilon \) holds for all \( B \in U(A, \varepsilon) \).

For any \( \delta > 0 \) there exists \( B \in U(A, \varepsilon) \) such that \( r(B) = \varepsilon \).
Proof.
Let $r := r(A)$.
The theorem is trivial if $r = 0$. Let $r > 0$.

(i) $A$ has an $r \times r$ submatrix $A_r$ such that $r(A_r) = r$ and, equivalently, $\det(A_r) \neq 0$. Because $\det(.)$ is a continuous function, there exists $\delta > 0$ such that $\det(B_\delta) > 0$ and, equivalently, $r(B_\delta) = r$ for all $B_\delta \in \mathcal{U}(A,A_r,\delta)$.

Let $B$ be a $k \times k$ matrix and let $B_\delta$ be the submatrix of $B$ that corresponds with $A_r$. Independent of the particular $B$ we have $\|B_\delta - A_r\| \leq \|B - A\|$ and $r(B_\delta) \leq r(B)$.
Hence, there exists $\delta > 0$ such that $r(B) \geq r$ for all $B \in \mathcal{U}(A,A_r,\delta)$.

(ii) The second part of the proof is trivial if $r = k$. Let $r < k$.

Let us suppose that the column space of $A$ is spanned by the set of columns of $A$: \{ $a_1, \ldots, a_k$ \}. We can define a $k \times \ell$ matrix $E = (v_1, \ldots, v_k, 0, \ldots, 0)$ such that $\|E\| < 1$ and \{ $a_1, \ldots, a_k, v_1, \ldots, v_k$ \} spans $\mathbb{R}^k$. For any $\delta > 0$, $r(A + \delta E) = k$ holds, whereas $\|E\| < \delta$.

The first part of the theorem states that $r(A)$ is an upper semi-continuous function of $A$.

Clearly, it has no sense to determine the rank of a matrix by means of a computer unless the matrix has full rank, a number is supplied that is larger than the rank. However, $r(A)$ is the minimum rank in a certain neighbourhood of $A$. This suggests that $\rho(A_\delta)$ as defined by

Definition 2.

\[ \rho(A_\delta) := \min \{ \text{rank}(B) | B \in \mathcal{U}(A,A_\delta) \} \]

might, for some $\delta > 0$, be an acceptable substitute for $r(A)$.

If $\delta$ is small enough, we have $\rho(A_\delta) = r(A)$; henceforth, we shall show that, for almost any $\delta$, $\rho(A_\delta)$ is not influenced by small changes in $A$ or $\delta$.

We shall characterize $\rho(A_\delta)$ with the following quantities.
Definition 3.
\[
\begin{align*}
\gamma_i(A) &:= \sup \{ \delta \mid r(B) < i \text{ for all } B \in U(A, \delta) \} \quad (1 \leq i \leq r(A)) \\
\gamma_0(A) &:= +\infty; \quad \gamma_{r(A)+1}(A) := 0 \quad (r(A) < i \leq r(A)).
\end{align*}
\]

The \( \gamma_i(A) \) will be called the critical values of \( A \).

With \( r := r(A) \) we have
\[\eqref{gamma}\]
\[
0 = \gamma_{r+1}(A) < \ldots < \gamma_r(A) < \ldots < \gamma_1(A) = \|A\|.
\]

From definition 3 it follows that, unlike \( U(A, \gamma_1) \), \( U(A, \gamma_2) \) does contain a matrix \( B \) with \( r(B) < i \) \( (1 \leq i \leq r(A)) \). Therefore, \( U(A, \gamma_2) \) is the smallest spherical neighbourhood of \( A \) that contains a matrix \( B \) with \( r(B) < i \).

**Theorem 4.**

Let \( \delta > 0 \). \( \rho(A, \delta) = p \) if and only if \( \gamma_{p+1}(A) < \delta \leq \gamma_p(A) \).

**Proof.**

Let \( \rho(A, \delta) = p \).

Firstly, it follows that \( U(A, \delta) \) contains no matrices \( B \) with \( r(B) < p \).

Hence, we have
\[
U(A, \delta) \subset U(A, \gamma_p) \quad \text{implying} \quad \delta \leq \gamma_p.
\]

Secondly, it follows that \( U(A, \delta) \) does contain a matrix \( B \) with \( r(B) = p \).

Hence, we have
\[
U(A, \delta) \supset U(A, \gamma_{p+1}) \quad \text{implying} \quad \delta > \gamma_{p+1}.
\]

The other part of the theorem may be readily verified.

\[
\begin{array}{c}
\gamma_1(A) \quad \gamma_2(A) \quad \gamma_3(A) \quad \gamma_4(A) \quad \delta \\
\hline
\rho(A, \delta) \\
\text{r-1} \quad \text{r-2} \quad \text{r-3} \\
\end{array}
\]

\( \rho(A, \delta) \) is continuous as a function of \( \delta \), unless \( \delta \) is equal to a critical value. We see that \( \rho(A, \delta) \), for almost any \( \delta \), is not influenced by small changes in \( \delta \).
Next, we shall investigate the sensitivity of $\varepsilon(A, \delta)$ to changes in $A$. Let $\delta > 0$ and let $\varepsilon > 0$.

If $B \in U(A, \delta)$, then $U(B, \delta) = U(A, \delta + \varepsilon)$. Hence

1. $\varepsilon(B, \delta) \geq \varepsilon(A, \delta + \varepsilon)$ for all $B \in U(A, \varepsilon)$.

Let $\delta > 0$.

If $B \in U(A, \delta)$, then $U(B, \delta) = U(A, \delta + \varepsilon)$. Hence

2. $\varepsilon(B, \delta) \leq \varepsilon(A, \delta + \varepsilon)$ for all $B \in U(A, \varepsilon)$.

**Theorem 6.**

Let $\delta > 0$ and let $\varepsilon(A, \delta) = p$. Then we have

1. $\varepsilon(B, \delta) = \varepsilon(A, \delta)$ for all $B \in U(A, \delta - \varepsilon)$.

2. $\varepsilon(B, \delta) \leq \varepsilon(A, \delta)$ for all $B \in U(A, \delta - \gamma_p - \varepsilon)$.

3. $\varepsilon(B, \delta) \leq x(A)$ for all $B \in U(A, \delta)$.

**Proof.**

1. Let $\varepsilon = \gamma_p - \delta$. As a consequence of theorem 4 we have $\varepsilon > 0$.

   Applying (2), it follows that $\varepsilon(B, \delta) \geq \varepsilon(A, \delta + \gamma_p - \varepsilon) = \varepsilon(A, \gamma_p - \varepsilon) = \varepsilon(B, \delta)$ for all $B \in U(A, \gamma_p - \delta)$.

2. As a consequence of theorem 4 we have $\delta - \gamma_p - \varepsilon > 0$.

   Let $\varepsilon = \varepsilon - \gamma_p - \varepsilon$, where $0 < \varepsilon < \delta - \gamma_p - \varepsilon$. So, $\varepsilon > 0$.

   Applying (3), it shows that $\varepsilon(B, \delta) \leq \varepsilon(A, \delta - \gamma_p - \varepsilon) = \varepsilon(A, \delta - \varepsilon)$ for all $B \in U(A, \delta - \gamma_p - \varepsilon)$.

   Hence, if we let $\varepsilon > 0$,

   $\varepsilon(B, \delta) \leq \varepsilon(A, \delta)$ for all $B \in U(A, \delta - \gamma_p - \varepsilon)$.

3. This is trivial, since, if $B \in U(A, \delta)$, then $A \in U(B, \delta)$ and thus $\varepsilon(B, \delta) \leq x(A)$.

**Remark.**

If $\varepsilon > 0$, then there exists $B \in U(A, \delta - \varepsilon + \varepsilon)$ such that $\varepsilon(B, \delta) < \varepsilon(A, \delta)$.

It depends on the particular $A$ whether there exists $B \in U(A, \delta - \gamma_p - \varepsilon)$, such that $\varepsilon(B, \delta) > \varepsilon(A, \delta)$.
From parts (i) and (ii) of theorem 5 it follows that, if $\rho$ is not equal to a critical value of $\delta$, $\rho(\delta)$ is continuous as a function of $\delta$.

Combining parts (i) and (ii) of theorem 5, it shows that $\rho(\delta)$, for almost any $\delta$, is not influenced by small changes in $\delta$.

Having investigated the sensitivity of $\rho(\delta)$ to changes in $\delta$ respectively, let us introduce a new notion.

**Definition 6.**

Let $\varepsilon > 0$.

$\rho(\delta)$ is called $\varepsilon$-stable in $\delta$, if

\[ \delta \leq \varepsilon \text{ and } \forall \delta' [0 < \delta' \leq \delta + \varepsilon \Rightarrow \rho(\delta' + \varepsilon) = \rho(\delta)] . \]

**Theorem 7.**

Let $\delta \geq \varepsilon > 0$ and suppose $\rho(\delta) = \rho$.

The following statements are equivalent:

(i) $\rho(\delta)$ is $\varepsilon$-stable in $\delta$.

(ii) $\rho(\delta) = \rho(\delta + \varepsilon)$ for all $\delta \leq U(A, \varepsilon)$.

(iii) $\rho(\delta) = \rho(\delta + \varepsilon) = \rho(\delta)$.

(iv) $\delta \leq \gamma_p(\delta) = \varepsilon$.

**Proof.**

We shall show that (i) $\Rightarrow$ (iii) $\Rightarrow$ (iv) $\Rightarrow$ (ii) $\Rightarrow$ (i).

(i) $\Rightarrow$ (iii). If in definition 6 we take $\delta' = \delta + \varepsilon$, we obtain

$\rho(\delta + \varepsilon) = \rho(\delta)$. Since $\rho(\delta)$ is not increasing as a function of $\delta$, we obtain $\rho(\delta + \varepsilon) = \rho(\delta)$.

(iii) $\Rightarrow$ (iv). Applying theorem 4, the result is easily obtained.

(iv) $\Rightarrow$ (iii). As $\varepsilon < \delta$, we have $\varepsilon < \delta < \gamma_p = \epsilon$. Application of parts (i) and (iii) of theorem 5 gives (ii).

(ii) $\Rightarrow$ (i). It is obvious that $\delta' = \min(\rho(\delta), \delta + \varepsilon) = \rho(\delta + \varepsilon)$.

**Theorem 8.**

Let $\varepsilon > 0$. There is a $\delta \geq \varepsilon$ such that $\rho(\delta) = \varepsilon$-stable.
Summarizing, if \( \rho(A, \delta) \) is \( \varepsilon \)-stable, then
\[
\tau(A) \geq \rho(B, \delta) \geq \rho(A, \delta) \quad \text{for all } B \in U(A, \varepsilon)
\]
and
\[
\tau(A) \geq \rho(A, \delta') \geq \rho(A, \delta) \quad \text{for all } \delta' \text{ with } 0 < \delta' \leq \delta + \varepsilon,
\]
that is, \( \rho(A, \delta) \) is non-decreasing for perturbations of the order of \( \varepsilon \) in \( A \) and \( \delta \).

Let us consider the practical use of \( \rho(A, \delta) \).

Suppose we have a numerical method available for computing the critical values of \( A \). Then a computer supplies with that method the exact critical values of a neighboring matrix of \( A \) (\( \tilde{A} \) say).

Let us suppose that a backward error analysis shows that \( \| A - \tilde{A} \| < \varepsilon \).

The computer can supply \( \rho(\tilde{A}, \delta) \) for any \( \delta > 0 \). If \( \delta \geq \varepsilon \), then \( \rho(\tilde{A}, \delta) \) is a lower bound for \( \tau(A) \) and, if \( \varepsilon \) and \( \delta \) are small enough, \( \rho(\tilde{A}, \delta) \approx \tau(A) \). One might be inclined to take \( \delta = \varepsilon \), because, if \( \varepsilon \) is small enough, we shall have \( \rho(\tilde{A}, \varepsilon) = \tau(A) \) and because a greater value of \( \delta \) may result in a poorer lower bound for \( \tau(A) \). However, taking \( \delta = \varepsilon \) as a general strategy has its drawbacks.

Firstly, when \( \varepsilon \) is a sharp estimate for \( \| A - \tilde{A} \| \), the distance of \( A \) to a matrix \( B \) with \( \tau(B) < \rho(\tilde{A}, \varepsilon) \) may be small compared to \( \varepsilon \). This is illustrated by the figure.

![Diagram](image)

Secondly, when \( \varepsilon \) is a rude estimate for \( \| A - \tilde{A} \| \) (a situation that should be avoided), it is unsatisfactory that a small increment of \( \varepsilon \) might result in a lower value of \( \rho \).

54
From the foregoing it is clear that \( \rho(\overline{A}, \delta) \) with \( \delta \geq \varepsilon \) does not have these disadvantages, if \( \rho(\overline{A}, \delta) \) is \( \varepsilon \)-stable. If \( \rho(\overline{A}, \delta) \) is \( \varepsilon \)-stable, we have

(i) \( U(\overline{A}, \varepsilon) \) contains no matrix \( \delta \) with \( \gamma(\delta) < \rho(\overline{A}, \varepsilon) \),

(since \( U(\overline{A}, \varepsilon) = U(\overline{A}, \delta + \varepsilon) \) and \( \rho(\overline{A}, \delta + \varepsilon) = \rho(\overline{A}, \delta) \)).

(ii) Small changes in \( \delta \) do not result in a lower value of \( \rho \).

Summarizing, we conclude:

Should one wish to use \( \rho(\overline{A}, \varepsilon) \), then it is preferable to determine as well whether \( \rho(\overline{A}, \varepsilon) \) is \( \varepsilon \)-stable.

If \( \rho(\overline{A}, \varepsilon) \) is not \( \varepsilon \)-stable, then it is not unrealistic to take a smaller value than \( \rho(\overline{A}, \varepsilon) \) as approximation for \( \gamma(A) \). This may be achieved by taking \( \rho(\overline{A}, \delta) \) with \( \delta > \varepsilon \) as an approximation for \( \gamma(A) \) where \( \delta \) is chosen so that \( \rho(\overline{A}, \delta) \) is \( \varepsilon \)-stable.

2.2. The \( \varepsilon \)-stable rank

Inspired by the last remark of 2.1 and the fact that, given \( \varepsilon > 0 \), we can always find a \( \delta > 0 \) such that \( \rho(\overline{A}, \delta) \) is \( \varepsilon \)-stable, we propose the following substitute for \( \gamma(A) \):

**Definition 9.**

\[ r_\varepsilon(A) := \max \{ \rho(\overline{A}, \delta) \mid \rho(\overline{A}, \delta) \text{ is } \varepsilon \text{-stable} \} \]

We shall refer to \( r_\varepsilon(A) \) as the \( \varepsilon \)-stable rank.

The reason to define \( r_\varepsilon(A) \) as a maximum is that we want \( r_\varepsilon(A) \) to be a best possible lower bound for \( \gamma(A) \).

From the definition of \( \varepsilon \)-stability and theorem 7, we obtain:

**Theorem 10.**

Let \( \varepsilon > 0 \).

\[ r_\varepsilon(A) = \max \{ t \mid \gamma_t - \gamma_{t+1} > \varepsilon \text{ and } \gamma_t \geq 2\varepsilon \} = \max_{\delta \geq \varepsilon} \{ \rho(\overline{A}, \delta) \mid \rho(\overline{A}, \delta + \varepsilon) = \rho(\overline{A}, \delta) \} . \]
The first line of theorem 10 may be understood as:
If \( A \) has a cluster of critical values close to zero, i.e.
\[
\{ \gamma_k, \ldots, \gamma_{k+1} \mid \gamma_k - \gamma_{k+1} \leq \varepsilon, 2 = k, \ldots, p^* \};\ 
\gamma_{p+1} > \varepsilon \text{ and } \gamma_p < 2\varepsilon \},
\]
then, neglecting the cluster, \( r_c(A) \) is taken equal to \( p \).
In order to attain some insight in the behaviour of \( r_c(A) \) as a function of \( \varepsilon \), we give two figures.
The first figure shows in which regions of the \( \delta - \varepsilon \) plane, \( r(A, \delta) = r(A, \delta + \varepsilon) \) and \( \delta > \varepsilon \) (the shaded areas).
The second figure is constructed by finding for each \( c \) the first region, where \( r(A, \delta) = r(A, \delta + \varepsilon) \) and \( \delta > \varepsilon \).
We see that \( r_c(A) \) may decay more than 1 and that \( r_c(A) = r(A) \) if \( \varepsilon < \frac{1}{2} \gamma_c \).
Furthermore, it is clear that \( r_c(A) \) is continuous as function of \( \varepsilon \) for almost any value of \( \varepsilon \).
Next, let us consider $r_c(A)$ as a function of $A$.

Let $\varepsilon > 0$ and let $r_c(A) = p$.

From the definition of $c$-stability and $r_c(A)$ we obtain that for any $\delta$ satisfying

\[ \delta \geq \varepsilon \text{ and } \gamma_{p+1} - \delta \leq \gamma_p - \varepsilon \]

it holds that

\[ r_c(A) = \sigma(A, \delta + \varepsilon) = \sigma(A, \delta) = p. \]

Let $\delta$ be accordingly chosen. If we apply theorem 5, it shows that

\[ \sigma(B, \delta + \varepsilon) = \sigma(B, \delta) = \sigma(A, \delta) \text{ for all } B \not\in \mathcal{U}(A, \delta) \]

where $\delta = \min(\gamma_p - (\delta + \varepsilon), \gamma_p - \gamma_{p+1})$.

Consequently,

\[ r_c(B) \geq r_c(A) \quad \text{for all } B \not\in \mathcal{U}(A, \delta). \]

Theorem 11.

Let $\varepsilon > 0$ and let $r_c(A) = p$. Let $\delta := \gamma_{p+1} + \frac{1}{2}(\gamma_p - \gamma_{p+1})$.

If $\delta \geq \varepsilon$, we define $\delta := \frac{1}{2}(\gamma_p - \gamma_{p+1})$, otherwise $\delta := \gamma_p - 2\varepsilon$.

Then

\[ r_c(B) \geq r_c(A) \quad \text{for all } B \not\in \mathcal{U}(A, \delta). \]

If $B \not\in \mathcal{U}(A, \delta)$, then $r_c(B)$ might be less than $r_c(A)$, dependent on the particular $A$.

Thus, $r_c(A)$ is upper semi-continuous as function of $A$ for almost any $\varepsilon$.

Let us give an estimate for $\delta := \inf \{ \delta \mid \sigma(A, \delta) \text{ is } c\text{-stable} \}$. As

\[ \gamma_{p+1} - \delta \geq \gamma_p - \varepsilon \text{ and } \delta \geq \varepsilon \Rightarrow r_c(A) = \sigma(A, \delta) = \sigma(A, \delta + \varepsilon) = p, \]

we have $\delta \leq \max(\varepsilon, \gamma_{p+1})$.

If $\gamma_{p+1} > \varepsilon$, then

\[ \delta = \gamma_{p+1} = (\gamma_{p+1} - \gamma_p - \varepsilon_2) + \cdots + (\gamma_{p+1} - \gamma_p - \varepsilon_t) + \gamma_{p+1}, \quad t \leq \tau - p. \]
Let $\xi$ be so chosen that $\gamma \neq \xi > 2 \epsilon$, but $\gamma_{\neq \xi < 2 \epsilon}$. Then, with theorem 10, it follows that

$$\delta_0 < \xi + 2 \epsilon \leq (\tau(A) - \tau_{\xi}(A))\epsilon + 2 \epsilon.$$

**Theorem 12.**

Let $\epsilon > 0$ and let $r_{\xi}(A) = \rho$. Let $\delta_0 := \inf \{ \delta : \rho(A, \delta) \text{ is } \epsilon \text{-stable} \}$. Then either $\delta_0 = \epsilon$, or $\delta_0 = \gamma_{\neq \xi < \epsilon}$.

Let us consider the practical use of $r_{\xi}(A)$.

As before we assume that a numerical method is available for computing the critical values of $A$. A computer supplies the exact critical values for a neighbouring matrix of $A$ ($\tilde{A}$, say). Again we assume that backward error analysis has shown that $\| A - \tilde{A} \| < \epsilon$. The computer then can supply $r_{\epsilon}(\tilde{A})$, which is a lower bound for $r(A)$.

Let $\delta$ be so chosen that $r_{\epsilon}(\tilde{A}) = r(A, \delta)$ and $\rho(\tilde{A}, \delta)$ is $\epsilon$-stable. We then have (theorems 7) that $\rho(\tilde{A}, \delta) > r_{\epsilon}(\tilde{A})$, or, in a $\delta$-neighbourhood of $A$ no matrix with lower rank than $r_{\epsilon}(\tilde{A})$ occurs.

Secondly, $\rho(\tilde{A}, \delta)$ is not influenced in a bad sense by small changes in $\delta$.

Let us compare $r_{\epsilon}(\tilde{A})$ and $\rho(\tilde{A}, \epsilon)$.

If $\rho(\tilde{A}, \epsilon)$ is $\epsilon$-stable, $r_{\epsilon}(\tilde{A}) = \rho(\tilde{A}, \epsilon)$. If $\rho(\tilde{A}, \epsilon)$ is not $\epsilon$-stable, then — as we saw in 2.1 — a matrix $\tilde{B}$ with $r(\tilde{B}) < \rho(\tilde{A}, \epsilon)$ might be close to $A$ and it is then reasonable to take a lower value as approximation for $r(A)$; indeed, in this situation we have $r_{\epsilon}(\tilde{A}) < \rho(\tilde{A}, \epsilon)$.

We shall say that the computation of $r_{\epsilon}(A)$ is a well-conditioned problem, if $r(A) > r_{\epsilon}(B) > r_{\epsilon}(A)$ for all $B \in B(\tilde{A}, \epsilon)$.

In that case, different computers but with the same relative machine precision will supply approximations $r_{\epsilon}(\tilde{A})$ satisfying $r(A) > r_{\epsilon}(\tilde{A}) > r_{\epsilon}(A)$.

In order that the computation of $r_{\epsilon}(A) = \rho$ is well-conditioned, it is sufficient that (using theorem 11):
\[ \gamma_p(A) - \gamma_{p+1}(A) \geq 3\varepsilon \]

and

\[ \gamma_{\varepsilon}(A) - \gamma_{\varepsilon+1}(A) \leq \varepsilon \text{ or } \gamma_{\varepsilon}(A) < 2\varepsilon, \text{ for } \varepsilon > p. \]

However, it is not that important that the computation of \( \gamma\varepsilon(A) \) is well-conditioned; it is always possible to make a statement concerning a certain neighbourhood of \( A \).

**Remark.**

If we are interested in the question whether \( A \) has full rank, our numerical method only needs to compute \( \gamma_k(A) \), since

\[ \gamma_k(A) = k \Leftrightarrow \gamma_k(A) \geq 2\varepsilon \quad \text{(Theorem 10)}. \]

If \( \gamma_k(A) = k \), then \( \rho(A, 3\varepsilon) = k \); if \( \gamma_k(A) < k \), then \( \rho(A, 3\varepsilon) < k \).

2.3. The Critical Values

If the critical values are known, the \( \varepsilon \)-stable rank can be computed (Theorem 10). These values depend on the matrix norm. With \( \| \cdot \|_2 \) or \( \| \cdot \|_\infty \) as matrix norm, the critical values can be expressed in terms of the singular values.

Let \( A \) have singular values \( 0 \leq \sigma_k(A) \leq \sigma_{k-1}(A) \leq \ldots \leq \sigma_1(A) \).

In the 2-norm it holds that

\[ \gamma_k^2(A) = \sigma_k^2(A) \quad (1 \leq i \leq k). \]

In the \( \infty \)-norm it holds that

\[ \gamma_k^\infty(A) = \left( \sum_{i=1}^{\infty} \sigma_i^\infty(A) \right)^{\frac{1}{k}} \quad (1 \leq i \leq k). \]

These results are a consequence of the theorem of Eckart and Young that the best approximation of \( A \) having rank \( p \) — in the 2-norm as well as in the \( \infty \)-norm — is obtained by taking the singular value decomposition of \( A \) and replacing \( \sigma_{p+1}, \ldots, \sigma_k \) by zero [2].

However, for practical applications, the computation of the singular values might be too laborious for the goal one has in mind. One has then to content oneself with lower and upper bounds for the critical values that are computed using an already available decomposition of \( A \). It should be emphasized that then \( \gamma_k(A) \) cannot always be determined. If for \( \gamma_k(A) \) a lower bound \( \gamma_k(\lambda) \) and an upper bound \( \gamma_k(\lambda) \) is known and \( \gamma_k(A) < 2\varepsilon \leq \gamma_k(\lambda) \), it cannot be decided whether \( \gamma_k(A) = k \) or \( \gamma_k(A) < k \).
In this thesis the c-stable rank will be used only to investigate whether a matrix, taking into account of possible round-off errors or noise, has full rank or not. Therefore, we restrict ourselves to lower and upper bounds for \( \gamma_k(A) \), assuming that \( A \) has full rank.

2.3.1.

**Theorem 13.**

Let \( A \) be a \( k \times l \) matrix (\( k \leq l \)) of full row rank. Let \( DA \) be a \( k \times l \) matrix. Then,

\[
\|DA\| < \max \{\|A^{(r)}\|^{-1} | A^{(r)} = I \} \implies r(A + DA) = k.
\]

**Proof.**

Let \( x^T(A + DA) = 0 \), for some vector \( x \). Since \( A \) has full row rank, \( A \) has a right-inverse \( A^{(r)} \). Now,

\[
x^T = x^TDA A^{(r)} \quad \text{and} \quad \|x\| \leq \|x^TDA\| \|DA A^{(r)}\|.
\]

If \( \|DA\| < \|A^{(r)}\|^{-1} \), we must have \( \|x\| = 0 \) and, consequently,

\[
r(A + DA) = k. \quad \square
\]

**Corollary.**

\[
\max \{\|A^{(r)}\|^{-1} | A^{(r)} = I \} \leq \gamma_k(A).
\]

**Remark.**

In the 2-norm or E-norm, we have \( \max \{\|A^{(r)}\|^{-1} \} = \|A^+\|^{-1} \), where \( A^+ \) is the pseudo inverse of \( A \) (Penrose [7] and Golub [5]). Moreover, the lower bound for \( \gamma_k(A) \) of \( \|A^+\|^{-1} \) is then sharp, since \( \gamma_k(A) = c_k(A) = \|A^+\|^{-1} \).

**Theorem 14.**

Let \( A \) be a \( k \times l \) matrix (\( k \leq l \)) and let \( A = LR \) with \( L \) a regular \( k \times k \) matrix and \( R \) a \( k \times l \) matrix. Then,

\[
\gamma_k(R) \|L^{-1}\|^{-1} \leq \gamma_k(A) \leq \gamma_k(R) \|LL^T\|.
\]

60
Proof.
Let $\Delta A$ be a $k \times k$ matrix and let $\Delta R := L^{-1} \Delta A$. Then $A + \Delta A = L(R + \Delta R)$.
Since $L$ is regular, $\tau(A + \Delta A) = \tau(R + \Delta R)$. We have

$$
\| L^{-1} \| \| \Delta A \| < \gamma_k(R) \Rightarrow \| \Delta R \| < \gamma_k(R) \Rightarrow \tau(R + \Delta R) = k.
$$

So

$$
\| \Delta A \| < \gamma_k(R) \Rightarrow \| \Delta R \| < \gamma_k(R) \Rightarrow \tau(R + \Delta R) = k.
$$

Consequently,

$$
\gamma_k(R) \leq \gamma_k(A).
$$

Similarly, employing $R = L^{-1}A$, one obtains $\gamma_k(A) \leq \gamma_k(R)$. So

$$
\gamma_k(R) = \gamma_k(A).
$$

If $A$ is of upper trapezoidal form, it might be easier to determine lower or upper bounds for $\gamma_k(A)$. It is always possible to reduce $A$ to a matrix $R$ of upper trapezoidal form, for instance by a Gaussian elimination process or by an orthogonal decomposition. Then theorem 14 gives the relation between $\gamma_k(A)$ and $\gamma_k(R)$.

$\gamma_k(A)$ might be situated everywhere in the interval given by theorem 14. It is therefore important that the size of the interval is relatively small, or, equivalently, a decomposition method is chosen so that $\| L^{-1} \| L$, the condition number of $L$, is not much larger than 1. From that point of view, an orthogonal decomposition ($L$ is orthogonal) is superior to a decomposition obtained by a Gaussian elimination process.

2.3.2.

Next, let $A$ be partitioned as $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$. We shall derive an expression for $A^*$ in terms of $A_1$ and $A_2$. Again, it is assumed that $A$ (and hence $A_1$ and $A_2$) have full row rank. The expression obtained for $A^*$ can also be found by applying a formula from Albert [1].

Let $b$ be an arbitrary vector in $\mathbb{R}^k$ and let $M(x_1, x_2) := \| A_1 x_1 + A_2 x_2 - b \|_2^2$. With $x_1$ fixed, $M(x_1, x_2)$ is minimized as function of $x_2$ by
\[ x^n_2 = (A_2^T)^+(b - A_1^T x^n_1) \quad \text{\(A_2^T\) has full column rank}, \]

\[ N(x_1, x^n_2) = \| (A_2^T A_2 - I) b - (A_2^T A_2 - I) A_1^T x_1 \|_2. \]

Before minimizing \( M(x_1, x^n_2) \) as function of \( x_1 \), we shall show that \((A_2^T A_2 - I) A_1^T\) has full column rank.

Let \( u \) be a vector satisfying \( (A_2^T A_2 - I) A_1^T u = 0 \). Since \((A_2^T A_2)^T = -(A_2^T A_2)^T\), it follows that

\[ (A_2^T A_2)^T \begin{bmatrix} -u \\ (A_2^T A_1 u) \end{bmatrix} = 0. \]

Consequently, \( u = 0 \) \((A^T\) has full column rank), implying that \((A_2^T A_2 - I) A_1^T\) has full column rank. Hence, \( M(x_1, x^n_2) \) is minimized as function of \( x_1 \) by

\[ x^n_1 = ((A_2^T A_2 - I) A_1^T)^+(A_2^T A_2 - I) b = ((A_2^T A_2 - I) A_1^T)^+ b. \]

Summarizing, \( M \) is minimized as function of \( x_1 \) and \( x_2 \) by

\[ \begin{bmatrix} x^n_1 \\ x^n_2 \end{bmatrix} = \begin{bmatrix} ((A_2^T A_2 - I) A_1^T)^+ \\ (A_2^T)^+(I - A_1^T((A_2^T A_2 - I) A_1^T)^+) \end{bmatrix} b = (A^T)^+ b. \]

\( b \) was arbitrary, so

\[ (4) \quad A^T = \begin{bmatrix} A_1^* \\ (I - A_1 A_2 A_2^*) \end{bmatrix} \quad \text{with} \quad A_1 = A_1 (A_2^T A_2 - I). \]

Similarly, it can be shown that

\[ (5) \quad A^T = \begin{bmatrix} (I - A_2 A_2^*) A_1^* \\ A_2^* \end{bmatrix} \quad \text{with} \quad A_2 = A_2 (A_1^T A_1 - I). \]

Theorem 15.
Let \( A \) be a \( k \times k \) matrix \((k \leq \ell)\) of full row rank. Let \( A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \).
If \( \tilde{A}_1 = A_1 (A_2^T A_2 - I) \) and \( \tilde{A}_2 = A_2 (A_1^T A_1 - I) \), then we have two lower bounds:

\[ \| A_1^* \| + \| A_2^* \| + \| A_1^* \tilde{A}_1 A_2^* \|^{-1} \leq \gamma_k(A), \]

\[ \| A_1^* \| + \| A_2^* \| + \| A_2^* \tilde{A}_2 A_1^* \|^{-1} \leq \gamma_k(A). \]

\[ \square \]

62
If, henceforth, we are interested in $r_{k}(A)$, it concerns matrices of which it is known that the $r$-stable rank of the matrix consisting of the first $(k-1)$ rows of $A$ is equal to $(k-1)$.

Therefore, let $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ be a row vector. Employing that then

\[ A_2^* = A_2^T \|A_2\|^2 \|A_1\|^2 \] and

\[ A_2^* = A_2^T \|A_2\|^2 \|A_1\|^2 \] we obtain from theorem 13 the lower bounds

\[ \frac{\|A_2^*\|^2}{\|A_2^*\|^2 + \|A_2\|^2 + \|A_1A_1^*\|^2} \leq \gamma_k(A) \]

and

\[ \frac{\|A_2^*\|^2}{\|A_2^*\|^2 + \|A_2\|^2 + \|A_2^*A_1^*\|^2} \leq \gamma_k(A) \].

Employing the 2-norm, we obtain from (4)

\[ \|A_2^*\|^2 = \|A_1A_2^T\|^2 + \|A_2^T(1 - A_1A_1^*)\|^2 \leq \|A_2^T\|^2 + \|A_2^T(1 - A_1A_1^*)\|^2 \leq \|A_2^T\|^2 + \|A_2^T\|^2(1 + \|A_1A_2^T\|^2) \]

and it follows that

\[ \frac{\|A_2^*\|^2}{(1 + \|A_2^T\|^2(\|A_1A_2^T\|^2 + \|A_2^T\|^2))^{1/2}} \leq \gamma_k(A) \].

Similarly, we obtain from (5)

\[ \|A_2^*\|^2 \leq \|A_1A_2^T\|^2 + \|A_2^T\|^2(1 + \|A_2^T\|^2) \].

Using that $\|A_2^*\|^2 = \|A_2^T\|^2 + \|A_2^T\|^2$, it readily follows that

\[ \frac{\|A_2^*\|^2}{(1 + \|A_1A_2^T\|^2(\|A_2^T\|^2 + \|A_2^T\|^2))^{1/2}} \leq \gamma_k(A) \]

\[ \frac{\|A_2^*\|^2}{(1 + \|A_2^T\|^2(\|A_2^T\|^2 + \|A_2^T\|^2))^{1/2}} \leq \gamma_k(A) \].
Although lower and upper bounds for the singular values are known in the literature (Ko Tan [1]), we believe that no lower bound for \( \gamma_k(A) \) has been given before that employs a partitioning of \( A \).

Next, let us derive an upper bound for \( \gamma_k(A) \).

Let \( A^{(1)} \) denote the matrix consisting of all rows of \( A \) except the \( i \)-th row which we denote by \( A_i^T \). Let \( P^{(1)} := A^{(1)} (A^{(1)})^T \) be the orthogonal projector on the row space of \( A^{(1)} \).

Obviously, \( r(A^{(1)}) = k-1 \) and \( \|A - AP^{(1)}\| = \|A_i^T (I - P^{(1)})\| \). Consequently, we have

\[
\gamma_k(A) \leq \min_{\text{rank}} \|A_i^T (I - P^{(1)})\|.
\]

**Remark.**

If \( A_2 \) is a row vector, we have in (5): \( \|A_2\| = \|A_i^T (I - P^{(k)})\| \).

2.3.3.

The bounds for \( \gamma_k(A) \) given in 2.3.2 have the disadvantage that the pseudo inverse of a matrix is involved, which reduces their practical value.

Let us assume that \( A \) has a decomposition

\[
AP = LK,
\]

where \( P \) is a permutation matrix, \( L \) is a regular \( k \times k \) matrix and \( K \) is a matrix that can be partitioned as

\[
\begin{pmatrix}
R_{11} & R_{12} \\
0 & R_{22}
\end{pmatrix}
\]

with \( R_{11} \) a regular \( (k-1) \times (k-1) \) matrix.

A right-inverse of \( K \) that is immediately spotted is

\[
K^{-1} = \begin{pmatrix}
R_{11}^{-1} & -R_{11}^{-1} R_{12} R_{22}^{-1} \\
0 & R_{22}^{-1}
\end{pmatrix}.
\]

Hence, a right-inverse of \( A \) is

\[
A^{-1} = P^T K^{-1} L^{-1} = P^T R^{-1} R^{-1} \quad (K = L^{-1}).
\]

We have
\[(13) \quad \|A^T\| = \|R^T\| M_k \leq B \begin{bmatrix} R_{\text{off}} & 0 \\ 0 & 0 \end{bmatrix} M_1 + B \begin{bmatrix} 0 & -R_{\text{off}} \\ 0 & R_{22} \end{bmatrix} M_2 = \|R_{\text{off}}\| (M_{11} \oplus M_{12}) + \|R_{22}\| (M_{21} \oplus M_{22}).\]

**Theorem 16.**

Let \( A \) be a \( k \times k \) matrix (\( k \leq k \)) with a decomposition such as (11). Then

\[
\|R_{22}\|^2 \leq \|R_{\text{off}}\|^2 (M_{11} \oplus M_{12}) + \|R_{22}\|^2 (M_{21} \oplus M_{22}).
\]

**Proof.**

Substitution of \( R_{\text{off}}^* = R_{22}^\top B R_{22}^{-1} \) in (13) and application of theorem 13, gives the desired result.

The lower bound for \( \gamma_k(A) \) given in theorem 16 still involves the evaluation of the inverse of a matrix. However, if \( R_{11} \) is a triangular matrix, this offers no major problems.

An upper bound for \( \gamma_k(A) \) that is easier to compute than the bound (10) is of course

\[
(14) \quad \gamma_k(A) \leq \min_{1 \leq i \leq k} \|a_i\|^2.
\]

Applying (14) and theorem 14 in case for \( A \) a decomposition such as (11) is known, we obtain the upper bound

\[
(15) \quad \gamma_k(A) \leq \|R_{22}\| L_1.
\]
3. NUMERICALLY UNSTABLE, RECURSIVE, MINIMAL REALIZATION ALGORITHMS

In this chapter we shall investigate three recursive algorithms for computing a minimal realization of an impulse response \( \{s_k\}_{k=0}^{\infty} \), namely the algorithms of Massey/Berlekamp, Rissanen, and a new algorithm, which is somewhat faster than the other two.

Under the preconditions that \( \{s_k\}_{k=0}^{\infty} \) admits a minimal realization of finite order \( n \) and that an upper bound \( M \) for \( n \) is known, one finds with all three algorithms the coefficients of the characteristic polynomial of a minimal realization of \( \{s_k\}_{k=0}^{\infty} \) in about \( 2Mn \) operations. Then a minimal realization of \( \{s_k\}_{k=0}^{\infty} \) can immediately be given (theorems 0.1 and 0.3). If the coefficients of the characteristic polynomial were given, it would cost \( (2M-n-1)n \) operations to verify them. This shows that all three algorithms are almost optimal.

Unlike the algorithms of Massey/Berlekamp and the new algorithm, the algorithm of Rissanen is formulated as a decomposition algorithm for the infinite Hankel matrix associated with \( \{s_k\}_{k=0}^{\infty} \). We shall show that the algorithm of Massey/Berlekamp and the new algorithm may also be formulated as decomposition algorithms.

All three algorithms are numerically unstable; if all computations are performed with a finite precision of \( \eta \), then \( \eta \) independent of the magnitude of \( n \) or \( M \) it is possible that the obtained characteristic polynomial does not correspond exactly with a neighbouring impulse response of \( \{s_k\}_{k=0}^{\infty} \), and simultaneously, the obtained characteristic polynomial is not a numerical neighbour of the exact characteristic polynomial of \( \{s_k\}_{k=0}^{\infty} \). The proof of this will be founded upon the formulation of the algorithms as decomposition algorithms.

Since the algorithm of Rissanen is originally formulated as a decomposition algorithm, we shall not discuss the algorithms in the historical order but first treat the algorithm of Rissanen.

A point of criticism of all three algorithms which will not be stressed here, is that from a numerical point of view, a polynomial is certainly not characterized in the best way by its coefficients.
3.1. The algorithm of Rissanen

We consider an impulse response \( \{s_i\}_{i=0}^{\infty} \) that admits a finite order minimal realization. We assume that for the order \( n \) an upper bound \( N \) is known.

\( N \) denotes the Hankel matrix associated with \( \{s_i\}_{i=0}^{\infty} \).

First of all, let us give an outline of the algorithm of Rissanen without bothering for details.

The algorithm starts out with a pair \( k \) and \( t \) such that

1. \( r(H_{k,t}) = r(H_{k+1,t}) = k \)

and a decomposition for \( H_{k+1,t}r \):

2. \( H_{k+1,k+1} \quad H_{k+1,t} \quad P = R_{k+1,t} \),

where

- \( H_{k+1,k+1} \) is a unit lower triangular, \( (k + 1) \times (k + 1) \) matrix,
- \( P \) is a permutation matrix,
- \( R_{k+1,t} \) is an upper trapezoidal, \( (k+1) \times t \) matrix with \( R_{k+1,i,t} = 0 \) for \( 1 \leq i \leq k \) and \( R_{k+1,k+1,t} = 0 \).

From theorem 0.4 and 0.7, it follows that the coefficients of the characteristic polynomial of a partial minimal realization of \( \{s_i\}_{i=0}^{\infty} \) are found in the last row of \( H_{k+1,k+1} \) (e.g. \( H_{k+1,k+1} R_{k+1,t} = H_{k+1,t} \)).

If \( k + t \geq 2N \), we have found the coefficients of the characteristic polynomial of a minimal realization of \( \{s_i\}_{i=0}^{\infty} \) (theorem 0.6), and we are ready.

Otherwise, if \( k + t < 2N \), there are two possibilities.

Either (1) holds for all values of \( t' \), \( k \leq t' \leq 2N - k \), which is verified by extending (2) to a decomposition for \( H_{k+1,2N-k} \) and establishing that the last row of \( R_{k+1,2N-k} \) is zero, and then we are ready.

Or there exists \( t' > k \) such that

\( \Delta = r(H_{k,t'}) = r(H_{k+1,t'}) < r(H_{k+1,t'+1}) = k + 1 \)

and in that case \( k' = k' > k \) is determined so that

\( r(H_{k,t'+1}) = r(H_{k+1,t'+1}) = k' \),

which is the starting point again; the decomposition for \( H_{k',t'+1}r \) is obtained by updating the decomposition of \( H_{k+1,t'+1}r \) and the procedure is repeated. Since \( k' + t' > k + t \), the process on the whole is finite.
The algorithm (in pseudo-algol)

\begin{align*}
1 & \text{ $k := 1$; while } s_k = 0 \text{ do } k := k + 1; \\
2 & \text{ } s_k = 4; m := k + 1; \text{ (factorize } H_{k+1,4}) \\
3 & \text{ while } m < 2 \times N \text{ do } \\
4 & \begin{aligned}
& \text{ begin } k := k + 1; m := m + 1; \text{ (factorize } H_{k+1,4}) \\
& \text{ while } r(H_{k+1,4}) = k + 1 \text{ do } \\
& \text{ begin } k := k - 1; m := m + 1; \text{ (factorize } H_{k+1,4}) \\
& \end{aligned}
5 & \text{ end }
6 & \text{ end }
7 & \text{ end }
\end{align*}

We shall give a formal proof of the correctness of this algorithm. Viz., it is finite and it supplies in $k$ the order $n$ of a minimal realization of \{ $x_i, i = 1$ \}.

First of all the finiteness. The loop starting in line 5 can be executed at most \((4-k)\) times, because $r(H_{k+1,4}) = k + 1$ implies \(k + 1 \leq k\) and $k$ increases by one in any execution of the loop. Hence, also the loop starting in line 3 is finite, because $m$ increases at least by one in an execution of the loop. Consequently, the algorithm describes a finite computational process.

After execution of line 2, we have $k = t$, and $s_1 = s_2 = \ldots = s_{t-1} = 0$, but $s_t \neq 0$. So

\[
H_{k+1, t} = H_{k+1, t} - \begin{bmatrix}
\begin{array}{cccc}
& & & s_k \\
& \cdot & & \\
& & \cdot & \\
& & & \cdot
\end{array}
\end{bmatrix}, \text{ with } s_t \neq 0.
\]

Consequently, we then have

\[(1) \quad r(H_{k, t}) = r(H_{k+1, t}) = k.\]
Next, let us derive two invariant relations, one for the loop in line 5 and one for the loop in line 3.

Suppose that just before the execution of the while-statement in line 5, we have

\[ r(H_{k+1}, 1) = k. \]  

If \( r(H_{k+1}, 1) \neq k+1 \), then, obviously, (3) still holds after execution.

If \( r(H_{k+1}, 1) = k+1 \), then \( k := k+1 \), \( k \) is not changed. Hence, (3) holds after line 6 is executed. Consequently, (3) is an invariant relation for the while-statement in line 5.

Suppose that (1) holds just before the execution of the while-statement in line 3. If \( m = 2n \) then, obviously, (1) still holds after execution.

If \( m < 2n \), then after line 4 has been executed it holds that \( r(H_{k+1}, 1) = k \), implying that \( r(H_{k+1}, 1) = k \) (or (3)) holds just before line 5 is executed.

Because (3) is invariant we have after execution of the while-statement in line 5 that \( r(H_{k+1}, 1) = k \) and \( r(H_{k+1}, 1) \neq k \), which is equivalent with (1).

So (1) is an invariant relation for the while-statement in line 3.

We can now conclude the proof of the correctness.

Since (1) holds after execution of line 2 and (1) is an invariant, we have, eventually,

\[ m = k + \ell \geq 2 \times n \quad \text{and} \quad r(H_{k+1}, 1) = r(H_{k+1}, 1) = k. \]

Applying theorems 0.4 and 0.5, it follows that \( m = k \). So, the algorithm is correct, provided that we can factorize \( H_{k+1}, 1 \) and determine \( r(H_{k+1}, 1) \).

Next, let us turn to the factorization and the determination of the rank.

Since (1) is invariant, it follows that, whenever \( H_{k+1}, 1 \) is factorized in the computational process, we have \( r(H_{k+1}, 1) = k \).

**Theorem 1.**

If \( r(H_{k+1}, 1) = k \), then \( H_{k+1}, 1 \) has a decomposition:

\[ H_{k+1}, 1 = P \cdot R_{k+1}, 1, \]

where \( H_{k+1}, 1 \) is a unit lower triangular, \((k+1) \times (k+1)\) matrix, \( P \) is a permutation matrix.
$R_{k+1,k}$ is an upper trapezoidal, $(k+1) \times k$ matrix with $(R_{k+1,k})_{i,i} \neq 0$ for $1 \leq i \leq k$ and $(R_{k+1,k})_{k+1,k+1} \neq 0$ if $r(R_{k+1,k}) = k+1$. The last row of $R_{k+1,k}$ is zero if and only if $r(R_{k+1,k}) = k$.

The proof of this theorem is elementary and, therefore, omitted. It is obvious that with the decomposition (4), $r(R_{k+1,k})$ can be determined.

The determination of the decomposition of the form (4).

(i) In line 2.

Let $P$ be the $k \times k$ matrix \[
\begin{bmatrix}
0 & \cdots & 0 \\
1 & \cdots & 0
\end{bmatrix}.
\]

The matrix $R_{k+1,k}P$ is regular and has lower triangular form.

It is readily verified that the decomposition

\[
\begin{bmatrix}
t_0 \cdot (R_{k+1,k}P)^{-1} \\
t_1 & \cdots & t_k
\end{bmatrix} R_{k+1,k}P = \begin{bmatrix}
t_0 & 0 \\
0 & \cdots & t_k \\
0 & \cdots & 0
\end{bmatrix},
\]

where $t = (t_0, \ldots, t_k)^T$ is the unique vector such that $t^TR_{k+1,k} = 0^T$, satisfies the conditions of theorem 1.

(ii) In line 4.

We have $r(R_{k+1,k-1}) = r(R_{k+1,k}) = k$ and a decomposition for $R_{k+1,k}$ with the properties of (4):

\[
M_{k+1,k} R_{k+1,k-1} = R_{k+1,k}P = R_{k+1,k-1}.
\]

The last row of $R_{k+1,k-1}$ is zero.

For $R_{k+1,k}$ it applies that

\[
M_{k+1,k} R_{k+1,k} \begin{bmatrix}
P & 0 \\
O^T & 0
\end{bmatrix} = \begin{bmatrix}
R_{k+1,k-1} & M_{k+1,k} \begin{bmatrix}
t_k \\
\vdots \\
0
\end{bmatrix}
\end{bmatrix}.
\]

If the last component of $M_{k+1,k} \begin{bmatrix}
t_k \\
\vdots \\
0
\end{bmatrix}$ is zero, we have in (5) a decomposi-
tion for $R_{k+1,k}$ as in (4); otherwise, $R_{k+1,k}$ has $(K+1)$ linearly independent columns implying that $r(R_{k+1,k}) = K + 1$ and we obtain a decomposition for $R_{k+1,k}$ as in (4) by adapting the permutation matrix in (5) such that $R_{k+1,k}^{(k+1,k+1)} \neq 0$.

(iii) In line 6.

We have $r(R_{k,k}) = k$ and a decomposition for $R_{k,k}$ with the properties of (4):

$$W_{k,k} = R_{k,k}^{(k,k)} = k$$

The element $(R_{k,k}, k, k) \neq 0$.

For $R_{k,k}$ it applies that

$$R_{k,k} = \begin{bmatrix} \cdots & 0 & 1 \\ 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} R_{k+1,k} P = \begin{bmatrix} R_{k,k}^{(k,k)} \\ \cdots \\ c^{(k)} \end{bmatrix},$$

where $(m_{1}, \ldots, m_{k-1})$ denotes the last row of $W_{k,k}$.

Due to the Hankel structure of $R_{k+1,k}$, $(t-1)$ components of $c^{(t)}$ are also components of the last row of $R_{k+1,k}$; accordingly, $c^{(t)}$ has at least $(k-2)$ components zero.

If the matrix at the right-hand side of (6) is not upper trapezoidal, then it is brought into that form by premultiplying the left- and the right-hand side of (6) with elementary transformation matrices of the form

$$\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix},$$

where the multiplier $u$ is suitably chosen (Gaussian elimination).

Now, whether or not transformation was necessary, we have a decomposition

$$W_{k+1,k+1} = R_{k+1,k+1} P = R_{k+1,k+1}^{(k+1,k+1)}$$

with $W_{k+1,k+1}$ unit lower triangular, $R_{k+1,k+1}$ of upper trapezoidal form and the matrix consisting of the first $k$ rows of $R_{k+1,k+1}$ equal to $R_{k,k}$.

If the last row of $R_{k+1,k+1}$ is zero, then $r(R_{k+1,k+1}) = k$ and we have obtained a decomposition for $R_{k+1,k+1}$ as in (4); otherwise $r(R_{k+1,k+1}) = k + 1$ and we obtain a decomposition as in (4) by adapting $P$ such that $(R_{k+1,k+1}^{(k+1,k+1)}) \neq 0$. 72
At the end of the computational process, we have a decomposition for \( H_{k+1,1} \) with the same properties as \((6)\), we know that \((1)\) holds and that \( k + k \geq 2n \). Consequently, the last row of \( H_{k+1,1} \) is zero and with theorems 0.4, 0.6 and 0.7, it follows that the last row of \( H_{k+1,k+1} \) contains the coefficients of the characteristic polynomial of a minimal realization of \( \{ s_i \}_{i=1}^m \).

Remarks.

1. This algorithm is essentially the algorithm as it is proposed in [4]. However, there is a difference in presentation. In [4], Kismen presents his algorithm based on the coefficients of polynomials \( \Phi(t)\), which are defined as follows:

Let \( m(t) := \max \{ m \mid \{ s_i \}_{i=1}^m \) admits a realization of order \( t \}, \) then \( \Phi(t)\) is the characteristic polynomial of degree \( t \) of a realization of \( \{ s_i \}_{i=1}^m \).

It can be shown that in case \( t \leq n \), the \((t+1)\)-th row of the final matrix \( M_{n+1,n+1} \) contains the coefficients of \( \Phi(t)\).

2. The algorithm can be accelerated somewhat by replacing lines 5 and 6 by

\[
\textbf{5} \quad \text{if } t(\{ s_i \}_{i=1}^m) = k+1 \text{ then } k := 2; \\
\textbf{6} \quad m := k+1; \text{ (decompose } H_{k+1,k+1});
\]

(theorem 0.9).

Doing so, the algorithm transforms essentially into the algorithm of which theorem 0.10 states that it determines the order of a partial minimal realization of \( \{ s_i \}_{i=1}^m \).

3. It is not essential that \( M_{n+1,n+1} \) is unit lower triangular.

A decomposition

\[
M_{n+1,n+1} = H_{n+1,n+1} P = R_{n+1,n-2n}
\]

with \( R_{n+1,n-2n} \) upper trapezoidal, diagonal \( \neq 0 \) and last row zero, and \( H_{n+1,n+1} \) regular, is also suitable for finding the rank of \( H_{n+1,n-2n} \) and the coefficients of a characteristic polynomial.
3.3 The algorithm of Massey/Berlekamp

Again we consider an impulse response \( s_i \) that admits a finite order minimal realization and we assume that for the order \( n \) an upper bound \( N \) is known. With the algorithm of Massey/Berlekamp one can find the coefficients of the characteristic polynomial of a complete minimal realization. The algorithm does so by recursively determining the coefficients of polynomials \( P_m(\lambda) \).

**Definition 3.**

\( P(\lambda) \) is the characteristic polynomial of a partial minimal realization of \( \{ s_i \}_{i=1}^{\infty} \). \( P_0(\lambda) := 1 \).

The algorithm determines \( P_m(\lambda) \) for the consecutive values of \( m \):

\( m = 2k, 2k+1, \ldots, 2n \), where \( k = \min ( i \mid s_i \neq 0 ) \).

\( P_2(\lambda) \) is computed. If for some \( k \geq 2k \), \( P_{2k}(\lambda) \) is the characteristic polynomial of a minimal realization of \( \{ s_i \}_{i=\infty}^{\infty} \) but not of \( \{ s_i \}_{i=1}^{2k-1} \), then \( P_{2k+1}(\lambda) \) is constructed using \( P_{2k}(\lambda) \) and \( P_{2k}(\lambda) \).

In general, if \( P_{m+1}(\lambda) \neq P_m(\lambda) \), \( P_{m+1}(\lambda) \) is determined using \( P_m(\lambda) \) and the last polynomial in the sequence \( \{ P_{m}(\lambda) \}_{m=0}^{n} \) that has lower degree than \( P_m(\lambda) \).

Finally, \( P_{2n}(\lambda) \) is obtained. Since \( N \) is an upper bound of \( n \), it follows with theorem 0.6 that \( P_{2n}(\lambda) \) is the characteristic polynomial of a minimal realization of \( \{ s_i \}_{i=1}^{\infty} \).

The algorithm (in pseudo-algo)

1. \( P_0(\lambda) := 1 \)

2. \( k := 1; \) while \( s_k = 0, k = 1 \) do \( k := k + 1 \);

3. \( m := 2 \times k; \) (compute \( P_m(\lambda) \));

4. while \( m < 2 \times N \) do begin if \( P_m(\lambda) \) realizes \( \{ s_i \}_{i=1}^{\infty} \)

5. \( \quad \text{then } P_{m+1}(\lambda) := P_m(\lambda) \)

6. \( \quad \text{else (construct } P_{m+1}(\lambda) \);

7. \( m := m + 1 \)

8. end
The determination of $\mathcal{P}_{2k}(\lambda)$ (line 3).

If $k = \min \{ s_\lambda \mid i \neq 0 \}$, then $H_{s_\lambda, k}$ is a triangular matrix with non-zero diagonal.

Therefore, $r(H_{s_\lambda, k+1}) = r(H_{s_\lambda, k}) = 1$. With theorem 0.4, it follows that $[s_\lambda]_{k+1}^{2k}$ admits a partial minimal realization of order $k$. Theorem 0.7 shows that

$$
\mathcal{P}_{2k}(\lambda) = \lambda^k + x_1 \lambda^{k-1} + \ldots + x_1
$$

where $(x_1, \ldots, x_1)$ is the vector so that

$$(x_1, \ldots, x_1)H_{s_\lambda, k+1} = 0^T.
$$

This triangular system of equations is easy to solve; the solution is unique.

The determination of $\mathcal{P}_{m+1}(\lambda)$ (lines 5 and 6).

Let

$$
\mathcal{P}_m(\lambda) = \lambda^q + x_1 \lambda^{q-1} + \ldots + x_1
$$

since $m \geq 2k > \deg \{ \mathcal{P}_{2k}(\lambda) \} = k$, it follows with theorem 0.12 that $m > q$.

Applying theorem 0.7, we obtain

$$(7) \quad (x_1, \ldots, x_1)H_{s_{m+1}^{s_\lambda}} = (0, \ldots, 0, 1).
$$

If $d = 0$, then $\mathcal{P}_m(\lambda)$ realizes $(s_{m+1}^{s_\lambda})$ and we can take $\mathcal{P}_{m+1}(\lambda) := \mathcal{P}_m(\lambda)$.

Let us suppose that $d \neq 0$, so $\mathcal{P}_{m+1}^{s_\lambda}(\lambda) \neq \mathcal{P}_m(\lambda)$.

Let of the sequence $(\mathcal{P}_0(\lambda), \mathcal{P}_{2k}(\lambda), \mathcal{P}_{2k+1}(\lambda), \ldots, \mathcal{P}_{m+1}(\lambda))$

$$
\mathcal{P}_t(\lambda) = \lambda^p + y_1 \lambda^{p-1} + \ldots + y_1
$$

be the last polynomial of lower degree than $\mathcal{P}_m(\lambda)$.

If $t = 0$, then $p = 0$, $q = 1$ and, thus,

$$(y_1, \ldots, y_1)H_{1, q} = (0, \ldots, 0, c), \quad \text{with } c = s_\lambda \neq 0.
$$

If $t \neq 0$, then, with theorem 0.11, it follows that $c = t - p = 1$ since $\mathcal{P}_t(\lambda) \neq \mathcal{P}_{t+1}(\lambda)$, we obtain from theorem 0.7 that also in case $t \neq 0$,

$$(8) \quad (y_1, \ldots, y_1)H_{s_\lambda, q} = (0, \ldots, 0, c), \quad \text{with } c \neq 0.
$$
It follows from theorem 0.11 that degree \((P_{m+1}(\lambda)) = \max(q_{m+1} - q)\).

We distinguish two cases:

(i) \(q > m - q + 1\) or degree \((P_{m+1}(\lambda)) = q\).

From (8) we obtain by inspection:

\[
\begin{pmatrix}
\varepsilon_1 & \ldots & \varepsilon_{m-q+1} \\
\vdots & \ddots & \vdots \\
0 & \ldots & 0 \\
\end{pmatrix}
\begin{pmatrix}
\varepsilon_{q-m} \\
\varepsilon_q \\
\vdots \\
\varepsilon_{q+p-m+1} \\
\varepsilon_{p+q} \\
\vdots \\
\varepsilon_{q/1} \\
\varepsilon_{2m+1}
\end{pmatrix} = (0, \ldots, 0, c)
\]

So, because of (7) and (8):

\[
\{(x_1, \ldots, x_{q+1}) = d^{-1}(0, \ldots, 0, y_1, \ldots, y_{p+1}, 0, \ldots, 0)\} \cap (m-q+1, c) = 0^2.
\]

According to theorem 0.7, we have

\[
P_{m+1}(\lambda) = P_m(\lambda) - d^{-1} \lambda^{2q-m+1} P_{\xi}(\lambda)
\]

(ii) \(q \leq m - q + 1\) or degree \((P_{m+1}(\lambda)) = m - q - 1\).

From (8) we obtain

\[
(Y_1, \ldots, Y_{p+1}, 0, \ldots, 0) \cap (m-q+1, c) = (0, \ldots, 0, c)
\]

and from (7)

\[
\{(0, \ldots, 0, x_1, \ldots, x_{q+1}) = d^{-1}(0, \ldots, 0, d)\} \cap (m-q+1, 2q) = (0, \ldots, 0, d)
\]

Similarly as in case (i) it follows that

\[
P_{m+1}(\lambda) = d^{-1} \lambda^{2q} P_m(\lambda) - d^{-1} P_{\xi}(\lambda)
\]

Remark.

As one not might have expected, it is possible that \(P_{m+1}(\lambda) \neq P_m(\lambda)\) although degree\((P_{m+1}(\lambda)) = degree(P_m(\lambda))\) (case (i)).
1.3. A new algorithm

This algorithm is also concerned with a recursive determination of the coefficients of \( P_m(\lambda) \), the characteristic polynomial of a partial minimal realization of \( \{s_i\}_{i=1}^m \).

The essential difference with the algorithm of Massey/Berlekamp is that this algorithm determines only those elements of the set

\((P_0(\lambda), P_1(\lambda), P_{2\lambda}(\lambda), \ldots, P_m(\lambda))\)

that are unique (according to theorem 0.11, \( P_m(\lambda) \) is unique if \( m \geq 2 \) deg \( P_m(\lambda) \)).

As in the algorithm of Massey/Berlekamp, we need a fictitious polynomial \( P_0(\lambda) := 1 \).

The algorithm (in pseudo-algol):

1. \( \kappa := 0; P_0(\lambda) := 1; \)
2. \( \ell := 1; \) while \( s_\ell = 0 \) do \( \ell := \ell + 1; \)
3. \( \kappa := \ell; m := \kappa + \ell; \) (compute \( P_{\ell}(\lambda) \));
4. while \( m < 2 \times \ell \) do if \( r_{\ell+1, \ell+1} = \ell + 1 \)
   then begin \( k := \ell + 1; \) \( \ell := \ell + 1; \) \( m := k + \ell; \)
   (construct \( P_{\ell}(\lambda) \))
   end
5. else begin \( \ell := \ell + 1; \) \( m := m + 1 \) end

The algorithm is essentially the same as the algorithm of which theorem 0.11 states that it determines the order of a partial minimal realization of \( \{s_i\}_{i=1}^m \), with the exception that at lines 3 and 5 the determination of \( P_m(\lambda) \) is inserted. In the proof of that theorem we showed that after execution of line 3 we have

\( r(\lambda_{\ell+1, k}) = r(\lambda_{\ell+1, \ell+1}) = \ell \)

and that this relation is not changed afterwards.

Applying theorem 0.4, it follows that if in the computational process \( P_m(\lambda) \) has to be determined, then \( \{s_i\}_{i=1}^m \) admits a unique minimal realization of
order \( k \); the characteristic polynomial of that realization has degree \( k \) and coefficients \( (x_1, \ldots, x_k) \), where \( (x_1, \ldots, x_k) \) is the unique vector such that

\[
(x_1, \ldots, x_k) H_{k+1, \ell} = 0^T.
\]

With theorem 0.7, it follows that \( r(H_{n+1, \ell}) = \ell + 1 \) if and only if \( P_{\ell}(s) \) does not realize \( \{z_i\} \).

The determination of \( P_{\ell}(s) \).

(i) In line 3. See the algorithm of Massey/Berlekamp.

(ii) In line 6.

Any change of \( k \) in the computational process is followed by the evaluation of a characteristic polynomial, which as a consequence of the foregoing has degree \( k \).

It is obvious that if \( k \) changes in line 5, \( k \) has assumed at least two preceding values.

Let \( p < q < r \) be three subsequent values for \( k \) and let the corresponding polynomials of degrees \( p, q \) and \( r \) be, respectively,

\[
P_{q}(\lambda) := \lambda^p + y_p \lambda^{p-1} + \ldots + y_1, \\
P_{r}(\lambda) := \lambda^q + x_q \lambda^{q-1} + \ldots + x_1, \\
P_{n}(\lambda) := \lambda^r + \ldots .
\]

We shall show how \( P_{q}(\lambda) \) can be computed using \( P_{q}(\lambda) \) and \( P_{r}(\lambda) \).

If \( p = 0 \), then \( q = 1 \), and

\[
(y_1, \ldots, y_p) H_{p+1, q} = (0, \ldots, 0, c) \quad \text{with } c = a_1 \neq 0 .
\]

If \( p > 0 \), then it follows with theorem 0.7 that also then

\[
(y_1, \ldots, y_{p+1}) H_{p+1, q} = (0, \ldots, 0, c) \quad \text{with } c \neq 0 .
\]

since, otherwise, after the value of \( p \), \( k \) cannot assume the value of \( q \).

Similarly, we obtain

\[
(x_1, \ldots, x_q) H_{q+1, r} = (0, \ldots, 0, d) \quad \text{with } d \neq 0 .
\]

Let \( X \) be the \((r-q+2) \times (r+1)\) matrix
\[
\begin{pmatrix}
\mathbf{X} := \begin{bmatrix}
x_1 & \ldots & x_q \\
\vdots & \ddots & \vdots \\
x_r & \ldots & x_q \\
y_1 & \ldots & y_p \\
0 & \ldots & 0 \\
0 & \ldots & 0
\end{bmatrix}
\end{pmatrix}^{r-q+1}.
\]

Then, by inspection,

\[
X_{X+1} = \begin{bmatrix}
0 & \ldots & \ast \\
\vdots & \ddots & \vdots \\
0 & \ldots & 0 \\
0 & \ldots & 0
\end{bmatrix} := \mathbf{Y}.
\]

Let \( \mathbf{v} \) be the unique vector so that \( \mathbf{v}_1 = 1 \) and \( \mathbf{v}_2^T = \mathbf{0}^T \). Then, if \( \mathbf{w}^T := \mathbf{v}^T \mathbf{X} = (\mathbf{w}_1, \ldots, \mathbf{w}_p, 1) \), we have \( \mathbf{w}^T H_{X+1} = \mathbf{0}^T \).

Applying theorem 0.7, it follows that

\[
P_m(\lambda) = \lambda^{r} + \lambda^{r-1} + \cdots + 1 = (\lambda^{r-q} + \lambda^{r-q-1} + \cdots + \lambda^{r-q+1}) P_{m_2}(\lambda) + \mathbf{v}_2 \mathbf{v}_2^T M_1(\lambda).
\]

1.4. The algorithms of 3.2 and 3.3 as decomposition algorithms for the Hankel matrix

In the foregoing we gave three algorithms for finding a minimal realization of an impulse response \( (\mathbf{s}_{ij})_{i,j=1}^n \). Unlike the algorithm of Massey/Bezukamp and the new algorithm we proposed in 3.3, the algorithm of Rissanen is formulated as a decomposition algorithm for the infinite Hankel matrix \( \mathbf{H} \) associated with \( (\mathbf{s}_{ij})_{i,j=1}^n \). Since all three algorithms supply the rank of the Hankel matrix \( \mathbf{H} \) (according to theorem 0.1), this rank is equal to the order of a minimal realization of \( (\mathbf{s}_{ij})_{i,j=1}^n \), it is to be expected that the algorithms of 3.2 and 3.3 can also be formulated as decomposition algorithms.

In the following we shall prove that the algorithm of Massey/Bezukamp is a decomposition algorithm for \( \mathbf{H} \). For the algorithm of 3.3 the same can be shown; however, this proof goes very similarly and will be omitted.
Theorem 3.
Let \( z := \min\{i \mid z_i \neq 0\} \) and \( z \leq n \). If for some \( m \) with \( 2k \leq m \leq 2k + 1 \) the algorithm of Massay/Boleskamp has supplied the set of characteristic polynomials \( \{P_0(\lambda), P_2(\lambda), P_{2k+1}(\lambda), \ldots, P_m(\lambda)\} \), then a factorization can be constructed

\[
M_{t,p+1} R_{t,m-p} = \mathbb{R}_{t,m-p},
\]

where

\( p = \deg(P_m(\lambda)) \),

\( t = \min(p+1, m+1-p) \),

\( M_{t,p+1} \) is a \( t \times (p+1) \) matrix of full row rank,

\( R_{t,m-p} \) is a \( t \times (m-p) \) matrix of upper trapezoidal form with \( (R_{t,m-p})_{i,i} \neq 0 \) for \( 1 \leq i \leq t-1 \) and \( s^T R_{t,m-p} = 0^T \).

The last row of \( M_{t,p+1} \) contains the coefficients of \( P_m(\lambda) \).

Proof.
The proof is rather tedious and lengthy. First, we introduce some auxiliary notation.

\[
\{P_i(\lambda)\}_{i=1}^k \text{ is the subset of } \{P_0(\lambda), P_2(\lambda), P_{2k+1}(\lambda), \ldots, P_m(\lambda)\} \text{ obtained by discarding } P_i(\lambda), 2k + 1 \leq i \leq m-1, \text{ whenever } \deg(P_i(\lambda)) = \deg(P_{i+1}(\lambda)).
\]

Except perhaps for \( P_m(\lambda) \), \( \{P_i(\lambda)\}_{i=1}^k \) contains precisely those \( P_i(\lambda) \) that are uniquely determined (Theorem 0.11).

We have by definition

\[
t_1 = 0, \quad t_2 = 2k, \quad \ldots, t_{k-1} = m-1 \text{ and } t_k = m
\]

and, since \( P_{2k}(\lambda) \) is unique, \( P_{2k}(\lambda) = P_{2k}(\lambda) \).

Let

\[
P_i(\lambda) = \lambda^{n_i} + x_{i+1} \lambda^{n_{i+1}} + \cdots + x_k \lambda^k
\]

It is readily seen that

\[
n_1 = 0, \quad n_2 = 1, \quad n_{i+1} > n_i \quad (1 \leq i < k)
\]
and it follows from theorem 0.11 that

\[(11) \quad x_{i+1} = x_i + 1 - n_i \quad (2 \leq i < k).\]

Since \( m \geq t_{k-1} + 1 \), it shows from \( (11) \) that

\[(12) \quad m \geq n_k + n_{k+1} \quad (1 \leq i < k-1).\]

Because \( F_k \) realizes \( \{a_i\}_{i=1}^{c_k} \) but not \( \{a_i\}_{i=1}^{c_{k-1}} \) \( \{a_i\}_{i=1}^{c_{k-1}} \quad (1 < i \leq k-1) \), we must have (theorem 0.7)

\[(13) \quad (x_{1,1}, \ldots, x_{1,n_{k+1}})^T H_{k+1, n_{k+1}} = (0, \ldots, 0, c_i), \quad c_i \neq 0 \quad (1 \leq i \leq k-1)\]

(The validity of (13) in case \( i = 1 \) is readily verified.)

The rest of the proof is a matter of putting two and two together.

Let \( 1 \leq i < k-1 \). Because \( n_i + 1 > n_{i+1} \), we can define the \((n_{i+1} - n_i) \times (n_{i+1} - 1)\) matrix

\[
X_i = \begin{bmatrix}
0 & x_{i,1} & \cdots & x_{i,n_i} \\
\vdots & \ddots & \ddots & \vdots \\
x_{i,1} & \cdots & x_{i,n_i} & 0 \\
0 & \cdots & \cdots & 0 \\
\end{bmatrix}
\]

Because \( m-n_k > n_{k+1} \), we obtain applying (13)

\[(14) \quad X_i H_{k+1,m-n_k} = \begin{bmatrix}
0 & \cdots & \cdots & 0 \\
c_i & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & c_i \\
\end{bmatrix}, \quad 1 \leq i < k-1, \quad c_i \neq 0 .\]

(Note that (14) does not hold for \( i = k-1 \), unless \( m-n_k \leq n_k \).)
Let \( i = k - 1 \).

We define

\[
X_{k-1} = \begin{pmatrix}
\vdots & \vdots \\
\epsilon_{k-1} & X_{k-1,1} & \cdots & X_{k-1,n_{k-1}} & 1 & 0 & \cdots & 0 \\
\end{pmatrix}
\min(n_{k-1}, m-n_{k-1})
\]

Since \( m-n_{k-1} \geq n_{k-1} \), we have \( \min(n_{k-1}, m-n_{k-1}) \geq 0 \).

If \( m-n_{k-1} = n_{k-1} \), then \( X_{k-1} \) is an "empty" matrix.

If \( m-n_{k-1} \geq n_{k-1} \), then \( X_{k-1} \) is comparable with \( X_i \), \( 1 \leq i < k-1 \).

Applying (13) it follows that

(15) \[
X_{k-1} H_{k+1, m-n_k} = \begin{pmatrix}
\epsilon_{k-1} & \cdots & \cdots & \cdots \\
\epsilon_{k-1} & \cdots & \cdots & \cdots \\
\epsilon_{k-1} & \cdots & \cdots & \cdots \\
\end{pmatrix}
\min(n_{k-1}, m-n_{k-1})
\]

Having completed all preparations we now see by inspection that

(16) \[
\begin{pmatrix}
X_1 \\
\vdots \\
X_{k-1} \\
X_k \\
\vdots \\
X_{n_k} \\
\end{pmatrix}
H_{k+1, m-n_k} = \begin{pmatrix}
\hat{R}_{k+1, m-n_k} \\
\end{pmatrix}
\]

where \( \hat{R}_{k+1, m-n_k} \) has upper trapezoidal form with non-zero diagonal and last row zero.

We have

\[
s = \alpha_2 + (\alpha_3 - \alpha_2) + \cdots + (\alpha_{k-1} - \alpha_{k-2}) + \min(n_{k-1}, m-n_{k-1}, m-n_{k-1}) + 1
\]

\[
= \min(n_{k-1}, m-n_{k-1}) + 1
\]

The matrix at the left of \( \hat{R}_{k+1} \) in (16) is by definition \( M_{k+1} \).
Since

\[
\begin{bmatrix}
X_{k+1} & \ldots & X_{k,n_k} \\
X_{k-1} & & \\
\vdots & & \\
X_1 & & \\
& & \ddots \\
& & & X_1
\end{bmatrix}
= \begin{bmatrix}
* & \ldots & * \\
\vdots & & \vdots \\
\vdots & & \vdots \\
* & \ldots & 0 \\
\vdots & & \vdots \\
& & \ddots \\
& & & 0
\end{bmatrix}
\]

it holds that \( M_{t,n_k+1} \) has full row rank. \( \Box \)

**Corollary 4.**
If \( t = p + 1 = n_k + 1 \), then the factorisation is a decomposition and \( M_{t,n_k+1} \) is unit lower triangular up to a permutation of its rows. \( \Box \)

**Theorem 3.**
The algorithm of Massey/Barilekamp starts essentially with the factorization of theorem 3 for \( m = 2t \), and repeatedly updates the factorization for \( m \) to a factorization for \( m+1 \), until \( m = 2N \).

**Proof.**
Let us suppose that \( P_{2t}(\lambda), P_{2t+1}(\lambda), \ldots, P_m(\lambda) \) are available, but \( P_{m+1}(\lambda) \) not.

If \( P_m(\lambda) \) realises \( \{ s_i \}_{i=1}^{m+1} \), then \( P_{m+1}(\lambda) = P_m(\lambda) \) and the factorization of theorem 3 for \( m+1 \) can easily be obtained.

If \( P_m(\lambda) \) does not realize \( \{ s_i \}_{i=1}^{m+1} \), then \( P_{m+1}(\lambda) \neq P_m(\lambda) \).

Let \( \delta = \text{degree}(P_m(\lambda)) \). From theorem 0.1, we obtain

\[
q = \text{degree}(P_{m+1}(\lambda)) = \max(p, m+2-p) .
\]

Consequently,

\[
m+2-p \leq q \text{ or, } m+2-q \leq p < q+1 .
\]

Therefore,

\[
t' = \min(q+1, m+2-q) = m+2-q .
\]

Except for the last row of \( M_{t',q+1} \) that should contain the still unknown coefficients of \( P_{m+1}(\lambda) \), we can now construct the factorization of theorem 3 for \( M_{t',q+1} \):

\[
M_{t',q+1} = \begin{bmatrix}
H_{q+1,m+1-q} & H_{q+1,m+1} - q = \delta + 2-q, m+1-q
\end{bmatrix}
\]
If, provisionally, we take for the last row of $H_{n+1}$, the shifted row:
\[(0, \ldots, 0, x_{k+1}, \ldots, x_{n+1})\]
\[-q-p\]
where the $x_{i}$ denote the coefficients of $P_m(\lambda)$, then
\[
P_{m+2-q, n+1-q} = \begin{bmatrix}
1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 1
\end{bmatrix}
\]

Premultiplying with an elementary transformation matrix of the form
\[
\begin{bmatrix}
1 & \cdots & 0 \\
0 & \cdots & 1
\end{bmatrix}
\]
with suitably chosen multiplier $u$, annihilates the last row of $H_{n+1}$ and transforms the last row of $H_{n+1}$ into the coefficients of $P_{m+1}(\lambda)$. This is exactly the way in which the algorithm of Massey/Berlekamp obtains $P_{m+1}(\lambda)$.

From theorem 5 it follows that the algorithm of Massey/Berlekamp can be formulated as a decomposition algorithm for the infinite Hankel matrix associated with \(\{s_i\}_{i=1}^\infty\).

In 3.2 we saw that, at the end of the algorithm, \(m = 2k\) and degree \((P_{m}(\lambda)) = n\). Applying theorem 3, we obtain that \(t = \min(n+1, 2N+1-n) = n+1\) \((N \geq n)\).
Consequently, the factorization for $H_{n+1, 2N-n}$ is a decomposition with $H_{n+1,n+1}$ (corollary 4) unit lower triangular up to a permutation of its rows.

**Theorem 5.**
Let $k = \min(i \mid s_i \neq 0)$ and $n \leq N$. If for some $m$, with $2k \leq m \leq 2N$, the algorithm of 3.3 has supplied the set of polynomials \(\{P_i(\lambda)\}_{i=1}^k\) of degree respectively: $p_1 = 0$, $p_2 = 1$, $p_3 = \ldots$, $p_k$, and if \(\{s_i\}_{i=1}^m\) is realized by \(P_i(\lambda)\) but not by \(P_k(\lambda)\), then a decomposition can be constructed:

\[
H_{k+1, n+1} = H_{k+1, n+1} = R_{k+1, n+1}
\]

where
\( M_{n_k+1, n_k+1} \) is a regular \((n_k+1) \times (n_k+1)\) matrix that is unit lower triangular up to a permutation of its rows,

\( R_{n_k+1, m-n_k} \) is a \((n_k+1) \times (m-n_k)\) matrix of upper trapezoidal form with

\[
(R_{n_k+1, m-n_k})_{i, i} = 0 \quad \text{for} \quad 1 \leq i \leq n_k \quad \text{and} \quad (R_{n_k+1, m-n_k})_{i, i} = 0^T.
\]

The last row of \( M_{n_k+1, n_k+1} \) contains the coefficients of \( P_k(\lambda) \).

**Theorem 7.**

The algorithm of 3.3 starts essentially with the decomposition of theorem 6 for \( m = 2N \), and repeatedly updates the decomposition for \( m \) to a decomposition for a greater \( n \), until \( m \geq 2N \).

From theorem 7, it follows that also the algorithm of 3.3 can be formulated as a decomposition algorithm for the infinite Hankel matrix. In the updating procedure of this algorithm we have always decompositions, due to the fact that only characteristic polynomials are supplied that are uniquely determined.

The algorithms of Risén and Massey/Bezlekamp, as well as the algorithm of 3.3 employ the structure of a Hankel matrix in their updating procedures by taking, provisionally, for the still unknown rows of a new \( M_{n, m} \) shifted rows of a former \( M_{n, m} \).

### 3.5. The numerical stability of the algorithms

#### 3.5.1.

In section 3.4 we saw that the algorithms are decomposition algorithms for the Hankel matrix associated with \((a_i)_{i=1}^N\), i.e., finally a decomposition

\[
P_1^T M_{n+1, n+1} P_2 = R_{n+1, m-n}
\]

is supplied, where \( M_{n+1, n+1} \) is unit lower triangular, the \( P_1 \)'s are permutation matrices and \( R_{n+1, m-n} \) is upper trapezoidal with \((R_{n+1, m-n})_{i, i} \neq 0 \) for \( 1 \leq i \leq n \) and \((R_{n+1, m-n})_{i, i} = 0^T \) for \( i \geq 2N \) with \( N \) an upper bound for \( n \).

In the case of the algorithm of Risén, \( P_1 \) is the identity matrix; in the case of the other two algorithms, \( P_2 \) is the identity matrix.

All these algorithms find this decomposition in a recursive way.

Counting divisions and multiplications, it can be shown that the total number of operations is for each of the algorithms about \( m^2 \); the algorithm of 3.3
being somewhat faster than the others. This is a factor n less than other decomposition algorithms require; triangularization by Gaussian elimination, for example, would cost \(n^3N = \frac{1}{3}n^3\) operations. The reason that so few computations are needed, is that the algorithms exploit the Hankel structure.

This is, however, also the main reason for the numerical instability.

3.5.2.

Since the algorithms have the same characteristics, we treat only the algorithm of Mistress.

If the algorithm of Mistress is implemented on a machine in the way it is descibed in 3.1, then it will certainly deliver the wrong value for \(A\). In 3.1, the rank of \(H_{k+1}\) is taken to be \(k\), if \(e_{k+1}^T R_{k+1} e = 0\) and taken to be \(k+1\), otherwise. However, in the computational process, we, generally, have that \(e_{k+1}^T R_{k+1} e = f(e_{k+1}^T R_{k+1} e) \neq 0\), even if \(e_{k+1}^T R_{k+1} e = 0\). Therefore, we shall arrange that in any approximate decomposition \(\overline{R}_{k+1} = \overline{R}_{k+1} \preceq \overline{R}_{k+1}\) that may occur, it holds that

\[
\begin{align*}
\text{either } & (\overline{R}_{k+1})_{11} = 0, \quad \text{or } \| (\overline{R}_{k+1})_{11} \| > \eta \| \overline{R}_{k+1} \|, \\
\text{either } & (\overline{R}_{k+1})_{11} = 0, \quad \text{or } \| (\overline{R}_{k+1})_{11} \| > \eta \| \overline{R}_{k+1} \|. 
\end{align*}
\]

Let \(\eta\) be an a priori given upper bound for the order \(n\) of a minimal realization. If performed without round-off errors, the algorithm of Mistress corresponds with a mapping \(\varphi\) from the input set

\[
\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_N
\]

into the output set

\[
\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \ldots \cup \mathcal{R}_N,
\]

where \(\mathcal{P}_k (1 \leq k \leq N)\) is the set of all impulse responses that admit a minimal realization of order \(k\) and where \(\mathcal{R}_k (1 \leq k \leq N)\) is the set

\[
\mathcal{R}_k = \{ x : x \in \mathbb{R}^{N+1}, x_{k+1} \neq 0, x_{k+2} = \ldots = x_{N+1} = 0 \}.
\]

If \(x \in \mathcal{R}_k\), then \(x_1 + x_2 \lambda + \ldots + x_{N+1} \lambda^{N}\) is the characteristic polynomial of a minimal realization of some impulse response \((e_1)_n \in \mathcal{P}_k\).

So, \(\varphi\) has the property that \(\varphi(\mathcal{P}_k) \subseteq \mathcal{R}_k\).
If performed with round-off errors, the algorithm of Rissanen corresponds
with a mapping \( f(\epsilon) \) from \( \mathcal{D} \) into \( \mathbb{R} \). Unlike \( \epsilon \), \( f(\epsilon) \) has not the property
that \( f(\epsilon)(\mathcal{D}_\epsilon) \subseteq \mathcal{R}_k \).

A metric in \( \mathcal{D} \) is defined by:

\[
\|((s_{1}^{(1)})_{1=1}^{m}, (s_{1}^{(2)})_{1=1}^{m}) - ((s_{1}^{(1)})_{1=1}^{2N}, (s_{1}^{(2)})_{1=1}^{2N})\|_1^2 = \left( \frac{1}{2N} \sum_{i=1}^{2N} (s_{1}^{(1)} - s_{1}^{(2)})^2 \right)^{1/2}.
\]

Note that, if for some \( k \) \((0 \leq k < 2N)\) \( \mathcal{H}_{k+1,2N-k} \) is a Hankel block associated
with \((s_{1}^{(1)}), \ldots, (s_{1}^{(2)})_{1=1}^{2N-k} \), then

\[
\|((s_{1}^{(1)})_{1=1}^{m}, (s_{1}^{(2)})_{1=1}^{m}) - ((s_{1}^{(1)})_{1=1}^{2N}, (s_{1}^{(2)})_{1=1}^{2N})\|_2^2 = \|((s_{1}^{(1)})_{1=1}^{m}, (s_{1}^{(2)})_{1=1}^{m}) - ((s_{1}^{(1)})_{1=1}^{2N}, (s_{1}^{(2)})_{1=1}^{2N})\|_1^2.
\]

A metric in \( \mathbb{R} \cup \{0\} \) is defined by

\[
\|r_1 - r_2\|_2 = \|r_1\|_2 \quad \text{where} \quad r_1, r_2 \in \mathbb{R} \cup \{0\} \in \mathbb{R}^{N+1} \quad (l = 1,2).
\]

Now, applying the notations and definitions of chapter 1, the algorithm of
Rissanen is numerically stable on \( \mathcal{D} \) if it is backward stable on \( \mathcal{D} \):

\[
3 \quad C_{1} > 0 \quad \forall d \in \mathcal{D} \quad \exists \gamma > 0 \quad \forall \epsilon \in \mathcal{D} \quad \|f(\epsilon)(d) - f(d')\|_2 \leq C_{1} \|\epsilon\|_2
\]

or, if it is forward stable on \( \mathcal{D} \):

\[
3 \quad C_{2} > 0 \quad \forall d \in \mathcal{D} \quad \exists \gamma > 0 \quad \forall \epsilon \in \mathcal{D} \quad \frac{\|f(\epsilon)(d) - f(d)\|_2}{\|f(d)\|_2} \leq C_{2} \gamma
\]

We shall investigate whether the algorithm of Rissanen is numerically stable
on \( \mathcal{D} \).

Let \( N = 2 \) and let we consider \( d \in \mathcal{D} \):

\[
(s_{1}^{(1)})_{1=1}^{m} = [\epsilon, 1, 0, \ldots, 0] \quad \text{with} \quad 0 < \epsilon < 1.
\]

We assume that the unspecified elements of \((s_{1}^{(1)})_{1=1}^{m} \) are so chosen that \((s_{1}^{(1)})_{1=1}^{m} \)
admits a complete minimal realization of order 1 (from theorem 0.4 it follows
that this may be done: \( s_{1}^{(1)} = (s_{1}^{(1)} - \epsilon s_{0}^{(1)}) s_{0}^{(1)} - \epsilon s_{0}^{(1)} \) for \( l = 1,2 \).

Applying the algorithm of Rissanen to \( d \), we obtain with exact arithmetic
(see 3.1):

\[
\|f(\epsilon)(d) - f(d)\|_2 \leq C_{1} \|\epsilon\|_2
\]

\( \epsilon := 1; \ k := 1; \ m := 2; \) the decomposition for \( \mathcal{H}_{k+1,2N-k} = \mathcal{H}_{2,1} \) is:

\[
\begin{pmatrix}
1 & 0 \\
-\epsilon^{-1} & 1
\end{pmatrix}^{\mathcal{H}_{2,1}} = \begin{pmatrix}
\epsilon \\
0
\end{pmatrix}.
\]

87
k := 2; m := 3; the decomposition for $H_{k+1,k} = H_{2,2}$ is:
\[
\begin{bmatrix}
1 & 0 \\
-\varepsilon^{-1} & 1
\end{bmatrix}
\begin{bmatrix}
\varepsilon & 1 \\
0 & a
\end{bmatrix}, \text{ with } a = 1 - \varepsilon^{-1}.
\]

$r(H_{k+1,k}) = k + 1 = 2$;
k := 2; $n := 4$; the decomposition for $H_{k+1,k} = H_{3,2}$ is obtained as follows:
\[
\begin{bmatrix}
1 & 0 & 0 \\
-\varepsilon^{-1} & 1 & 0 \\
0 & -\varepsilon^{-1} & 1
\end{bmatrix}H_{3,2} =
\begin{bmatrix}
\varepsilon & 1 \\
0 & a \\
0 & -\varepsilon^{-1} - \varepsilon^{-1}
\end{bmatrix}.
\]

Premultiplying by the elementary transformation matrix
\[
T_1 =
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & -\alpha & 1
\end{bmatrix}
\]
gives
\[
\begin{bmatrix}
1 & 0 & 0 \\
-\varepsilon^{-1} & 1 & 0 \\
0 & -\varepsilon^{-1} & 1
\end{bmatrix}H_{3,2} =
\begin{bmatrix}
\varepsilon & 1 \\
0 & a \\
0 & -\varepsilon^{-1} - \varepsilon^{-1}
\end{bmatrix}.
\]

Premultiplying by the elementary transformation matrix
\[
T_2 =
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & (\alpha^{-1} + \varepsilon^{-1}) & 1
\end{bmatrix}
\]
gives the decomposition for $H_{3,2}^*$:
\[
\begin{bmatrix}
1 & 0 & 0 \\
-\varepsilon^{-1} & 1 & 0 \\
-(\alpha^{-1} + \varepsilon^{-1})a^{-1} - \varepsilon^{-1} & (\alpha^{-1} + \varepsilon^{-1})a^{-1} - \varepsilon^{-1} & 1
\end{bmatrix}H_{3,2}^* =
\begin{bmatrix}
\varepsilon & 1 \\
0 & a \\
0 & 0
\end{bmatrix}.
\]

The computational process is stopped, since $m = 2n$; the characteristic polynomial of a minimal realization of $(s_1^{n-1})$ is determined by $x \in E_2$:
\[
x = (-\varepsilon^{-1} + \varepsilon^{-1})a^{-1} - \varepsilon^{-1}, (\alpha^{-1} + \varepsilon^{-1})a^{-1} - \varepsilon^{-1})^T = \begin{pmatrix} -1 \\ \varepsilon^{-1} \\ \varepsilon^{-1} \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}^T.
\]

88
We shall study the effect of a single round-off error in the computation of \( \alpha \):

\[
\tilde{\alpha} = \alpha(1 + \eta), \quad 0 < \eta < 1.
\]

We assume that \( \eta \) is so small that the algorithm can be applied to the given impulse response on any machine with relative precision \( \eta \) without a breakdown in the computational process.

We then obtain

\[
\tilde{x} = \left( -\left( \tilde{\alpha}^{-1} + \epsilon^{-1} \right) \tilde{\alpha}^{-1} \epsilon^{-1} \right) \tilde{x}, \quad \left( \tilde{\alpha}^{-1} + \epsilon^{-1} \right) \tilde{x} = \epsilon^{-1} \tilde{x}
\]

and one readily verifies that

\[
(20) \quad \tilde{x} - x = \eta \left( -\frac{1}{\tilde{\alpha}^{-1} \epsilon^{-1}} \tilde{\alpha}^{-1} \epsilon^{-1} \tilde{x} \right) = \frac{\eta}{\tilde{\alpha}^{-1} \epsilon^{-1}} \tilde{x}.
\]

If the algorithm of Rissanen is forward stable, then it should be possible to give an upper bound for

\[
\frac{\|\tilde{x} - x\|}{\|x\|} \leq \eta
\]

that is independent of \( \epsilon \) (see (19)). However, \( \|\tilde{x} - x\| \) contains a term of the order of \( \epsilon^{-2} \) whereas \( \|x\| \) is bounded. Since \( \epsilon \) may be arbitrarily small it is impossible to give an upper bound independent of \( \epsilon \). So, the algorithm of Rissanen is not forward stable on \( \mathbb{D} \).

Next, we shall investigate whether the algorithm of Rissanen is backward stable on \( \mathbb{D} \).

If the algorithm is backward stable on \( \mathbb{D} \), then the obtained \( \tilde{x} \) corresponds exactly with a numerical neighbour \( \{ \tilde{x}_1 \} \) of the given input \( \{ x_1 \} \). So, in our example, we must then have that for some \( \tilde{\eta}_2 \) and for all \( \epsilon \) smaller than some \( \eta_0 \),

\[
(21) \quad \tilde{x}_1 - x_1 = 0 \quad \text{and} \quad \frac{\|\tilde{x}_1 - x_1\|_2}{\|x_1\|_2} < C_1 \eta
\]

where \( C_1 \) is independent of \( \tilde{\eta}_2 \) and \( \eta \).

Let \( \eta \) be chosen so small that, in the following analysis, we may account for higher order terms in \( \eta \) by multiplying the first order terms with a factor 1.1. Let \( \tilde{x} = (\tilde{x}_1, \tilde{x}_2, 0) = \tilde{x} - x \) and let \( \tilde{\eta}_{2,2} = \tilde{\eta}_2 \) and \( \tilde{\eta}_2 = \tilde{\eta}_{3,2} \) and suppose that the algorithm is backward stable. It follows from (21) that then
\[ 0^T = x^T \Delta H_{3,2} + \Delta x^T H_{3,2} = x^T \Delta H_{3,2} + (\Delta x_1, \Delta x_2)(H_{2,2} + \Delta H_{2,2}) \]

and, therefore, 
\[ \frac{\|\Delta x\|_2}{\|x\|_2} \leq \|\Delta H_{3,2}\|_2 \|H_{2,2}\|_2 \cdot \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \] .

Using \[ \frac{1}{\sqrt{2}} \|\Delta H_{3,2}\|_2 \leq 1 \cdot H_{2,2} \leq \sqrt{2} \|\Delta H_{3,2}\|_2 \] and (21), we obtain
\[ \frac{\|\Delta H_{3,2}\|_2}{\|x\|_2} \leq 2G_2 \|\Delta H_{3,2}\|_2 \leq 4C_1 \|H_{2,2}\|_2 \] (say),

and, hence
\[ \frac{\|\Delta x\|_2}{\|x\|_2} \leq 4C_1 \cdot C_2(H_{2,2}) . \]

Since \[ C_2(H_{2,2}) < 5 \], if \[ 0 < \varepsilon < 1 \], it follows that 
\[ \frac{\|\Delta x\|_2}{\|x\|_2} < 22C_1C_2 \]

is independent of \( \varepsilon \).

However, it is a contradiction with (20) that 
\[ \frac{\|\Delta x\|_2}{\|x\|_2} \] has an upper bound that is independent of \( c \). Therefore, our supposition that the algorithm of Rissanen is backward stable, is false.

Summarizing, it follows that the algorithm of Rissanen and, similarly the algorithm of Massey/Berlekamp and the algorithm of 3.1, is numerically unstable on \( D \).

3.5.3.

One reason for the instability, which is intuitively clear, is that, in updating the decomposition, nonstabilized elementary transformation matrices may occur. In the example in 3.5.2 with which we proved that the algorithms are numerically unstable, the transformation matrix \( T_1 \), as well as the transformation matrix \( T_2 \), are nonstabilized (if \( 0 < \varepsilon < 1 \), then \( |ae^{-1}| \) and \( |ae^{-1} + \varepsilon 0_{n-1}| \) are both greater than 1). These nonstabilized transformations need not occur if the demand that \( H_s^* \) should be unit lower triangular is dropped. It is not necessary that \( H_s^* \) should be unit lower tri-
angular (see the last remark of 3.1). Therefore, let us adapt the algorithm of Rissanen so that only stabilized transformations occur and investigate whether the adapted algorithm is numerically stable on D.

Again let $N = 2$ and consider the impulse response

$$\{s_i\}_{i=1}^{\infty} = \{e, 1, 0, \ldots\}, \quad 0 < e < 1.$$  

We apply the adapted algorithm to this sequence. We obtain, successively, with exact arithmetic:

$k = 1; m = 2; l = 1$; the decomposition for $H_{k+1,s} = H_{2,s}$ is:

$$
\begin{bmatrix}
0 & 1 \\
-1 & e
\end{bmatrix}
\begin{bmatrix}
1 \\
0
\end{bmatrix}.
$$

$k = 2; m = 3; l = 2$; the decomposition for $H_{k+1,s} = H_{2,s}$ is:

$$
\begin{bmatrix}
0 & 1 \\
-1 & e
\end{bmatrix}
\begin{bmatrix}
1 & 1 \\
0 & e
\end{bmatrix}, \quad \text{where } e = e - 1 \quad (|e| < 1).
$$

$k = 2; m = 4; l = 3$; the decomposition for $H_{k+1,s} = H_{2,s}$ is obtained as follows:

$$
\begin{bmatrix}
0 & 1 & 0 \\
-1 & e & 0 \\
0 & -1 & e
\end{bmatrix}
\begin{bmatrix}
1 & 1 \\
0 & e \\
1 & -e
\end{bmatrix}.
$$

Premultiplying by the stabilized transformation matrix

$$T_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -e & 0 & 1 \end{bmatrix}$$

gives

$$
\begin{bmatrix}
0 & 1 & 0 \\
-1 & e & 0 \\
0 & -e & e
\end{bmatrix}
\begin{bmatrix}
1 & 1 \\
0 & e \\
0 & -e & 1
\end{bmatrix}.
$$

Premultiplying by the stabilized transformation matrix

91
\[
\mathbf{T}_4 = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & -\frac{\beta-1}{\beta} & 1
\end{pmatrix}
\]
gives the desired decomposition for \( \mathbf{N}_{3,2} \):
\[
\begin{pmatrix}
0 & 1 & 0 \\
-1 & 0 & 0 \\
-\frac{\beta-1}{\beta} & -\frac{\beta-1}{\beta} & 1 - \frac{\beta-1}{\beta}
\end{pmatrix} \mathbf{N}_{3,2} = \begin{pmatrix} 1 & 0 \\
0 & \frac{\beta}{\beta-1} \\
0 & 0 \end{pmatrix}.
\]

The computational process is stopped, since \( m = 2 \lambda \); the characteristic polynomial of a minimal realization is determined by \( z \in \mathbb{R}_2 \):
\[
z = \left( -\frac{\beta-1}{\beta} , -\frac{\beta-1}{\beta} - \beta - 1 , c \right)^T = \left( -\frac{\beta}{\beta-1} , \frac{\beta}{\beta-1} , c \right)^T.
\]

We shall study the effect of a single round-off error in the computation of \( \bar{\beta} \):
\[\bar{\beta} = \beta(1+\eta), \quad 0 < \eta < \frac{1}{2} .\]
We assume that \( \eta \) is so small that the algorithm can be applied to the given impulse response on any machine with relative precision \( \eta \) without a breakdown in the computations.

We obtain
\[
z = \left( -\frac{\beta+1}{\beta} , -\frac{\beta+1}{\beta} - \beta - 1 , c \right)^T
\]
and one readily verifies that
\[
z - x = \eta(\frac{1}{(c-1)(1+\eta)} , \frac{-c}{(c-1)(1+\eta)} + 1 - c , 0)^T.
\]
It follows that
\[
\frac{\|z-x\|_2}{\|z\|_2} \sim \frac{1}{\|z\|_2} \eta \sim \eta^{-1},
\]
which is an important improvement compared to 3.5.2, but still it can be shown, as in 3.5.2, that the algorithm is numerically unstable on \( \mathbb{R} \).

The obvious reason is the use of the trick: take preliminary, as last row for a new \( \mathbf{N}_2 \), the shifted last row of the preceding \( \mathbf{N}_2 \). In our example we see that this trick gives \( \mathbf{N}_{3,3} \) a condition number of about \( \epsilon^{-1} \).
Remark.
In [3], Rissanen proposes a recursive algorithm to find a decomposition
\[ H_{k+1, k} P = L_{k+1, k+1} R_{k+1, k} \]
where \( P \) is a permutation matrix, \( L_{k+1, k+1} \) is unit lower triangular and \( R_{k+1, k} \) is upper trapezoidal.
In that article he does not exploit the Hankel structure. He updates the decomposition by Gaussian elimination in case \( H_{k+1, k} \) is extended with a row or a column. Since he does not use a pivoting technique and, consequently, it is not possible to bound pivots away from zero, also this algorithm can be shown to be numerically unstable.

3.6. The approach to be followed in chapter 4
In chapter 4 we shall propose an algorithm for finding a minimal realization of \( \{ a_k \}_{k=1}^\infty \) in a recursive way. This algorithm will be formulated as a decomposition algorithm for the Hankel matrix associated with \( \{ a_k \}_{k=1}^\infty \).
It will be shown that the algorithm is numerically stable.
From 3.5 it follows that it is necessary for numerical stability that in the updating procedure of an approximate decomposition
\[ H_{k+1, k+1}, R_{k+1, k} P \sim \bar{H}_{k+1, k} \]
nonstabilized transformation matrices do not occur; nor may the Hankel structure be exploited in the way it is done by the algorithms discussed in this chapter. This implies that the algorithm will need \( O(n^2) \) operations.
In 3.5 we did not pay much attention to the fact that the problem to determine the rank is, numerically, an ill-posed problem (chapter 2). That this is so, implies that we may not expect - even if the algorithm can be shown to be numerically stable - that the finally supplied value of \( n \) is the order of a minimal realization of \( \{ a_k \}_{k=1}^\infty \). Therefore, the new algorithm determines not the rank, but, instead, the \( c \)-stable rank of the successive Hankel blocks. We shall then be able to prove that the finally supplied value of \( n \) is equal to the lowest possible order of minimal realizations of numerical neighbors of \( \{ a_k \}_{k=1}^\infty \).
In chapter 5, it will be shown how this numerically stable algorithm can be used to approximate a given impulse response by one that admits a lower order minimal realization.
4. A NUMERICALLY STABLE, RECURSIVE, MINIMAL REALIZATION ALGORITHM

4.1. The algorithm

The algorithm is essentially the algorithm of Rissanen as it is discussed in 3.1. However, instead of the rank of $N_{k+1,t}$ the $\varepsilon$-stable rank of $N_{k+1,k}$ is determined and another decomposition method is used. In pseudo-algol the algorithm is:

1. (compute $k$ and $t$ such that $r_{\varepsilon}(N_{k+1,k}) = r_{\varepsilon}(N_{k+1,t}) = k$, where $c_1 = \text{tol} \cdot \|H_{k,t}\|_2$ and $c_2 = \text{tol} \cdot \|H_{k+1,t}\|_2$;)
2. $m := k + t$;
3. while $m < 2 \cdot N$ do
4. begin $t := t + 1; m := m + 1$; (decompose $N_{k+1,t}$);
5. $\varepsilon := \text{tol} \cdot \|H_{k+1,m}\|_2$;
6. while $r_{\varepsilon}(N_{k+1,m}) = k + 1$ do
7. begin $k := k + 1; m := m + 1$; (decompose $N_{k+1,m}$);
8. $\varepsilon := \text{tol} \cdot \|H_{k+1,m}\|_2$;
9. end
10. end

Assuming that the algorithm is applied to a non-trivial impulse response $(s_j)_{j=1}^w$ that admits a minimal realization of order $n$ and assuming that all computations are performed exactly, we shall show in this section that the algorithm is correct, i.e.,

under certain conditions for $N$ and tol, the finally supplied value of $k$ is equal to $n$ and with the decomposition of the final Hankel block $N_{k+1,k}$ a minimal realization triple for $(s_j)_{j=1}^w$ can be given.

In subsection 4.1.1 we show that the computational process is finite and that the finally supplied value of $k$ is equal to $n$, provided that $N \geq n$, that tol is small enough and provided that $r_{\varepsilon}(N_{k+1,k})$ can be determined from the decomposition for any pair $k$ and $t$ occurring in the computational process.
In subsection 4.1.2, we discuss two decomposition methods and show that, indeed, if \( r \) is small enough, \( r_c(H_{k+1}, \varepsilon) \) can be determined from the decomposition of \( H_{k+1} \).

In conclusion, in subsection 4.1.3 we show that with the decomposition of the final Hankel block \( H_{k+1} \), a minimal realization triple for \( \{ s_i \}_{i=1}^m \) can be given.

4.1.1.

Let us suppose that \( N \) is an a priori known upper bound for \( n \).

We define:

\[
C_0 := \max\{ C_z(H_{k+1, \varepsilon}) \mid 1 \leq k \leq t, k + \varepsilon \leq 2N + 1, \ r(H_{k, \varepsilon}) = k \}
\]

(1) \[
C_1 := \max\{ C_z(H_{k+1, \varepsilon}) \mid 1 \leq k \leq t, k + \varepsilon \leq 2N, \ r(H_{k, \varepsilon}) = k \}
\]

**Theorem 1.**

If \( 1 \leq k \leq t, k + \varepsilon \leq 2N + 1, 0 < \text{tol} + C_0 \leq \varepsilon \) and \( e = \text{tol} + \| H_{k, \varepsilon} \|_2 \), then

\[
r(H_{k, \varepsilon}) = k = r_c(H_{k, \varepsilon}) = k.
\]

**Proof.**

Since \( r(H_{k, \varepsilon}) \geq r_c(H_{k, \varepsilon}) \), we have \( r_c(H_{k, \varepsilon}) = k \) implies \( r(H_{k, \varepsilon}) = k \).

Let \( r_c(H_{k, \varepsilon}) = k \). The definition of \( C_0 \) implies that

\[
C_0 \geq C_z(H_{k, \varepsilon}) = \sigma_1(H_{k, \varepsilon}) \cdot \sigma_k^{-1}(H_{k, \varepsilon}).
\]

\( \sigma_1(H_{k, \varepsilon}) \geq \sigma_2(H_{k, \varepsilon}) \geq \ldots \geq \sigma_k(H_{k, \varepsilon}) \) denote the non-trivial singular values of \( H_{k, \varepsilon} \).

Consequently,

\[
\sigma_k(H_{k, \varepsilon}) \geq C_0^{-1} \sigma_1(H_{k, \varepsilon}) = C_0^{-1} \| H_{k, \varepsilon} \|_2 - (C_0 \text{tol})^{-1} \varepsilon > 2\varepsilon.
\]

Applying theorem 2.10, we obtain that \( r_c(H_{k, \varepsilon}) = k \).

**Theorem 2.**

If \( 1 \leq k \leq t, k + \varepsilon \leq 2N, 0 < \text{tol} + \max\{ C_0, C_1 \} \leq \varepsilon \) and \( e = \text{tol} + \| H_{k+1, \varepsilon} \|_2 \), then

\[
r(H_{k, \varepsilon}) = k \text{ implies } r_c(H_{k+1, \varepsilon}) = r(H_{k+1, \varepsilon}).
\]
Proof.
Let \( r_k(z) = k \), then either \( r_k(z_{k+1,z}) = k \) or \( r_k(z_{k+1,z}) = k+1 \).
If \( r_k(z_{k+1,z}) = k \), it follows from the definition of \( C_1 \) that
\[
C_1 \cap C_2(z_{k+1,z}) = C_1 \cap C_2(z_{k+1,z}) = \sigma_1(z_{k+1,z})^{-1} (z_{k+1,z})
\]
and thus
\[
\sigma_1(z_{k+1,z})^{-1} (z_{k+1,z}) = C_1 \cap C_2(z_{k+1,z}) = C_1 \cap C_2(z_{k+1,z}) = \sigma_1(z_{k+1,z})^{-1} (z_{k+1,z})
\]
because \( \sigma_1(z_{k+1,z}) = 0 \), it follows with theorem 2.10 that \( r_k(z_{k+1,z}) = k \).
If \( r_k(z_{k+1,z}) = k+1 \), it follows with theorem 1 that \( r_k(z_{k+1,z}) = k+1 \).

We assume that \( 0 < \text{cop} \max \{ C_0 \} \ < 1 \) and that, if for some pair \((k,z)\) in the computational process we have to determine \( r_k(z_{k+1,z}) \) with \( c = \text{cop} \|H_{k+1,z}\|_2 \), this may be done using the decomposition of \( H_{k+1,z} \).

Let \( t = \min \{ | \|H_{k+1,z}\|_2 \|_2 \} \). As in section 3.1 we have \( r_k(z_{k+1,z}) = r_k(z_{k+1,z}) = k \).
It follows that \( k < n \). Let \( k = \), then \( k + \leq 2n \). Applying theorems \( \text{cop} \) and \( \text{cop} \^2 \) we obtain that
\[
(*) \quad \sigma_1(z_{k+1,z})^{-1} (z_{k+1,z}) = k \quad \text{and} \quad k \leq t, k + t \leq 2n
\]
where \( \sigma_1(z_{k+1,z})^{-1} (z_{k+1,z}) = k \quad \text{and} \quad k \leq t, k + t \leq 2n
\]
The finiteness of the computational process is proved similarly as in 3.1.

Let us suppose that just before the execution of the while-statement in line 6 we have a pair \((k,t)\) such that
\[
(**) \quad r_k(z_{k+1,z}) = r_k(z_{k+1,z}) = k \quad \text{and} \quad k < \epsilon, k + \epsilon \leq 2N
\]
Since then \( r_k(z_{k+1,z}) = k \), theorem 2 shows that
\[
\sigma_1(z_{k+1,z})^{-1} (z_{k+1,z}) = \text{cop} \|H_{k+1,z}\|_2
\]
If \( r_k(z_{k+1,z}) = k \), we have after execution of the while-statement that (applying theorem \( \text{cop} \)) the formula (*) hold.
If \( r_k(z_{k+1,z}) = k + 1 \), then theorem 0.9 part (ii) shows that
\[
r_k(z_{k+1,z}) = k + 1 \quad \text{for} \quad k + 1 < t \leq t
\]
Hence we have \( t + \bar{t} \leq 2n \) for \( k + 1 \leq t \leq l \). Applying theorem 1 we obtain that
\[
\tau_k(H_{t,\bar{t}}) = t \quad \text{with} \quad c = tol \times \|H_{t,\bar{t}}\|_2 \quad \text{for} \quad k + 1 \leq t \leq l.
\]

It is now easily verified that after the execution of the while statement in line 6 the formula's (**) hold.

We saw that (**) holds just before the execution of the while statement in line 3.

If \( m = k + \bar{t} = 2n \), then (**) holds after the execution.

If \( m = k + \bar{t} < 2n \), then \( \bar{t} = t + 1 \) (line 4) and, thus, \( k < \bar{t} \) and \( k + \bar{t} < 2n \).

Applying Theorems 1 and 2, it follows from (**) that (after \( \bar{t} := t + 1 \)) we have \( \tau_k(H_{k,t}) = \tau_k(H_{k+1,t}) = k \). Hence (**) holds just before the execution of line 6. Consequently, after the execution of line 9, we have that (**) holds.

So, (**) is an invariant relation for the while statement in line 3. Hence, at the end of the computational process (applying Theorems 1 and 2) we have
\[
k + \bar{t} = 2n \quad \text{and} \quad \tau_k(H_{k,t}) = \tau_k(H_{k+1,t}) = k.
\]

From Theorems 0, 4 and 0.6 it follows that then \( k = n \).

4.1.2.

**Theorem 3.**

If \( \tau_k(H_{k,t}) = k \), then \( H_{k+1,t} \) has a decomposition
\[
(2) \quad H_{k+1,t} = H_{k+1,t} P = P H_{k+1,t},
\]
where \( H_{k+1,t} \) is a regular \((k+1) \times (k+1)\) matrix, \( P \) is a permutation matrix, \( H_{k+1,t} \) is an upper trapezoidal, \((k+1) \times t\) matrix with \( (R_{k+1,t})_{t+1,1} \neq 0 \) for \( 1 \leq t \leq k \) and \( (R_{k+1,t})_{k+1,k+1} \neq 0 \) if \( \tau_k(H_{k+1,t}) = k + 1 \).

The last row of \( H_{k+1,t} \) is zero if and only if \( \tau_k(H_{k+1,t}) = k \). \( \square \)

In Chapter 3 we saw that the decomposition method should have the property that the decomposition of a Hankel block can be updated with relatively few computations, whenever the Hankel block is enlarged with a column (as in line 4 of the algorithm) or is enlarged with a row (as in line 7). Furthermore, it is a demand that nonstabilized elementary transformation matrices do not occur in the updating process and that the Hankel structure be not
exploited in the way as it is done by the algorithms in chapter 3.
These demands rule out a number of decomposition methods: for instance,
Gaussian elimination because, if it is made recursive, nonstabilized ele-
mentary transformations may occur, Gram-Schmidt orthogonalization,
Householder orthogonalization and singular value decomposition, because
these three cannot be performed in a recursive way.

The first decomposition method.
Let, after execution of line 2, $H_{k+1,1}^p$ have a decomposition such as in (2).
(i) The updating process in line 4.
We have $r(H_{k+1,2}^{-1}) = k$ and a decomposition for $H_{k+1,2}^{-1}$ as in (2):

$$H_{k+1,k+1|H_{k+1,2}^{-1}}^p = R_{k+1,2}.$$  

Consequently,

$$H_{k+1,k+1|H_{k+1,2}^{-1}}^p \begin{pmatrix} P & 0 \\ 0^T & 1 \end{pmatrix}^{-1} = \begin{pmatrix} R_{k+1,2} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \varphi^T \\ \varphi \end{pmatrix}.$$  

If the last row of the matrix at the right-hand side is not zero, the per-
mutation matrix is adapted so that the element with the largest modulus in
the last row stands in the position $(k+1),(k+1)$. Doing so, a decomposition
for $H_{k+1,k+2}$ is obtained such as in (2).

(ii) The updating process in line 7.
We have $r(H_{k,k}^{-1}) = k$ and a decomposition for $H_{k,k}^{-1}$ as in (2):

$$H_{k,k}^{-1} P = R_{k,k}$$

with

$$(R_{k,k})_{k,k} \neq 0.$$  

Furthermore, we have

$$H_{k+1,k+2} F = \begin{pmatrix} R_{k,k} \\ \varphi \end{pmatrix} \begin{pmatrix} \varphi^T \\ \varphi \end{pmatrix}.$$  

99
The matrix at the right-hand side is transformed into upper trapezoidal form by premultiplying the left- and the right-hand side of (a) by
\[ T(k+1) := T(k,k+1) \ldots T(1,k+1), \]
where
\[
T(i,k+1) = \begin{bmatrix}
I & 0 \\
0 & \ddots & 0 \\
0 & 0& \ddots & \ddots \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & b_i \\
\end{bmatrix}^{k+2-i}
\]
or
\[
T(i,k+1) = \begin{bmatrix}
I & 0 \\
0 & \ddots & 0 \\
0 & 0& \ddots & \ddots \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & b_i \\
\end{bmatrix}^{k+2-i}
\]
(1 ≤ i ≤ k).

The multipliers \( u_{ij} \) are so chosen that
\[
T(i,k+1) \ldots T(1,k+1) \begin{bmatrix}
F_{k+1} \\
\vdots \\
(\alpha_{k+1}, \ldots, \alpha_k)^T \\
\end{bmatrix} = \begin{bmatrix}
\ast & \ast & \cdots & \ast \\
\ast & \ast & \cdots & \ast \\
\ast & \ast & \cdots & \ast \\
\ast & \ast & \cdots & \ast \\
0 & 0 & \cdots & 1 \\
\end{bmatrix}
\]
(1 ≤ i ≤ k).

That form of \( T(i,k+1) \) is chosen for which the multiplier \( u_{ij} \) is contained in \( i \)-th row. Consequently, the \( T(p,q) \) are stabilized elementary transformation matrices. If at the end of this process the transformed matrix at the right-hand side of (4) has non-zero last row, then the permutation matrix \( \Gamma \) is adopted such that the element with the largest modulus in the last row stands in the position \( (k+1), (k+1) \). The regularity of
\[
H_{k+1,k} := T(k+1) \begin{bmatrix}
\gamma_{k+1} & 0 \\
\vdots & \vdots \\
\alpha_{k+1} & 0 \\
\end{bmatrix}^{k+2-i} T^T
\]
follows from the regularity of \( H_{k,k} \) and \( T(k+1) \). Consequently, we have obtained a decomposition for \( H_{k+1,k} \) of the same kind as (2).
A second decomposition method.

Let after the execution of line 2, $H_{k+1,k}$ have a decomposition such as in (2) with $H_{k+1,k+1}$ orthogonal.

The updating of the decomposition in line 4 of the algorithm proceeds along the same lines as in the first decomposition method. Since in that updating process $H_{k+1,k}$ does not change, $H_{k+1,k+1}$ is still orthogonal after the updating.

In line 7 of the algorithm, we have $\tau(H_{k+1,k}) = k$ and a decomposition for $H_{k+1,k}$ such as in (2):

$$M_{k,k} R_{k,k} \mathbb{P} = R_{k,k}$$

with $\langle R_{k,k}, R_{k,k} \rangle \neq 0$ and $M_{k,k}$ orthogonal.

Furthermore, we have

$$M_{k+1,k} = \begin{bmatrix} M_{k+1,k} & 0 \\ 0^\top & 1 \end{bmatrix}, \quad \mathbb{P} = \begin{bmatrix} R_{k+1,k} \\ (\hat{\theta}_{k+1}, \ldots, \hat{\theta}_{k+1})^\top \end{bmatrix}$$

The matrix at the right-hand side is transformed into upper trapezoidal form by premultiplying the left- and the right-hand side of (5) by

$$T_{i,k} = T(1,k+1), \ldots, T(i,k+1),$$

where

$$T(1,k+1) = \begin{bmatrix} 1 & \vdots & \vdots & \vdots \\ \\
\vdots & 1 & \vdots & \vdots \\ \\
\vdots & \vdots & \ddots & \vdots \\ \\
\vdots & \vdots & \vdots & 1 \\ 
0 & \cdots & 0 & 1 \end{bmatrix}_{k+2-L}, \quad (1 \leq i \leq k).$$

The rotation angles $\phi_i$ are chosen such that

$$T(1,k+1) \cdots T(i,k+1) \left[ \begin{array}{c} R_{k,k} \\ (\hat{\theta}_{k+1}, \ldots, \hat{\theta}_{k+1})^\top \end{array} \right] = \begin{bmatrix} \ast & \ast & \ast & \ast & \ast & \ast \\ \\
\vdots & \ast & \ast & \ast & \ast & \ast \\ \\
\vdots & \vdots & \ast & \ast & \ast & \ast \\ \\
\vdots & \vdots & \vdots & \ast & \ast & \ast \\ \\
0 & 0 & 0 & \ast & \ast & \ast \\ \\
0 & 0 & 0 & 0 & \ast & \ast \end{bmatrix} (1 \leq i \leq k).$$
The \( T(p,q) \) are Givens matrices or plane rotations (in the \( p-q \) plane).

If the last row of the matrix \( T^{(k+1)} \) is not zero, then the permutation matrix \( P \) is adapted such that the largest element of the last row stands in the position \((k+1)_y(k+1)\).

The (regularity and) orthogonality of

\[
M_{k+1,k+1} = T^{(k+1)} \begin{bmatrix} 
M_{k+1,k+1} & 0 \\
0^T & 1 
\end{bmatrix}
\]

follows from the (regularity and) orthogonality of \( M_{k+1,k+1} \) and \( T^{(k+1)} \).

Summarizing, we have obtained a decomposition for \( N_{k+1,k+1} \) such as in (2).

The number of operations.

We compare the number of operations that the decomposition methods require for the decomposition of a \( k \times k \) matrix \( A \); we assume that the principal minors of \( A \) are unequal to zero so that for the permutation matrix \( P \) the identity matrix can be taken. Let \( A_1 \) be the \( i \times i \) matrix

\[
A_1 = \begin{bmatrix} (A)_{1,1} & \cdots & (A)_{1,i} \\
\vdots & \ddots & \vdots \\
(A)_{i,1} & \cdots & (A)_{i,i} 
\end{bmatrix} \quad (1 \leq i \leq k),
\]

and let us suppose that with any one of the decomposition methods we may obtain the decomposition

\[
(6) \quad M_1 A_1 = R_1
\]

with \( M_1 \) a regular \( i \times i \) matrix and \( R_1 \) an upper triangular \( i \times i \) matrix.

The matrix \( M_1 \) is the product of a strictly ordered sequence of transformation matrices \( T(p,q) \) where \( 1 \leq p < q \leq i \). \( T(p,q) \) affects only the \( p \)-th and the \( q \)-th row of the matrix to which it is applied. Since there is only one transformation matrix \( T(p_0,q_0) \) in this sequence if \( 1 \leq p_0 < q_0 \leq i \) and since any transformation matrix is further characterized by two
numbers (a plane rotation by $\cos(q)$ and $\sin(q)$); a stabilized elementary transformation matrix by the multiplier $y$ and a number indicating whether the multiplier is on the diagonal or not), it is possible to store the information concerning all transformation matrices in an $i \times i$ array without actually computing $N_k$. It is obvious that this technique of storing $N_k$ in product form may result in a considerable saving in the number of operations. If the matrices $N_k$ are stored in product form, the approximate number of operations for updating (6) to a decomposition for $A_{i+1}$ (counting divisions and multiplications) is with the first decomposition method

\[ \frac{i}{2} \text{ for the evaluation of } x_{i+1} = \frac{1}{2} \begin{pmatrix} (A)_{1,i+1} \\ \vdots \\ (A)_{i,i+1} \end{pmatrix} \]

\[ \times \frac{i}{2} \text{ for the evaluation of } \tau_{i+1} = \begin{pmatrix} \frac{B_{i+1}}{C_{i+1}} & R_{i+1} \\ C_{i+1} & A \end{pmatrix} \]

The second decomposition method (which from now on will be referred to as recursive Givens decomposition) requires $i^2$ operations for updating (6) to a decomposition for $A_{i+1}$.

Consequently, the first decomposition method requires asymptotically $\frac{1}{2}k^3$ operations and recursive Givens decomposition $\frac{1}{3}k^3$ operations.

Remark.

If the matrices $N_k$ are actually computed, the first decomposition method will require $\frac{5}{6}k^3$ operations, and the second one $\frac{7}{3}k^3$ operations.

The number of operations needed by recursive Givens decomposition can be reduced. The product of a plane rotation and a vector (requiring 4 multiplications when evaluated as

\[ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \cos(q) & \sin(q) \\ -\sin(q) & \cos(q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} ca_1 + sa_2 \\ -sa_1 + ca_2 \end{bmatrix} \]

\[ c = \cos(q), \quad s = \sin(q) \]

can be evaluated with three multiplications if one switches partly to the tangent of the half rotation angle $r = \tan(\frac{1}{2} q)$. 

103
Since
\[ \tau = \frac{1 - c}{c} = \frac{c}{1 + c} \]
and, thus,
\[ c = 1 - \tau, \quad s = \tau + \tau c \]
it follows that
\[ b_1 = ca_1 + sa_2 = a_1 + s(a_2 - a_1) = a_2^\tau + c(a_1 + a_2) = (a_1 + b_2) + a_2 \]
\[ b_2 = -ca_1 + sa_2 = -a_1 + c(a_2 - a_1) = a_2 - a(a_1 + a_2) = -(a_1 + b_1) + a_2, \]
Consequently, there are four ways to evaluate \( b_1 \) and \( b_2 \) with three multiplications:

\[
\begin{align*}
(7) & \quad b_1 = a_1 + s(a_2 - a_1) & b_1 = a_2 \tau + c(a_1 + a_2) \\
(8) & \quad b_2 = -a_1 + c(a_2 - a_1) & b_2 = a_2 = s(a_1 + a_2) \\
(9) & \quad b_1 = a_1 c + a_2 s & b_1 = s(a_1 + b_2) + a_1 \\
(10) & \quad b_2 = -a_1 s + a_2 c & b_2 = -a_1 s + a_2 c
\end{align*}
\]

Applying (7), (8), (9) or (10), recursive Givens decomposition requires asymptotically \( k^3 \) operations for decomposing the matrix \( A \), which is still three times more than the first method requires. The formulas (9) and (10) can also be found in Cull, Golub, Murray and Saunders [2].

Remark.
With this modification it is necessary to store three parameters of each plane rotation: \( c, s \) and \( \tau \).

Another way to reduce the number of operations of recursive Givens decomposition is due to Gentleman [1] and Hammarling [3]. Let us suppose that we have a matrix stored in product form

\[
\begin{bmatrix}
d_1 & 0 \\
0 & d_2
\end{bmatrix}
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1k} \\
a_{21} & a_{12} & \cdots & a_{2k}
\end{bmatrix}
\]

104
and we want to premultiply this matrix with a plane rotation. Because, if 
$c = \cos(\theta)$, $s = \sin(\theta)$ and $t = \tan(\theta)$, we have

$$
\begin{pmatrix}
  c & s \\
-\frac{s}{c} & c
\end{pmatrix}
\begin{pmatrix}
  a_1 & 0 \\
  0 & a_2
\end{pmatrix}
= 
\begin{pmatrix}
  a_{11} & \frac{a_{21}}{a_{11}} \\
-\frac{a_{21}}{a_{11}} & a_{11}
\end{pmatrix}
\begin{pmatrix}
  1 & \frac{d_1}{d_2} \\
-\frac{d_1}{d_2} & 1
\end{pmatrix},
$$

it costs asymptotically $2k$ operations to obtain the new product form

$$
\begin{pmatrix}
  c d_1 & 0 \\
  0 & c d_2
\end{pmatrix}
\begin{pmatrix}
  (a_{11} + \frac{d_1}{d_2} a_{21}) & \ldots & (a_{1k} + \frac{d_1}{d_2} a_{2k}) \\
-\frac{d_1}{d_2} a_{11} + a_{21} & \ldots & -\frac{d_1}{d_2} a_{1k} + a_{2k}
\end{pmatrix}.
$$

Applying this technique, recursive Givens decomposition requires asymptotically $\frac{1}{2}n^2$ operations, which is twice as much as the first decomposition method.

Remarks.

1. There are eight ways to write
$$
\begin{pmatrix}
  c & s \\
-\frac{s}{c} & c
\end{pmatrix}
\begin{pmatrix}
  a_1 & 0 \\
  0 & a_2
\end{pmatrix}
$$
as a product of a new diagonal matrix and a matrix $X$ so that the multiplication of $X$ and
$$
\begin{pmatrix}
  a_{11} & \ldots & a_{1k} \\
  a_{21} & \ldots & a_{2k}
\end{pmatrix}
$$
costs asymptotically $2k$ operations. These eight ways are determined by the following forms of $X$:

$$
X = 
\begin{pmatrix}
  1 & * \\
  * & 1
\end{pmatrix}, 
\begin{pmatrix}
  1 & * \\
  * & 1
\end{pmatrix}, 
\begin{pmatrix}
  1 & * \\
  * & 1
\end{pmatrix}, 
\begin{pmatrix}
  1 & * \\
  * & 1
\end{pmatrix}
$$

and

$$
\begin{pmatrix}
  a & * \\
  * & 1
\end{pmatrix}, 
\begin{pmatrix}
  a & * \\
  * & 1
\end{pmatrix}, 
\begin{pmatrix}
  a & * \\
  * & 1
\end{pmatrix}, 
\begin{pmatrix}
  a & * \\
  * & 1
\end{pmatrix}
$$

105
In practice, it is sufficient to use \( \begin{bmatrix} 1 & -1 \frac{c}{\sqrt{2}} \\ -1 & 1 \end{bmatrix} \) in case \( |c| \leq \frac{1}{\sqrt{2}} \) and 
\( \begin{bmatrix} 1 & 1 \frac{c}{\sqrt{2}} \\ 1 & -1 \end{bmatrix} \) in case \( |c| > \frac{1}{\sqrt{2}} \).

2. There is clearly a danger of underflow in the elements of the diagonal matrix when a sequence of plane rotations is involved. This can be avoided by storing the exponent of the diagonal elements separately or by normalizing occasionally (Hammerling [2]).

3. With this technique it is necessary to store four parameters of each plane rotation: one parameter to indicate whether \( \begin{bmatrix} 1 & -1 \frac{c}{\sqrt{2}} \\ -1 & 1 \end{bmatrix} \) or 
\( \begin{bmatrix} 1 & 1 \frac{c}{\sqrt{2}} \\ 1 & -1 \end{bmatrix} \) is used and, if for instance \( \begin{bmatrix} 1 & -1 \frac{c}{\sqrt{2}} \\ -1 & 1 \end{bmatrix} \) is used, the values of \( c, \frac{d_1}{d_2} \) and \( \frac{d_2}{d_1} \).

The determination of \( R(k+1) \) with the decomposition.

In subsection 4.1.1 we saw that application of the algorithm to a nontrivial impulse response \( \{c_k\}_{k=1}^\infty \) supplies the order of a minimal realization, provided that \( \epsilon \) is small enough and \( R(k+1) \), with \( \epsilon = \epsilon \), \( \left\| R_{k+1} \right\|_2 \), can be determined from the decomposition of \( R_{k+1} \):

\[ M_{k+1,k+1} R_{k+1,k+1} P = R_{k+1,k+1} \]

Omitting the subscripts, we know that \( R \) can be partitioned as

\[ R = \begin{pmatrix} R_{11} & R_{12} \\ 0^T & R_{22} \end{pmatrix} \]

with \( R_{11} \), a regular \( k \times k \) matrix of upper triangular form. \( R_{22} = 0^T \) if and only if \( \tau(R(k+1)) = k \). Partitioning \( M \) correspondingly, we found in chapter 2 that

\[ \| b \| \leq \sigma_{k+1}(M) \| b \| \]

where

\[ \| b \| = \frac{\| b \|_{2,2}}{\| (M_{21}M_{22}) R_{21} \left( R_{11}^{-1} R_{12} \right) \|_{2,2} + \| (M_{11}M_{12}) R_{21} \|_{2,2} + \| R_{11}^{-1} R_{12} \|_{2,2}} \]

106
and 

\[ \mathbf{ub} = \mathbf{E}_2 \mathbf{b}_2 \| \mathbf{M}^{-1} \|_2. \]

From theorem 2.10, it follows that \( r_c(\mathbf{H}_{k+1,k}) \) can be determined with the lower and upper bound for \( r_c(\mathbf{H}_{k+1,k}) \) unless

\[ 1 \mathbf{b} < 2 \mathbf{c}(= 2 \text{tol} \mathbf{E}_{k+1,k} \|_{L_2}) \leq \mathbf{ub}. \]

Because either \( \mathbf{ub} = \mathbf{ub} = 0 \text{ or } 0 \leq \mathbf{ub} \leq \mathbf{ub} \), \( (11) \) cannot hold if \( \text{tol} \) is small enough. The number of possible pairs \( k \) and \( \ell \) in the computational process is finite; so \( \text{tol} \) can be chosen so small that \( r_c(\mathbf{H}_{k+1,k}) \) can be determined with the decomposition of \( \mathbf{H}_{k+1,k} \) for any possible pair \( k \) and \( \ell \).

In order to evaluate \( r_c(\mathbf{H}_{k+1,k}) \), \( \mathbf{ub} \) as well as \( \mathbf{ub} \) should be computed. We see that with the first decomposition method more operations are necessary to determine \( \mathbf{ub} \) and \( \mathbf{ub} \) than with recursive Givens decomposition, because in the latter case \( \mathbf{M} \) is orthogonal. Even if \( \mathbf{ub} \) is replaced by

\[ \mathbf{ub}' = \frac{\| \mathbf{E}_{22} \mathbf{b}_2 \|_2}{\| \mathbf{E}_{11}^{-1} \mathbf{x}_2 + \mathbf{E}_{12}^{-1} \mathbf{x}_2 \|_2}, \]

the number of operations required to evaluate \( \| \mathbf{M} \|_2 \) and \( \| \mathbf{M}^{-1} \|_2 \) - or the order of magnitude of these numbers - makes the first decomposition comparable with the second decomposition method as far as the amount of work is concerned. Because a decomposition with orthogonal \( \mathbf{M} \) is superior (see subsection 2.3.1) when the \( \varepsilon \)-stable rank is determined, we shall from now on assume that the algorithm employs recursive Givens decomposition.

#### 4.1.3

At the end of the computational process we have

\[ \mathbf{M}_{k+1,k+1} \mathbf{H}_{k+1,k} \mathbf{P} = \mathbf{H}_{k+1,k}. \]

with \( k = n, k + 1 = 2n \) and the last row of \( \mathbf{H}_{k+1,k} \) equal to zero. In chapter 3 we already saw that a minimal realization triple for \( \{ a_k \}_{k=1}^n \) is given by
with \( (x_i, \ldots, x_{k+1}) = e_k^{T} \sigma_{k+1}^{-1} \) where \( \sigma = (\sigma_{k+1}^{-1} \sigma_{k+1}^{T})^{-1} \).

We shall give a number of minimal realization triples that can be constructed with (12), where the realization matrix does not have companion form. We assume that \( k > k \). Omitting the subscripts and defining \( \Lambda := \Lambda_{s} \), \( \Lambda := \Lambda_{u} \), we obtain from (12)

\[
H = LU,
\]

The last row of \( U \) is zero.

**Notation.**

\[
\begin{align*}
U_1 &= \begin{bmatrix} s_1 & \cdots & s_{k-1} \\ \vdots & \ddots & \vdots \\ s_k & \cdots & s_{k+1} \end{bmatrix} & U_2 &= \begin{bmatrix} s_2 & \cdots & s_{k} \\ \vdots & \ddots & \vdots \\ s_{k+1} & \cdots & s_{2k-1} \end{bmatrix} \\
L_1 &= \begin{bmatrix} l_{1,1} & \cdots & l_{1,k} \\ \vdots & \ddots & \vdots \\ l_{k,1} & \cdots & l_{k,k} \end{bmatrix} & L_2 &= \begin{bmatrix} l_{2,1} & \cdots & l_{2,k} \\ \vdots & \ddots & \vdots \\ l_{k+1,1} & \cdots & l_{k+1,k} \end{bmatrix} \\
U_1 &= \begin{bmatrix} u_{1,1} & \cdots & u_{1,k-1} \\ \vdots & \ddots & \vdots \\ u_{k,1} & \cdots & u_{k,k-1} \end{bmatrix} & U_2 &= \begin{bmatrix} u_{1,2} & \cdots & u_{1,k} \\ \vdots & \ddots & \vdots \\ u_{k,2} & \cdots & u_{k,k} \end{bmatrix}
\end{align*}
\]

\( U_1^{T} := (u_{i,1}, \ldots, u_{i,k-1}) \) \( 1 \leq i \leq k-1 \)

\( L_{1}^{T} := (l_{i,1}, \ldots, l_{i,k}) \) \( 1 \leq i \leq k-1 \) (the rows of \( L_1 \) and \( L_2 \))

\( U_2^{T} := (u_{i,1}, \ldots, u_{i,k}) \) \( 1 \leq i \leq k \) (the columns of \( U_1 \) and \( U_2 \)).
Theorem 5.

$L_1$ is regular, $U_1$ has full row rank.

Proof.

We have, as the last row of $U$ is zero,

\[ H_2 = L_1 U_1. \]

Since $r(H_2) = k$, the statement follows. \( \square \)

Theorem 6.

\[ H_2 = L_2 U_1 = L_1 U_2. \]

Proof.

Employing the fact that $H_2$ appears in two different ways as a submatrix of $H$ and that the last row of $U$ is zero, the result follows. \( \square \)

From theorems 5 and 6, we obtain, if $U_1^{(r)}$ is a right-inverse of $U_1$,

\[ L_2 = L_1 U_2^{(r)} \]
\[ U_2 = L_1^{-1} L_2 U_1^{(r)} \]

and

\[ L_1^{-1} L_2 = U_2 U_1^{(r)} \Rightarrow A. \]

From these three relations it follows that

\[ \kappa_1^T i_1 = \delta_1^T A \quad (1 \leq i \leq k) \]
\[ u_1^{(r)} = A u_1 \quad (1 \leq i \leq k-1). \]

If $1 \leq i \leq k$, we obtain from (14)

\[ s_i = \kappa_1^T u_1 = \kappa_1^T A^{-1} u_1. \]

If $k < i \leq k+1$, we have

\[ s_i = \kappa_1^T u_1 = \kappa_1^T A^{-1} u_1 = \kappa_1^T A^{-1} u_1. \]

Consequently, the following theorem applies:
Theorem 7.
A minimal realization triple for \( \{a_i\}_{i=1}^\infty \) is given by

\[
(15) \quad (A, u_1, 1)_k,
\]

where \( A = L_1 L_2 + U_1 \rho \).

Corollary.
Other realizations for \( \{a_i\}_{i=1}^\infty \) are given by

\[
(16) \quad (H_2 H_1(x), h_1, e_1)_k
\]

and

\[
(17) \quad (H_1 H_2(x), H_1 h_1, h_1)_{k-1},
\]

where \( H_1(x) \) is a right-inverse of \( H_1 \).

The realizations (16) and (17) are readily obtained from (15). The triple (16) is equal to the triple (13); however, it is derived in another way. We have, essentially, three minimal realizations for \( \{a_i\}_{i=1}^\infty \):

\[
(15) \quad (A, u_1, 1)_{k}
\]

\[
(16) \quad (H_2 H_1(x), h_1, e_1)_{k}
\]

\[
(17) \quad (H_1 H_2(x), H_1 h_1, h_1)_{k-1},
\]

If the decomposition (14) holds approximately, these three will all supply different impulse responses \( \{a_i\}_{i=1}^\infty \). In that case we expect that the triple

\[
(16) \quad (U, U(x), x)_{k}
\]

will supply the best impulse response because, obviously, this triple contains the most information about the matrix \( H \).

110
Remarks.
1. In theorem 7 we have

$$A = L_1^{-1} L_2 = U_2 U_1^F$$

Obviously, we may take for $v_1^F$ any right-inverse of $U_1$. Henceforth, we shall use the pseudo inverse of $U_1$.

2. Because the matrix $K$ is stored in product form, the vector

$$x_1^T = x_{n-1}^T (I - e_{k+1} e_{k+1}^T)$$

is not directly available. By applying the full sequence of plane rotations to $x_1$, the vector $x_1$ is obtained with approximately $k^2$ operations.

4.2. The numerical stability of the algorithm

In 4.1 being concerned with: if the computations are performed without errors, is then the supplied output correct, we are in this section concerned with the question: if the computations are performed with small errors, is then the supplied output "satisfactory".

In subsection 4.2.1 we discuss the numerical stability of recursive Givens decomposition, defined in 4.1.2. In subsection 4.2.2 we study the determination of $T_{k+1}^{-1} H_k$ with the decomposition, and, finally, in the subsections 4.2.3 and 4.2.4 we investigate the numerical stability of the algorithm.

4.2.1.

Recursive Givens decomposition starts out with the decomposition of a small Hankel block and updates the decomposition whenever the Hankel block is enlarged with a row or a column. For any $k$ and $p$ in the computational process a decomposition is obtained of the form

$$H_{k+1}^{p+1} B_{k+1}^{p+1} F = R_{k+1}^{p+1}$$

Let us suppose that the starting point is $H_{k+1}$. Then every column of $H_{k+1}^{p+1}$ originates by premultiplying a column of $H_{k+1}^{p+1}$ (by $a_j$ say) by a sequence of plane rotations:

$$X_{k+1}^{p+1} = (T(k,k+1), T(1,k+1), \ldots, T(1,k+1), \ldots, T(1,2))$$

where $T(p,q)$ for $1 \leq p < q \leq k+1$ is a plane rotation in the $p$-$q$ plane.
Let all computations be performed with a relative precision of \( \eta \), where \( \eta \) is so small that no break down in the computational process occurs. We investigate the multiplication of a vector \( v_1 \) by a sequence of plane rotations \( T_1 \ldots , T_s \). We may compute approximate plane rotations \( \tilde{T}_1 \), \( 1 \leq i \leq s \), and thence compute the vectors

\[
v_{i+1}^{*} = \tilde{T}_i (v_i^{*}), \quad 1 \leq i \leq s.
\]

We define

\[
\tilde{v}_1 = v_{1+1}^{*} = \tilde{T}_1 v_1.
\]

The methods for computing \( T_1 v_1 \) that we discussed in subsection 4.1.2 have the property that for some positive number \( \kappa \)

\[
\| \tilde{v}_1 \|_2 \leq \kappa \| v_1 \|_2.
\]

For instance, if the computation of \( v_{1+1}^{*} \) is so organized that four multiplications are involved, then one may prove that \( \kappa = 3 + 2\sqrt{2} \) (Wilkinson [5]).

Consequently,

\[
\| v_{s+1}^{*} \|_2 = \| T_s \ldots T_2 \tilde{v}_1 \|_2 \leq \| T_s \ldots T_2 \tilde{v}_2 \|_2 + \| T_s \ldots T_2 \tilde{v}_2 \|_2 + \cdots + \| T_1 \tilde{v}_1 \|_2 \leq \kappa (1 + \kappa) + \cdots + (1 + \kappa)^{s-1}.
\]

Hence, if \( \eta \) is small enough,

\[
\| v_{s+1}^{*} \|_2 \leq \| \tilde{T}_s \ldots \tilde{T}_1 v_1 \|_2 \leq 1.1 \kappa \| v_1 \|_2.
\]

Let \( \tilde{R}_{k+1,k} \) be the matrix obtained with inexact computations and let \( R_{k+1,k} \) be the exact product of the exact plane rotations occurring in the computational process (so \( R_{k+1,k} \) is orthogonal).

From (18) it follows that

\[
R_{k+1,k} = \tilde{R}_{k+1,k} + \delta R_{k+1,k}
\]

where \( \delta R_{k+1,k} \) has upper trapezoidal form and, if \( \kappa \) is small enough,

\[
\| (\delta \tilde{R}_{k+1,k})_2 \|_2 \leq 1.1 \| k (1 + k) \kappa \| \tilde{R}_{k+1,k} \|_2.
\]
Hence,
\[ \| \Delta \hat{R}_{k+1,k} \|_2 \leq 1, \| \Delta \|_2 (1 + \| k^{(k)} \|_2) \leq 1, \| k^{(k)} \|_2 \leq 1, \| H_{k+1,k} \|_2 \leq 1, \| H_{k+1,k} \|_2 \leq 1. \]

From (19), it follows that
\[ H_{k+1,k+1} \begin{pmatrix} H_{k+1,k} & \Delta H_{k+1,k} \\ \Delta H_{k+1,k} & H_{k+1,k} \end{pmatrix} \begin{pmatrix} P \Delta H_{k+1,k} \\ \Delta H_{k+1,k} \end{pmatrix} = \hat{R}_{k+1,k} \]

with
\[ \| \Delta H_{k+1,k} \|_2 \leq \| H_{k+1,k} \|_2 \| \Delta H_{k+1,k} \|_2 \leq 1, \| (1 + \| k^{(k)} \|_2) \|_2 \leq 1, \| H_{k+1,k} \|_2 \leq 1, \| H_{k+1,k} \|_2 \leq 1. \]

From Chapter 1, it follows that recursive Givens decomposition is backward stable; \( \hat{R}_{k+1,k} \) is obtained by exactly applying a sequence of plane rotations to a matrix that is a numerical neighbour of \( H_{k+1,k} \).

Remarks:
1. The approximate plane rotations \( \hat{R}(p,q) \), where \( 1 \leq p < q \leq k+1 \), are stored, not the exact plane rotations, \( R \), so we cannot determine the exact \( H_{k+1,k+1} \).
2. The matrix \( \Delta H_{k+1,k} \) is a full matrix; \( \Delta H_{k+1,k} \) is not a Hankel matrix.
3. Employing that the plane rotations affect only two components of the vectors to which they are applied and employing that there is a regularity in the sequence of rotation matrices, it is possible to give a sharper bound than (21); using a similar technique as Wilkinson in [5, pp. 134-137] one may prove that the bound can be sharpened by a factor \( \sqrt{2} \).

A posteriori error analysis of the decomposition methods.

In the computational process \( r(\hat{H}_{k+1,k}) \) is determined from (20).
In order that this is possible at all, it is necessary that \( \| \Delta \|_2 \leq \| \Delta \|_2 \) and, consequently, that \( \| \Delta \|_2 \) is an upper bound for \( \| \Delta R_{k+1,k} \|_2 \) is known. An a priori bound is generally too pessimistic. An a posteriori estimate for \( \| \Delta \|_2 \) may be obtained by applying the same sequence of Givens matrices to \( H_{k+1,k} \) but now in double length (the parameters determining the Givens matrices have to be computed in double length as well).
4.2.2.
In subsection 4.1.1 we saw that, if

\[ 0 < \text{tol} \max \{ C_i, C_j \} < \epsilon, \]

the finally supplied value of \( k \) is equal to \( n \) provided that \( r(x_{k+1}, x) \) with \( \epsilon = \text{tol} \max \{ H_{k+1}, H \} \) can be determined from the decomposition of \( H_{k+1} \) for any pair \( (x_i, x_j) \) occurring in the computational process. In subsection 4.1.2 we showed that, \( \text{tol} \) being small enough and the computations being exact, \( r(x_{k+1}, x) \) can be determined from the decomposition of \( H_{k+1}^{(l)} \). In this subsection we show that, provided that \( n \) is small enough, a value of \( \epsilon \) to be found such that (22) holds and \( r(x_{k+1}, x) \) can be determined from the approximate decomposition:

\[ H_{k+1} \approx H_{k+1} \quad \text{and} \quad H_{k+1}^{(l)} = \tilde{H}_{k+1} + \Delta H_{k+1} \]

Furthermore, we shall discuss how the particular value of \( \epsilon \) may be chosen (a subject that has been avoided so far).

Let \( (k, l) \) be a pair such that \( 1 \leq k \leq l \leq 2n \) and that, if \( \text{tol} \) satisfies (22), \( r(x_{k+1}, x) = r(x_{k+1}, x) = k \) holds with \( \epsilon = \text{tol} \max \{ H_{k+1}, H \} \). Let \( H_{k+1} \) have the decomposition (19) and let

\[ \delta_{k+1, \epsilon} := \| \tilde{H}_{k+1} - H_{k+1} \|_2 \]

From chapter 2 it follows that

\[ \text{tb} \leq \sigma_{k+1} (H_{k+1}) \leq \text{ub} \]

where (employing the same partitioning as in subsection 4.1.2)

\[ \text{tb} = \frac{1}{\| H_{22} \|_2} \quad \text{and} \quad \text{ub} = \| \tilde{H}_{22} \|_2 \]

Hence,

\[ \max(0, \delta_{k+1}) \leq \sigma_{k+1} (H_{k+1}) = \| \tilde{H}_{k+1} + \Delta H_{k+1} \|_2 \]

This result is based on two well known theorems:
(a) If a symmetric matrix \( A \) is perturbed by a matrix \( \Delta \), then all eigenvalues of \( A \) are at most perturbed by \( \| \Delta \|_2 \).

(b) The matrix
\[
\begin{bmatrix}
0 & \Delta \\
\Delta^T & 0
\end{bmatrix}
\]
has eigenvalues \( \pm c_1(\Delta), \ldots, \pm c_k(\Delta) \).

Consequently, if \( \hat{E}_{k+1,t} \) is perturbed by \( \Delta \hat{E}_{k+1,t} \), the singular values of \( \hat{E}_{k+1,t} \) are at most perturbed by \( \| \Delta \hat{E}_{k+1,t} \|_2 \).

From theorem 2 it follows that, if \( tol \) satisfies (22),
\[ r_c(\hat{E}_{k+1,t}) = r(\hat{E}_{k+1,t}) \]
holds with \( c = tol \cdot \| \hat{E}_{k+1,t} \|_2 \). In order that \( r_c(\hat{E}_{k+1,t}) \) can be determined from (24) we should also have (theorem 2.10)
\[ 2c \leq \max(0, \text{tb} - \hat{E}_{k+1,t}^2) \text{ in case } r(\hat{E}_{k+1,t}) = k+1 \]
or
\[ 2c > \text{ub} + \hat{E}_{k+1,t} \text{ in case } r(\hat{E}_{k+1,t}) = k. \]

Consequently, in addition to (22), \( tol \) should satisfy
\[ \text{tol} \leq \frac{\max(0, \text{tb} - \hat{E}_{k+1,t}^2)}{\| \hat{E}_{k+1,t} \|_2} \text{ in case } r(\hat{E}_{k+1,t}) = k+1. \]
or
\[ \text{tol} > \frac{(\text{ub} + \hat{E}_{k+1,t})}{\| \hat{E}_{k+1,t} \|_2} \text{ in case } r(\hat{E}_{k+1,t}) = k. \]

The bounds (25a) and (25b) depend on \( n \). Since we are dealing with a rational process of which the outcome and the interim results depend continuously on the data, the decomposition (19) has, if \( n \) is small enough, the same properties as the decomposition of \( \hat{E}_{k+1,t} \) that would have been obtained with exact computations. Therefore, if \( n \) is small enough, then the bound (25a) is larger than some positive constant or the bound (25b) may be less than any positive number. Hence, then \( tol \) exists such that (22) and (25) are satisfied and \( r_c(\hat{E}_{k+1,t}) \) can be determined from (24).

The number of possible pairs \((k,t)\) is finite. Consequently, if \( n \) is small enough, then \( tol \) exists such that (22) and (25) hold for all eligible \((k,t)\) and \( r_c(\hat{E}_{k+1,t}) \) can be determined with the decomposition.
Summarizing, we have

\[ (26) \quad \forall \epsilon \in \mathbb{R}_+ \exists \eta_0 \in \mathbb{R}_+ \exists \tau_0 \forall \eta < \eta_0 \forall \epsilon \in \mathbb{R}_+ \]

Here, as in chapter 3, \( D_n \) denotes the set of all impulse responses admitting a minimal realization of order \( n \). \( R_n \) denotes the set of all minimal realization triples of order \( n \) (which is slightly different from the definition of \( R_n \) in chapter 3). \( f(t)(d) \) denotes a minimal realization triple that may be supplied by the algorithm.

Next, let us discuss the particular choice of \( \tau_0 \). First of all, we note that, if (22) and (23) hold for a \( \tau_0 \) that depends on \( k \) and \( t \), it still may be shown that the finally supplied value of \( k \) equals \( n \). Let us suppose that, indeed, \( \tau_0 \) depends on \( k \) and \( t \), and that

\[ \tau_0^k = 0, \quad \delta_{k+1} \leq \epsilon \leq \delta_{k+1} \epsilon (1 + \| R_{k+1}^{-1} R_{12} \|_2^2) \| R_{k+1} \|_2 \]

Because \( \delta_{k+1} \) approaches zero if \( n \) approaches zero, (22) holds if \( n \) is small enough. In case \( r(\delta_{k+1} \epsilon) = k+1 \), (23a) holds for the same reason if \( n \) is small enough (note that the right-hand side of (23a) is a lower bound for \( C_0^{-1} \)).

Let us consider (23b). Since \( r(\delta_{k+1} \epsilon) = k \), we have, if \( n \) is small enough, that in (19) the last row of \( R_{k+1} \epsilon \) is linearly dependent on the first \( k \) rows of \( R_{k+1} \epsilon \). Omitting the subscripts and partitioning as before, it follows that

\[ -\delta_{21} \begin{pmatrix} R_{11} + \Delta R_{11} \\ R_{12} + \Delta R_{12} \\ R_{22} + \Delta R_{22} \end{pmatrix} = 0 \\ \text{and, hence,} \\ R_{22} = -\begin{pmatrix} \Delta R_{21} \\ \Delta R_{22} \end{pmatrix} \left( \begin{pmatrix} R_{11} + \Delta R_{11} \\ R_{12} + \Delta R_{12} \end{pmatrix} \right)^{-1} \begin{pmatrix} R_{11} + \Delta R_{11} \\ R_{12} + \Delta R_{12} \end{pmatrix} \end{pmatrix} \]

Since \( \| R_{22} \|_2 \), we obtain

\[ \| R_{22} \|_2 \leq \| R_{22} \|_2 \| \begin{pmatrix} R_{11} + \Delta R_{11} \\ R_{12} + \Delta R_{12} \end{pmatrix} \|_2^{-1} \]
whence, if \( \eta \) is small enough,

\[
(27) \quad \frac{a b}{1 + |a|} \frac{1}{\frac{1}{\| \mathbf{R}_{11} \|_2} \mathbf{R}_{12}} < \frac{\| k+1 \|_2}{\| k+1 \|_2} \leq \frac{1}{\| \mathbf{R}_{11} \|_2} \mathbf{R}_{12}.
\]

Substituting (27) in the right-hand side of (25b) gives

\[
\frac{(a b + k+1 \frac{1}{\| k+1 \|_2})}{\| k+1 \|_2} < \frac{1}{\| k+1 \|_2} \frac{1}{\| \mathbf{R}_{11} \|_2} \mathbf{R}_{12} < \frac{1}{\| k+1 \|_2} \frac{1}{\| \mathbf{R}_{11} \|_2} \mathbf{R}_{12}.
\]

So, (25b) is satisfied if \( \eta \) is small enough.

Consequently, if in lines (5) and (8) of the algorithm

\[
\epsilon := \text{tol} \| \mathbf{R}_{k+1} \|_2
\]

is replaced by

\[
(28) \quad \epsilon := 6 \frac{k+1}{\| k+1 \|_2} (1 + \| \mathbf{R}_{11} \|_2 \mathbf{R}_{12}),
\]

then, provided that \( \eta \) is small enough, \( k_{c}(\mathbf{R}_{k+1}) \) can be determined from (24) and the finally supplied value of \( k \) is equal to \( n \).

The algorithm can only detect to a certain extent whether \( \eta \) is small enough: in case \( k_{c}(\mathbf{R}_{k+1}) \) cannot be determined with the decomposition. However, if \( k_{c}(\mathbf{R}_{k+1}) \) can be determined, this is a strong indication that \( \eta \) is small enough.

Remark.

If for some particular input impulse response it is a priori known which values for \( \eta \) are small enough, then it is not necessary to verify (25a) as well as (25b). The algorithm may verify either (25a) or (25b).

For instance, if one finds

\[
\epsilon := 6 \frac{k+1}{\| k+1 \|_2} (1 + \| \mathbf{R}_{11} \|_2 \mathbf{R}_{12}) > 1 \frac{1}{\| \mathbf{R}_{11} \|_2} \mathbf{R}_{12} = \frac{1}{\| \mathbf{R}_{11} \|_2} \mathbf{R}_{12}.
\]

then \( k_{c}(\mathbf{R}_{k+1}) = k \), otherwise \( k_{c}(\mathbf{R}_{k+1}) = k+1 \).
In this subsection we investigate the sensitivity of a realization triple to small changes in the impulse response and, conversely, the sensitivity of the impulse response to small changes in a realization triple. It will appear that in both cases the condition number of \( H_{2,2} \) is involved.

Let us consider the impulse response

\[
\{ \bar{s}_1 \}_{i=1}^\infty = \{ s, 1, \frac{3}{2}, 2, \ldots \} \text{ where } 0 < s < 1,
\]

that admits a minimal realization triple of order 2 \( (\bar{H}_{2,2}) = r(\bar{H}_{3,2}) = 2 \).

The condition number of \( H_{2,2} \) is:

\[
C_2(H_{2,2}) = \frac{2(a-1)^2 - 4(a-1) + 2(a+1) \sqrt{(a+1)^2 - 4(a-1)}}{4|s-1|}.
\]

If \(|s-1| < 1\), we obtain in first order approximation

\[
C_2(H_{2,2}) = \frac{6}{|s-1|}.
\]

Employing (16), we find the minimal realization triple for \( \{ s \}_{i=1}^\infty \):

\[
(\begin{bmatrix}
0 & a - 1 \\
-1 & 2a - 1
\end{bmatrix}, \begin{bmatrix}
1 \\
0
\end{bmatrix}, \begin{bmatrix}
s \\
1
\end{bmatrix}) \leftarrow (A_2, b_2, c_2).
\]

Let us suppose that the realization matrix \( A \) is perturbed with \( \Delta A \) such that \(|\Delta A| < \eta |A|\).

For instance,

\[
\Delta A = \begin{bmatrix}
0 & (s-1)(1+s) \\
-1 & (2s-1)(1+s)
\end{bmatrix}.
\]

The impulse response that is generated by the triple \( (A_i, b_i, c_i) \) is:

\[
\{ \bar{s}_1 \}_{i=1}^\infty
\]

with

\[
|\Delta s| < \eta (i = 1, 2).
\]
\[ s_1 = a, \]
\[ s_2 = 1 + s_1, \]
\[ s_3 = 1 + s_1 + (1 + s_1)g_2\frac{(2\alpha - 1)}{\alpha - 1}, \]
\[ s_4 = 2 + 20\frac{1}{1 + s_1} + 82\frac{(2\alpha - 1)}{\alpha - 1} + 82(1 + s_2)\frac{(2\alpha - 1)}{\alpha - 1}. \]

We see that \( \left( \frac{4}{\pi} \sum_{i=1}^{n} (s_i - \bar{s}_i)^{-1} \right) \) is of the order of \( n(\alpha - 1)^{-2} \text{ or } nC_2^2(H_{2,2})^2 \). Hence, no realization algorithm is backward stable on the input set \( \mathcal{P}_n \), because the backward error may be of the order of \( C_2^2(H_{2,2})^2 \text{ or } C_2^2(H_{2,2})^2 \) cannot be bounded on \( \mathcal{P}_n \).

Next, let us investigate the sensitivity of a realization triple to small changes in the impulse response.

We consider the same impulse response

\[ (s_i)_{i=1}^{\infty} = \{a, 1, 2, 3, \ldots \} \] where \( 0 < a < 1 \).

Let us suppose that \( (s_i)_{i=1}^{\infty} \) is perturbed:

\[ (\bar{s}_i)_{i=1}^{\infty} = \{a + \delta, 1 + \delta, 2, \ldots \} \]

with \( |\delta| \leq \eta \) (\( i = 1, 2 \)).

By applying (16) one may verify that \( (s_i)_{i=1}^{\infty} \) has a minimal realization triple with realization matrix

\[
\bar{A} = \begin{bmatrix}
0 & 1 \\
(1 + \bar{s}_2)^2 - 2(1 + \bar{s}_2) & 2\bar{s} - (1 + \bar{s}_2)(1 + \bar{s}_2) \\
\bar{a}(1 + \bar{s}_2)^2 - (1 + \bar{s}_2)^2 & \bar{a}(1 + \bar{s}_2) - (1 + \bar{s}_2)^2
\end{bmatrix}
\]

Let

\[ \Delta A = \bar{A} - A. \]
If $n$ is so small that higher order terms may be neglected, we find
\[
\begin{align*}
\Delta A_{21} &= \frac{-1 + 2a_2 - 2a_2}{a_2(a - 1)^2} - \frac{1}{a - 1} - \frac{1}{a - 1} (2a_2 - 2a_2^2 + \frac{2a_2 - a_2}{a - 1}) .
\end{align*}
\]

Hence, $\frac{\|\Delta A\|}{\|A\|}$ is of the order of $\frac{C_2}{n^2}$. 

Consequently, any realization algorithm is not forward stable on the input set $\mathcal{P}_n$, because the forward error may be of the order of $C_2(n, n)$ and $C_2(n, n)$ cannot be bounded on $\mathcal{P}_n$.

The conclusion of this subsection is that any realization algorithm is numerically unstable on $\mathcal{P}_n$, since the condition number of $H_{n, n}$ cannot uniformly be bounded on $\mathcal{P}_n$. In the next subsection we shall investigate on which subset of $\mathcal{P}_n$ the algorithm proposed in this chapter is numerically stable.

Remark.

The algorithms discussed in chapter 3 are numerically unstable on subsets of $\mathcal{P}_n$ for which $C_2(n, n)$ is small; this is obvious from the example with which the numerical instability was proved.

4.2.4. 

Let $N$ be an a priori known natural number such that $N > n$. We saw in subsection 4.2.2 that application of the proposed algorithm to some $d \in \mathcal{P}_n$ results in a value of $k$ that is equal to $n$ provided that all computations are performed with a relative precision of $\eta_0(d)$.

In subsection 4.2.3 however, we saw that the algorithm can only be numerically stable on a subset of $\mathcal{P}_n$.

We consider the following subset of $\mathcal{P}_n$ as input set:

\[
\mathcal{P}_{n}^{(B_1, B_2)} := \{ d \in \mathcal{P}_n \mid C_2(n, n) < B_1; \frac{\|H^{-1}_{n, n}\|}{\|A_{21}\|} \leq B_2 \}
\]

where $B_i > 0$ ($i = 1, 2$). The output set is

\[
\mathcal{R}_n := \{ (A, b, c) \in \mathcal{P}_n \mid (A, b, c) \text{ is a realization triple for some } d \in \mathcal{P}_n \}.
\]
In $\mathbb{R}_n$ we define the semi-metric

$$
u'(\rho, \beta, \gamma)^n, (\rho, \beta, \gamma)^n) := \inf_{T \in \mathcal{B}_n} \left( \| A - TA^{-1} B \|_{2,2} + \| B - TB^{-1} C \|_{2,2} + \| C - TC^{-1} D \|_{2,2} \right).$$

Performed without round-off errors, the algorithm corresponds to the mapping

$$f : \mathcal{P}_n(B_1, B_2) \to \mathbb{R}_n.$$ 

Performed with round-off errors, the algorithm corresponds to the mapping

$$f^\varepsilon : \mathcal{P}_n(B_1, B_2) \to \mathbb{R}_n$$

(we assume that if the algorithm is applied to $d \in \mathcal{P}_n(B_1, B_2)$, indeed, $n$ is so small that $f^\varepsilon(d) \in \mathbb{R}_n$.)

We shall show that the algorithm is forward stable on $\mathcal{P}_n(B_1, B_2)$ for all positive numbers $B_1$ and $B_2$:

$$\exists \varepsilon > 0 \forall \| \delta \|_n, \| \epsilon \|_n \in \mathbb{R}_n, \| \eta \|_n \in \mathbb{R}_n [\nu'([f^\varepsilon(d) + \delta], [f(d) + \epsilon]) < C \nu'([f(d) + \delta], [f(d) + \epsilon])].$$

**Remark.**

It is clear from 4.2.3 why a subset of $\mathcal{P}_n$ is considered for which $C_2(H_{n,n})$ is bounded. It is less clear why $\| x \|_n$, where $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, should be bounded. However, then it is possible to bound $C_2(H_{n,n})$, which also influences the forward error ($H_{n+1,n}$ is the final Hankel block considered in the computational process). The vector $(s_n - 1)$ contains the coefficients of the characteristic polynomial of $(s_1 s_2 \cdots s_n)^{n+1}$ (theorem 0.3).

Instead of $\mathcal{P}_n(B_1, B_2)$ we may also consider a subset of $\mathcal{P}_n$ for which $C_2(H_{n,n})$ is bounded (by $B_1$ say) and for which $|s_{\infty}|$ is bounded by $\| H_{n,n} \|_2$, where $\| \cdot \|_2$ is some positive number. For then we have

$$\| x \|_n \leq \| s_1 \|_n \leq \| s_2 \|_n \leq \| s_3 \|_n \leq \cdots \leq \| s_{\infty} \|_n.$$ 

On the other hand, if $C_2(H_{n,n})$ is bounded by $B_1$ and $\| x \|_n$ by $B_2$ then

$$|s_{\infty}| \leq \| H_{n,n} x \|_n \leq B_2 \| H_{n,n} \|_2.$$ 

$\square$
Let the algorithm, with \( n < n_0 \), be applied to \( d \in D_n(\mathcal{F}_1, \mathcal{F}_2) \).

At the end of the computational process we have for the final Hankel block

\[
H_{k+1, l} = k + l = 2k \quad \text{and} \quad k = n,
\]

(30) \[ N_{k+1, l} \in H_{k+1, l} \quad \text{if} \quad F = E_{k+1, l} \quad \text{and} \quad \Delta E_{k+1, l} \quad \text{is}, \]

(31) \[ \phi_{k+1}^T(\tilde{E}_{k+1, l} + \Delta \tilde{E}_{k+1, l}) \in \text{Span}(E_{k+1, l} \quad \text{and} \quad \Delta \tilde{E}_{k+1, l} \quad \text{for} \quad 1 \leq i \leq k). \]

The matrix \( N_{k+1, l} \) is orthogonal, \( F \) is a permutation matrix and \( \tilde{E}_{k+1, l} \) has upper trapezoidal form. For \( \frac{\|\Delta E_{k+1, l}\|_2}{\|E_{k+1, l}\|_2} \), an upper bound is known that is independent of \( d \) and \( n \). Formula (31) holds because \( r(E_{k+1, l}) = r(\tilde{E}_{k+1, l}) = k \) and \( n \) is small enough (see 4.1.1 and 4.2.1).

Omitting the subscripts, it follows from (30) that

\[
H = N^T(\tilde{R} + \Delta \tilde{R})F^T.
\]

Let \( \tilde{R} \) be partitioned as \( \begin{bmatrix} \tilde{R}_{11} & \tilde{R}_{12} \\ 0 & \tilde{R}_{22} \end{bmatrix} \), with \( \tilde{R}_{11} \) an \((n \times n)\) regular upper triangular matrix. If \( n \) is small enough, we may define the \((n+1) \times (n+1)\) regular matrix

\[
T = \begin{bmatrix}
I & 0 \\
\tilde{R}_{21} & \tilde{R}_{11}^{-1} & 1
\end{bmatrix}.
\]

It follows that

\[
H = (M^T)^{-1}(T(\tilde{R} + \Delta \tilde{R})F^T)
\]

is a decomposition for \( H \) of the form \( H = LU \), where

\[
L := (M^T)^{-1} = (TH)^{-1}, U := T(\tilde{R} + \Delta \tilde{R})F^T
\]

and the last row of \( U \) is equal to zero (applying 31).

Let

\[
L + AL = H^{-1} = M^T \quad \text{and} \quad U + AU = T F^T.
\]

122
Employing notation 4 of subsection 4.1.3 we obtain that

\[(32) \quad \mathbf{f}(d) = (U_b^T, u_1^h, u_1^h)_{\mathbb{R}}^* \]

whereas the algorithm may supply

\[(33) \quad \mathbf{f}(\mathcal{I})(d) = (U_b^T + \Delta U_b)(U_1^h + \Delta U_1)^* u_1^h + \Delta u_1^h, t_1^h + \Delta t_1^h)_{\mathbb{R}} \]

Because \( T \) does not affect the first \( n \) rows of \((\mathbb{R} + \Delta \mathbb{R})\), \( \mathbf{f}(\mathcal{I})(d) \) can be directly obtained from \( \mathbf{U}^T \) and \( \mathbf{H}^T \).

Before showing that the algorithm is forward stable on \( \mathcal{P}_{n}(B_1, B_2) \) we give upper bounds for \( \sigma_{\mathbb{R}}(L_1), C_2(L_1), C_2(\mathbb{R}_{11}) \) and \( C_2(H_1) \) that do not depend on \( d \) and \( n \). First of all we need the following lemma.

Lemma 4.
Provided that \( n < n_0(d) \), the permutation matrix \( P \) in (30) can be partitioned as

\[ P = \begin{pmatrix} P_{11} & 0 \\ 0 & P_{22} \end{pmatrix} \]

with \( P_{11} \) an \( n \times n \) (permutation) matrix.

Proof.
In the computational process, \( k \) successively assumes the values

\[ k_1, k_1 + 1, \ldots, n - 1, n \]

If \( k_1 = n \), then the algorithm starts out with a decomposition of the Hankel block \( H_{n+1,n} \).

If \( k_1 \neq n \), then in the computational process a pair \( (k_i, k_j) \) occurs such that

\[(i) \quad k = n - 1 \]

\[(ii) \quad r(H_{k+1,k+1}) = r(H_{k+1,k+1}) = k + 1 \]

\[(iii) \quad \text{for } H_{k+1,k+1} \text{ a decomposition is known.} \]

From theorem 0.9 it follows that \( n \geq k \). Since also \( k \geq k + 1 = n \), this shows that \( k = n \).
Consequently, at a certain stage in the computational process a decomposition is obtained:

\[ M_{n+1,n+1} M_{n+1,n} = R_{n+1,n} + \Delta R_{n+1,n}, \]

Hereafter, \( M_{n+1,n} \) is extended with columns only; the corresponding new columns of \( R \) are possibly interchanged, but not the old columns of \( R \). So, indeed

\[ P = \begin{pmatrix} P_{11} & 0 \\ 0 & P_{22} \end{pmatrix} \]

with \( P_{11} \) an \( n \times n \) permutation matrix.

Lemma 6 shows that

\[ M_{n,n} P_{11} = L_1 (R_{11} + \Delta R_{11}), \]

Hence, if \( \gamma \) is small enough, we have the upper bound for \( C_2(R_{11}) \).

\[ C_2(R_{11}) \leq 1.1 C_2(L_1) C_2(M_{n,n}) \]

(For \( \frac{\| \Delta R_{11} \|_2}{\| R_{11} \|_2} \) as well as \( \frac{\| \Delta R_{22} \|_2}{\| R_{22} \|_2} \), the upper bound \( 1.1 \ln((n+1)^{3/2} + \gamma n \) applies).

Next, let us consider \( c^{-1}_n(L_1) \) and \( C_2(L_1) \).

We have \( L = M^{-1} T^{-1} \) or \( ML = T^{-1} \) (\( M \) is orthogonal).

Let us partition \( M \) and \( L \) as

\[ \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix}, \]

such that \( M_{22} \) and \( L_{22} \) are the elements \( M_{n+1,n+1} \) and \( L_{n+1,n+1} \) respectively (so \( L_1 = L_{11} \))

We have

\[ M_{11} L_{11} + M_{12} L_{21} = I \]

\[ M_{21} L_{11} + M_{22} L_{21} = \lambda_{21} \quad (\lambda_{21} = \Delta R_{21} (R_{11} + \Delta R_{11})^{-1}). \]

124
It follows that (since $M_{22} \neq 0$ if $\epsilon$ is small enough)

$$
L_{11}^{-1} = M_{11}^{-1} - \frac{M_{12} \delta_{21}^{-1} M_{21}^{-1}}{M_{22}},
$$

Hence, putting $y := |M_{22}|$, $\delta := \delta_{21}^{-1/2}$ and employing the orthogonality of $M$,

$$
\|L_{11}^{-1}\|_2 \leq \frac{1}{1 - \frac{\delta (1 - y^2)}{y}},
$$

provided the denominator at the right-hand side is positive.

It is given that, in the Euclidean norm, $x = H_{n_1}^{-1} \begin{bmatrix} s_{n_1} \\ s_{2n} \end{bmatrix}$ is bounded by $B_2$.

Because $(x^T, -\delta_1)\bar{H} = 0^T$ and also $\sigma_{n_1}^{-1} \bar{L}_{n_1}^{-1} \bar{H} = 0^T$, we obtain

$$
(x^*, 0) = \bar{e}_{n_1}^T L_{n_1}^{-1} (1 - e_{n_1}^T e_{n_1}) e_{n_1}^T L_{n_1}^{-1} e_{n_1} = \begin{bmatrix} \begin{bmatrix} M_{11} \\ M_{21} \end{bmatrix} \\ \begin{bmatrix} M_{12} \\ M_{22} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} -\delta_2 \end{bmatrix} \\ \begin{bmatrix} 0 \end{bmatrix} \end{bmatrix},
$$

Consequently, we have

$$
\|x\|_2 \leq \frac{\|\delta_2 L_{n_1}^{-1} + M_{21}\|_2}{\|\bar{M}_{22}\|_2} \leq B_2.
$$

Hence,

$$
\frac{\|M_{21}\|_2}{\|M_{22}\|_2} = \frac{\delta}{\|M_{22}\|_2} \leq B_2,
$$

or

$$
\frac{\|\delta_2 M_{21}\|_2}{\|\bar{M}_{22}\|_2} = \frac{\delta}{\|\bar{M}_{22}\|_2} \leq B_2.
$$

One may verify that from (35) and this last inequality it follows that

$$
\sigma_{n_1}^{-1}(L_1) = \|L_{11}^{-1}\|_2 \leq 1 + (1 + B_2)
$$

provided that $1 \delta_{21} L_{12} \|\bar{R}_{11} + \delta_1 \bar{R}_{11}\|_2 B_2 \ll 1$.

So, we have

$$
\forall \epsilon \in \mathbb{R}_+(B_1, B_2), \exists \eta_0 \forall \eta \in \mathbb{R}_0 \left\{ \sigma_{n_1}^{-1}(L_1) \leq 1 + (1 + B_2) \right\},
$$

125.
and
\[(37) \quad \forall \delta D_0(n_1, n_2) \exists \eta_0 \forall \eta < \eta_0 \quad [C_2(L, 1) \leq 1.1 \{1 + B_2\}].\]

Furthermore, with (34) we find
\[(38) \quad \forall \delta D_0(n_1, n_2) \exists \eta_0 \forall \eta < \eta_0 \quad [C_2(R_{1,1}) \leq 1.2 B_1 \{1 + B_2\}].\]

Finally, let us investigate $C_2(R_{1,1})$.

Lemma 9.

Let $A$ be a $k \times k$ matrix with $k \leq i$ and $r(A) = k$. Let $b \in \mathbb{R}^k$.

There exist numbers $m_i$ (1 ≤ $i$ ≤ $k$) such that
\[
\frac{k}{m_i} \leq 1 \quad \text{and} \quad \sigma_i^2(A \mid b) = \sigma_i^2(A) + m_i b^T b \quad (1 \leq i \leq k).
\]

Proof.

If $b = 0$, the lemma is trivial. So, let $b \neq 0$.

The singular values of $(A \mid b)$ are the square roots of the eigenvalues of $AA^T + bb^T$. The matrix $bb^T$ has rank one and a non-zero eigenvector $b^T b$.

Applying Wilkinson [5, page 98], the lemma follows.

Corollary.

\[C_2(A \mid b) \leq C_2(A) \left(1 + \frac{\|b\|^2}{\|A\|^2_2}\right).
\]

Proof.

\[C_2^2(A \mid b) = \frac{\sigma_1^2(A \mid b)}{\sigma_k^2(A \mid b)} = \frac{\sigma_1^2(A) + m_1 b^T b}{\sigma_k^2(A) + m_k b^T b} \leq C_2^2(A) \left(1 + \frac{\|b\|^2}{\|A\|^2_2}\right) \tag*{\square}
\]

Corollary.

Let $B$ be some matrix with $k$ rows. Then
\[C_2(A \mid b) \leq C_2(A) \left(1 + \frac{\|B\|^2}{\|A\|^2_2}\right).
\]
Proof.

\[ c_2^2(A, s) \leq \frac{c_2^2(A) - \|H_h^2\|^2}{c_2^2(A)} \leq c_2^2(A) \left( 1 + \frac{\|H_h^2\|^2}{\|H_h^2\|^2} \right). \]

Let \( H \) be partitioned as \( (H_{11}, H_{12}) \) where

\[
H_{11} = H_{n \times n} = \begin{pmatrix} s_1 & \cdots & s_n \\ \vdots & \ddots & \vdots \\ s_n & \cdots & s_{2n-1} \end{pmatrix} \quad \text{and} \quad H_{12} = \begin{pmatrix} s_{n+1} & \cdots & s_{n+i} \\ \vdots & \ddots & \vdots \\ s_{2n} & \cdots & s_{n+i-1} \end{pmatrix} (i = 2k - n).
\]

Because the vector \( (x^T, -1)^T \), where \( x = H_{n \times 1}^{-1} \begin{pmatrix} s_{n+1} \\ \vdots \\ s_{2n} \end{pmatrix} \), contains the coefficients of the characteristic polynomial of \( d = (s_i)_{i=1}^m \), we have (Theorem 0.3)

\[
\begin{pmatrix} s_{n+1} & \cdots & s_{n+j} \\ \vdots & \ddots & \vdots \\ s_{n+j} & \cdots & s_{2n+j-1} \end{pmatrix} x = \begin{pmatrix} s_{n+j+1} \\ \vdots \\ s_{2n+j} \end{pmatrix} \quad \text{for all } j \geq 0.
\]

We obtain

\[
\|H_{12}e_1\|_2^2 = \|H_{11}\|_2 \|x_1\|_2 \leq \|H_{11}\|_2 \|e_1\|_2 \leq \|H_{11}\|_2 \|e_2\|_2^2,
\]

\[
\|H_{12}e_2\|_2^2 = \begin{pmatrix} s_2 & \cdots & s_n \\ \vdots & \ddots & \vdots \\ s_n & \cdots & s_{2n} \end{pmatrix} x + \begin{pmatrix} 1 \\ \vdots \\ 0 \end{pmatrix} \|E_2 \|_2 \|H_{11}\|_2 \|e_2\|_2 \|H_{11}\|_2 \|e_2\|_2^2 (1 - B_2^2) B_2^2.
\]

Continuing like this we find

\[
\|H_{12}x\|_2^2 \leq \|H_{11}\|_2 \|e_2\|_2^2 (1 - B_2^2) B_2^2 \quad (1 \leq t \leq n - 1 = 2k - 2n - 1).
\]

Hence

\[
\|H_{12}\|_2^2 \leq \|H_{11}\|_2^2 (1 + B_2^2)^{2k-2n-1}.
\]

Applying Lemma 10, we conclude

\[ C_2^2(B_1) \leq C_2^2(B_1) \left( 1 + \frac{\|H_{12}\|^2}{\|H_{12}\|^2} \right) \leq B_1 (1 + B_2^2)^{2k-2n-1}. \]
After this investigation of $\pi_n^{-1}(\ell_n)$, $C_2(\ell_n)$, $C_2(\ell_n^+)$, and $C_2(\ell_n^+)$, we can prove that the algorithm is forward stable on $D_{\pi_n}(\ell_1, \ell_2)$.

As a consequence of (32) and (33), we obtain

$$\frac{\|e(e(e(d_1), f(d_1)))\|}{\|e(e(e(d_1), f(d_1)))\|} \leq \frac{\|e(e(e(d_1), f(d_1)))\|}{\|e(e(e(d_1), f(d_1)))\|} + \|\Delta U\|_2 + \frac{\|\Delta U\|_2}{\|U_1\|_2}.$$

The first term at the right-hand side of (40)

$$(U_2 + \Delta U_2) (U_1 + \Delta U_1) - U_2 U_1 =$$

$$= (\Delta U_2 + U_2 (I - U_1^T (U_1 + \Delta U_1))) (U_1 + \Delta U_1)^+ =$$

$$= (\Delta U_2 + U_2 (I - U_1^T U_1)) - U_2 U_1^T (U_1 + \Delta U_1)^+.$$

In subsection 4.1.3 we saw that $U_1 = L_1 U_1$ and $U_2 = L_1 U_2$. The row space of $U_2$ is contained in the row space of $U_1$ (a consequence of theorem 6.1).

$L_1$ is regular. Therefore, the row space of $U_2$ is contained in the row space of $U_1$.

Hence,

$$(41) \quad U_2 (I - U_1^T U_1) = 0$$

and it follows that

$$\|e(e(e(d_1), f(d_1)))\| \leq \|e(e(e(d_1), f(d_1)))\| + \|\Delta U\|_2 + \|\Delta U\|_2 = \frac{\|\Delta U\|_2}{\|U_1\|_2} + \frac{\|\Delta U\|_2}{\|U_2\|_2} + \frac{\|\Delta U\|_2}{\|U_2\|_2}.$$

Because of (41), we have

$$\|U_2\|_2 \leq \|U_2\|_1 \|U_1\|_2.$$

Therefore, if $n$ is small enough, it follows that

$$\|e(e(e(d_1), f(d_1)))\| \leq \|e(e(e(d_1), f(d_1)))\| + \|\Delta U\|_2 + \|\Delta U\|_2 = \frac{\|\Delta U\|_2}{\|U_1\|_2} + \frac{\|\Delta U\|_2}{\|U_2\|_2}.$$

Since $U_1 = L_1 U_1$ and $U_1^+ = U_1^{++}$, it holds that $C_2(U_1) \leq C_2(L_1) C_2(U_1)$.

Consequently, in view of (37) and (39), $C_2(U_1)$ has an upper bound that does
not depend on \(d\) and \(n\). Also \(\frac{\Delta U_1, U_2}{U_1, U_2} \leq \frac{\Delta U_2, U_2}{U_2, U_2}\) \(n\) has an upper bound that is independent of \(d\) and \(n\) (subsection 4.2.1).

Hence,

\[
\exists \epsilon > 0 \forall d \in \mathcal{D}(B_1, B_2) \exists \eta_0 \forall n < n_0 \left\{ \frac{\|U, U\|}{\|U, U\|} < Cn \right\}.
\]

The second term at the right-hand side of (40)

It instantly follows from (30) that

\[
\exists \epsilon > 0 \forall d \in \mathcal{D}(B_1, B_2) \exists \eta_0 \forall n < n_0 \left\{ \frac{\|\Delta u, \bar{U}\|}{\|U, U\|} < Cn \right\}.
\]

The third term at the right-hand side of (40)

We have

\[
\frac{\bar{e}_1^T}{\bar{e}_1^T} = \frac{\bar{e}_1^T}{\bar{e}_1^T} \quad \text{and} \quad \Delta \bar{e}_1^T = \frac{\bar{e}_1^T}{\bar{e}_1^T} \left[ \begin{array}{c} 0 \\ -\bar{u}_{21} (\bar{R}_{11} + \bar{R}_{11})^{-1} \end{array} \right].
\]

Therefore, if \(n\) is small enough, we have

\[
\frac{\|\Delta \bar{e}_1^T\|}{\|\bar{e}_1^T\|} \leq 1.1 \left( \frac{(\bar{e}_1^T, \bar{e}_1^T)}{\bar{e}_1^T, \bar{e}_1^T} \right) C_2(\bar{R}_{11}) \frac{\|\bar{R}_{11}^2\|}{\|\bar{R}_{11}^2\|} \leq 1.1 \frac{C_2(\bar{R}_{11})}{\sigma(\bar{L}_{11})} \frac{\|\bar{R}_{11}^2\|}{\|\bar{R}_{11}^2\|}.
\]

Employing (36) and (38), we obtain

\[
\exists \epsilon > 0 \forall \eta \in \mathcal{D}(B_1, B_2) \exists \eta_0 \forall n < n_0 \left\{ \frac{\|\Delta L\|}{\|L\|} < Cn \right\}.
\]

Combining (43), (44) and (45), we see that the algorithm is forward stable on \(\mathcal{D}(B_1, B_2)\).
Remark.

1. We tacitly assumed that $\xi_1 \neq 0$, $\nu_1 \neq 0$ and $U_2^\top \nu_1 \neq 0$. Since only non-trivial impulse responses are considered, $\xi_1 \neq 0$ and $\nu_1 \neq 0$ hold. However, $U_2^\top \nu_1$ may be 0. In that case (41) shows that $U_2 = 0$ and, therefore, we have $(s_1)_{[1]}^\infty = (s_1,0,0,\ldots)$ with $s_1 \neq 0$. The numerical stability of the algorithm for these impulse responses is readily verified.

2. We also assumed that $$(\xi_1 + \Delta \xi_1)^T = e_1^T M (1 - e_n^T M_n^+)$$ is calculated exactly. This is impossible, since we have stored the approximate plane rotations $(\mathbf{P}_r^T)$ $$(1 \leq p < q \leq k+1)$$. However, $\xi_1 + \Delta \xi_1$ can be evaluated with a relative error that depends on $n$ but not on $d \in \mathcal{D}_n (B_1, B_2)$. Hence, the numerical stability of the algorithm is still guaranteed.

3. Given any impulse response $d \in \mathcal{D}_n$, we can indicate a subset of $\mathcal{D}_n$ containing $d$ and a neighborhood of $d$ such that the algorithm is forward stable on the subset. This is certainly not true for the algorithm of Grissman. Let us consider the impulse response $d = (0,1,1,0,\ldots) \in D_2$. Although the algorithm of Grissman may do well for $d$, it does not do so for a neighbor of $d$: $d_0 = (0,1,1,0,\ldots) \in D_2$. In 3.5 we showed that the forward error in a realization triple may be of the order of $\varepsilon^{-2n}$. So the algorithm of Grissman cannot possibly be forward stable on a neighborhood of $d$.

4.3: Assessment of the algorithm.

In this section we shall investigate – given an input $(s_1)_{[1]}^\infty$ – how small $n$ should be and give an estimate for the number of operations (counting divisions and multiplications).

4.3.1.

In section 4.2 we showed that the algorithm is forward stable on the input set $\mathcal{D}_n (S_1, S_2)$:

$$(29) \quad \exists C_0 \forall d \in \mathcal{D}_n (B_1, B_2) \exists n_0 \forall n < n_0 \left[ \left| u'(f(f)(d), f(d)) \right| < C_0 u'(f(f)(0), f(0)) \right].$$

We did not bother to give a best possible value for $C_0$, because such a value would have little practical significance. However, it is important to know how large $n_0$ may be.
In subsection 4.2.2 we saw that a local value of \( n \) is used. We there saw that, if for some pair \((k,l)\) one has found that \( r_c(H_{k,l}) = r(H_{k,l}) = k \), with

\[
e = \text{tol}_{k+1, l} \| H_{k+1, l} \|_2 - 0.6 \mu_{k+1, l} (1 + \| (E_{11}^{-1} E_{12}) \|_2) \cdot \|
\]

\( r_c(H_{k+1, l}) \) can be determined using the decomposition (19) so that \( r_c(H_{k+1, l}) = r(H_{k+1, l}) \), provided that \( n \) is so small that

\[(46) \quad e \leq \frac{1}{2} \max\{0, \mu_{k+1, l} \}, \text{ if } r(H_{k+1, l}) = k + 1 \]

or

\[(47) \quad e > \frac{1}{2} (\mu_{k+1, l} + 1), \text{ if } r(H_{k+1, l}) = k \]

\[
= \frac{\| \bar{R}_{22} \|_2}{1 + \| (E_{11}^{-1} E_{12}) \|_2 + \| (E_{11}^{-1})^{-1} \|_2 \| \bar{R}_{22} \|_2} \cdot \mu_{k+1, l}
\]

If \( r(H_{k+1, l}) = k \), then \( \mu_{k+1, l} = \| R_{22} \|_2 \) may be as large as \( n \| H_{k, l} \|_2 C_2(H_{k, l}) \) and hence \( e \) should be at least of the same magnitude.

The right-hand side of (46) is of the order of \( n \| H_{k, l} \|_2 C_2^{-1}(H_{k, l}) \). Consequently, to guarantee (46), we should demand that

\[n C_2^2(H_{k, l}) \ll 1, \text{ for all eligible } (k, l) .\]

Hence, we should have (see formula (1))

\[(48) \quad n C_0^2 \ll 1 .\]

The demand (22) implies that, if \( \text{tol}_{k+1, l} \) is chosen as above,

\[(49) \quad n C_0 \max\{C_0, C_1\} \ll 1 ,\]

which is stronger than (48).

In subsection 4.2.4 we also repeatedly used the phrase "if \( n \) is small enough". One may verify that there it is sufficient that \( n C_0 \ll 1 \). So the demand that the order of a minimal realisation should be found imposes the major restriction on \( n \).
4.3.2.

The total of the operations required by the algorithm is the sum of the operations required for the decomposition, the operations for evaluating $r_e(R_{n+1,k})$, and the operations for computing the realization triple. We shall assume that it is a priori known how small $n$ should be for a particular impulse response. In that situation $r_e(R_{n+1,k})$ can be determined either by verifying (46) or by verifying (47) - see subsection 4.2.2, remark 1.

The number of operations needed for decomposing $R_{n+1,k}$ recursively is, asymptotically: $\ln^2 \frac{1}{3} n^2$. ($\frac{1}{2} n^2$ for $R_{n+1}^{11}$ and $2(\lambda-\mu)\cdot\ln(\lambda-\mu)$ for $R_{n+1}^{21}$.)

If, in order to compute an a posteriori estimate of $\delta_{n+1,k} = \| \delta R_{n+1,k} \|_2$, the entire decomposition is performed in double length too, an additional number of operations is needed: $\ln^2 \frac{1}{3} n^2$.

In case (46) is used, the evaluation of $r_e(R_{n+1,k})$ requires the computation of $\| (\tilde{R}_{11}^{-1} \tilde{R}_{12}^{-1}) \|_2$ as well as $\| (\tilde{R}_{11}^{-1}) \|_2$. In case (47) is used, it is necessary to compute $\| (\tilde{R}_{11}^{-1} \tilde{R}_{12}^{-1}) \|_2$.

An exact computation of these numbers involves $O(n^2)$ operations, which is a factor $n$ more than the total of all other operations. Therefore, one would like to be able to give reasonable sharp upper bounds for these numbers in case (46) is used, to give a reasonable sharp lower bound in case (47) is used, and to compute these bounds with comparatively little effort. From section 4.2 it is clear that also $r_e(R_{n+1,k})$ can be determined, provided that $\delta$ is small enough.

It seems that there is no way to compute a reasonable lower bound with few operations. However, it is possible to compute upper bounds for $\| (\tilde{R}_{11}^{-1} \tilde{R}_{12}^{-1}) \|_2$ and $\| (\tilde{R}_{11}^{-1}) \|_2$ using a device due to Kalasalo [4].
Lemma 10.
Let $R$ be a $(k \times k)$ upper triangular matrix with $(R)_{k,k} > 0$ $(1 \leq i \leq k)$. Let $R'$ be a $(k \times k)$ upper triangular matrix with $(R')_{k,k} = (R)_{k,k}$ $(1 \leq i \leq k)$ and $(R')_{i,j} = -1 (R)_{i,j}$ $(1 \leq i \leq j \leq k)$. Then $\|R^{-1}\|_2 \leq \|R'\|_2^{-1} \|e\|. \hfill \square$

Lemma 11.
Let $R$ be a $(k \times k)$ upper triangular matrix with $(R)_{k,k} > 0$ $(1 \leq i \leq k)$ and $(R)_{i,j} = -a_{j} (1 \leq i < j \leq k)$ with $a_k > 0$. Then
\[
\|R^{-1}\|_2^2 = \sum_{i=1}^{k} u_i (R)_{i,i}^{-2}
\]
where
\[
u_i = 1, \ u_i = (1 + D_{i-1})^2 u_{i-1} - 2D_{i-1} (2 \leq i \leq k)
\]
with
\[
D_i = a_i (R)_{i,i}^{-1} (1 \leq i \leq k-1).
\]
See Varasale [6] for the proof of these lemma's. Lemma 11 shows that $\|R^{-1}\|_2$ may be computed in about $6k$ operations.

Corollary.
Let $R$ be defined as in lemma 11 and let $b \in \mathbb{R}^k$. Then $\|R^{-1}b\|_2$ can be computed in about $4k$ operations.

Upper bounds for $\|\left(\left(\hat{R}_{11}\right)^{-1} \hat{R}_{12}\right)\|_2$ and $\|\left(\hat{R}_{11}\right)^{-1}\|_2$ may be computed as follows:

Putting $(\bar{R}_{11})_{i,i} := |(\hat{R}_{11})_{i,i}| (1 \leq i \leq k)$,

$(\bar{R}_{11})_{i,j} := -\max_{1 \leq i \leq k} |(\hat{R}_{11})_{i,k}| (1 \leq i < j \leq k)$,

then $\|\left(\bar{R}_{11}\right)^{-1}\|_2$ is an upper bound for $\|\left(\hat{R}_{11}\right)^{-1}\|_2$.

Putting $b_k := \max_{1 \leq i \leq k} |(\hat{R}_{12})_{i,k}| (1 \leq i \leq k)$,
then \[ \| ( \mathbf{R}^{-1} \mathbf{R} + \mathbf{R}^{-1} \mathbf{R} \mathbf{R}^{-1} ) \|_2 \leq 1 + (k - 1) \| \mathbf{R}^{-1} \|_2 \| \mathbf{b} \|_2. \]

It can be shown that the upper bounds may be obtained with about 9k operations. So, globally, the evaluation of \( r_\epsilon (H_{k+1, k}) \) costs about \( O(n^2) \) operations.

The number of operations for computing a realization triple is chiefly determined by the evaluation of \( (U_2 + DU_2)(U_1 + DU_1)^* \). This may be done with approximately \( 2n^2 - n^3 \) operations, which is about two times more than the number of operations needed for the decomposition.

**Remarks.**

1. Even if it is unknown whether \( n \) is small enough, we only have to verify (46) because we may compute an upper bound for \( nC^2 (H_{k+1, k}) \). If \( n \) is not small enough, this becomes evident since then the upper bound will be large.

2. A realization triple in companion form can be given with \( O(n^2) \) operations. However, as we remarked before, such a triple contains information only about the first 2n elements of \( \{ s_i \}_{i=1}^m \). Moreover, for instance the eigenvalues of a matrix are very poorly determined by its companion form.

3. It is conceivable that the magnitude of the numbers \( C_0 \) and \( C_1 \), defined by (1), may be reduced by scaling the input \( \{ s_i \}_{i=1}^m \).

If \( \{ s_i \}_{i=1}^m \) admits a realization \( (A, B, C) \), then the scaled input \( \{ a^i s_i \}_{i=1}^m \) \( (a \neq 0) \) admits the realization triple \( (A_a, B_a, C_a) \).
5. FILTERING AND APPROXIMATION

In this chapter we shall assume that we have only noisy input impulse responses to work on.

In section 5.1 we investigate under which conditions on the noise one may retrieve the order of a minimal realization of \( \{s_i^m\}_{i=1}^\infty \) from the noisy impulse response \( \{\tilde{s}_i^m\}_{i=1}^\infty \).

In section 5.2 we suppose that the noise does not satisfy these conditions.

There we study the two following problems:

(i) given an impulse response \( \{s_i^m\}_{i=1}^\infty \) and a neighbourhood of \( \{s_i^m\}_{i=1}^\infty \), to determine the minimum order of all impulse responses in that neighbourhood,

(ii) given an impulse response \( \{s_i^m\}_{i=1}^\infty \) and a number \( r \), to determine the nearest impulse response \( \{\tilde{s}_i^m\}_{i=1}^\infty \) that admits a minimal realization of order \( r \).

Throughout this chapter it is an assumption that the computations are performed exactly (or with a relative precision \( n \) that is so small that the effect of computational errors may be neglected compared to the noise).

5.1. Filtering

We consider a noisy input impulse response \( \{\tilde{s}_i^m\}_{i=1}^\infty \); the Hankel blocks associated with this response are denoted by \( \tilde{H}_{x_i,k} \). Again, let \( N \) be an upper bound for the order \( n \) of a minimal realization of the unperturbed \( \{s_i^m\}_{i=1}^\infty \).

In chapter 4 we saw that \( n \) may be found by considering a sequence of Hankel blocks \( H_{x_i,k} \) with \( 1 \leq k \leq N \) and \( r(H_{x_i,k}^+) = k \). Consequently, \( n \) may be retrieved from \( \{\tilde{s}_i^m\}_{i=1}^\infty \), if it is possible to recognize from \( \tilde{H}_{x_i,k}^+ \) whether or not \( r(H_{x_i,k}^+) = k \) and \( r(H_{x_i,k+1}^+) = k+1 \) for all \( (k,i) \) with \( 1 \leq k \leq N \) and \( k+i \leq 2N \) or, equivalently, if it is possible to recognize from \( \tilde{H}_{x_i,k}^+ \) whether or not \( r(H_{x_i,k}^+) = k \) for all \( (k,i) \) with \( 1 \leq k \leq N \) and \( k+i \leq 2N \).
Lemma 1.  
Let $A$ be a $k \times k$ matrix ($k \leq \ell$) with $r(A) = k$. Then

$$C_2(A + \Delta A) \leq C_2(A) \left(1 + \frac{\|\Delta A\|_2}{\|A\|_2} \right)$$

for all $\Delta A$ such that $C_2(A) \frac{\|\Delta A\|_2}{\|A\|_2} < 1$.

Proof.  
Let $\Delta A$ be a $k \times k$ matrix. We have

$$C_2(A + \Delta A) = \frac{\sigma_1(A + \Delta A)}{\sigma_k(A + \Delta A)} \leq \frac{\sigma_1(A) + \|\Delta A\|_2}{\sigma_k(A) - \|\Delta A\|_2} \leq C_2(A) \left(1 + \frac{\|\Delta A\|_2}{\|A\|_2} \right),$$

provided that the denominator at the right-hand side is positive.

Lemma 2.  
Let $A$ be a $k \times k$ matrix ($k \leq \ell$) with $r(A) < k$. Then

$$C_2(A + \Delta A) \leq 1$$

for all $\Delta A$ such that $r(A + \Delta A) > r(A)$.

Proof.  
Since the smallest non-zero singular value of $A + \Delta A$ is less than $\|\Delta A\|_2$ if $r(A + \Delta A) > r(A)$, we have

$$C_2(A + \Delta A) = \frac{\sigma_1(A + \Delta A)}{\sigma_k(A + \Delta A)} \leq \frac{\sigma_1(A) - \|\Delta A\|_2}{\|\Delta A\|_2} = \left(\frac{\|\Delta A\|_2}{\sigma_k(A)}\right)^{-1} = 1.$$  

Remark.  
These lemmas have no significance in case $k = 1$. If a $(1 \times 1)$ matrix has full rank, then its condition number is always equal to one. Furthermore, lemma 2 has no significance in case $A = 0$.  

136
Let
(1) \( D_0 := \max \left\{ \frac{\sqrt{e_{k,t}^2 - e_{k+1,t}^2}}{\sqrt{K_{k,t}^2}} \mid 1 \leq k \leq t, k + t \leq 2N, K_{k,t} \neq 0 \right\} \).
(2) \( C_0 := \max \{ C_2(K_{k,t}) \mid 1 \leq k \leq t, k + t \leq 2T + 1, r(K_{k,t}) = k \} \).
(3) \( C_1 := \max \{ C_2(K_{k+1,t}) \mid 1 \leq k \leq t, k + t \leq 2N, r(K_{k+1,t}) = k \} \).

(C_0 and C_1 are defined as in chapter 4.)

Lemma 1 shows that
\[ \max \{ C_2(K_{k,t}) \mid 1 \leq k \leq t, k + t \leq 2N, r(K_{k,t}) = k \} \leq C_0 \frac{1 + D_0}{1 - C_0 D_0}, \]
provided that \( C_0 D_0 < 1 \).

Lemma 2 shows that (assuming that \( r(K_{k,t}) \) has full rank)
\[ \min \{ C_2(K_{k,t}) \mid 1 \leq k \leq t, k + t \leq 2N, r(K_{k,t}) < k \} \geq \frac{1}{C_0} - 1. \]

Hence, if
(4) \[ \frac{1}{C_0} - 1 > C_0 \frac{1 + D_0}{1 - C_0 D_0} \quad \text{and} \quad C_0 D_0 < 1 \]
then, in principle it is possible to recognize from \( K_{k,t} \) whether or not \( r(K_{k,t}) = k \) (unless \( k = 1 \) or \( K_{k,t} = 0 \)). Equivalent with (4) is
(5) \[ D_0 < \frac{1}{2C_0} - 1. \]

After this introduction, let us investigate whether with the algorithm of chapter 4 we may determine \( n \) and, if so, which condition the noise should satisfy.

It is clear from the preceding chapters that \( n \) may be determined with the algorithm if
(1) an initial pair \((k,t)\) is known such that \( 1 \leq k \leq t, k + t \leq 2N \) and \( r(K_{k,t}) = k \).
(ii) for all \((k, l)\) with \(1 \leq k < l, k + l \leq 2n\), \(\text{tol}_{k,l}\) can be chosen such that
\[
\tau_{\epsilon_{k,l}}(\mathbf{H}_{k,l}) = k = \tau(\mathbf{H}_{k,l}) = k \\
(\epsilon = \text{tol}_{k,l} + \|\mathbf{H}_{k,l}\|_2)
\]
and
\[
\tau(\mathbf{H}_{k,l}) = k = \tau_{\epsilon_{k+l+1,l}}(\mathbf{H}_{k+l+1,l}) = \tau(\mathbf{H}_{k+l+1,l}) \\
(\epsilon = \text{tol}_{k+l+1,l} + \|\mathbf{H}_{k+l+1,l}\|_2)
\]
(compare this with theorems 4.1 and 4.2).

(iii) for all \((k, l)\) with \(1 \leq k < l, k + l \leq 2n\) and \(\tau(\mathbf{H}_{k,l}) = k\), we can compute \(\tau_{\epsilon_{k+l+1,l}}(\mathbf{H}_{k+l+1,l})\), with \(\epsilon = \text{tol}_{k+l+1,l} + \|\mathbf{H}_{k+l+1,l}\|_2\), from the decomposition

\[
N_{k+l+1,l} \mathbf{R}_{k+l+1,l} P = \mathbf{R}_{k+l+1,l}
\]

that is obtained by updating a decomposition of \(\widetilde{\mathbf{H}}_{k+l, l}\) (\(N\) is orthogonal, \(P\) is a permutation matrix, \(R\) is upper trapezoidal).

Ad (i).
If \(N\) is an upper bound for \(n\), then \((1,N)\) is such a pair.

Ad (ii).
From theorems 4.1 and 4.2 it follows that any \(\text{tol}_{k,l}\) such that

\[
\text{tol}_{k,l} + \max\{C_0, C_1\} < 1
\]
has the desired property.

Ad (iii).
From (6) we obtain

\[
N_{k+1,l} \mathbf{R}_{k+1,l} P = \mathbf{R}_{k+1,l} + \delta R_{k+1,l}
\]

with \(\delta R_{k+1,l} = \|\mathbf{H}_{k+1,l} - \mathbf{R}_{k+1,l}\|_2 = \|\mathbf{H}_{k+1,l}\|_2\).

As in subsection 4.2.2 it follows from (8) that

\[
\max(0, \|b - \tau_{k+1,l}\|_2) \leq \epsilon_{k+1,l}(\mathbf{H}_{k+l, l}) \leq \|b + \tau_{k+1,l}\|_2
\]

where

\[
b = \frac{\|R_{22}\|_2}{\|R_{11}^{-1} R_{12}\|_2 + \|R_{11}^{-1} R_{12}\|_2 + \|R_{22}\|_2}
\]

138
If \( t_{c_{k+1,k}} \) satisfies (7), then \( t_{c_{k+1,k}} = r(H_{k+1,k}) \) holds with
\[ r = \text{tol}_{k+1,k} \cdot \|H_{k+1,k}\|_2 \]
In order that \( t_{c_{k+1,k}} = r(H_{k+1,k}) \) can be determined from (9), we should have that in addition to (7), \( t_{c_{k+1,k}} \) should satisfy (compare subsection 4.2.2):
\[
\begin{align*}
(10a) & \quad \text{tol}_{k+1,k} \leq \frac{\max(0, tb - \delta_{k+1,k})}{\|H_{k+1,k}\|_2} \quad \text{in case } r(H_{k+1,k}) = k + 1 \\
(10b) & \quad \text{tol}_{k+1,k} > \frac{tb + \delta_{k+1,k}}{\|H_{k+1,k}\|_2} \quad \text{in case } r(H_{k+1,k}) = k.
\end{align*}
\]
With a similar reasoning as in subsection 4.2.2 one may verify that with
\[
\begin{align*}
(11) & \quad \text{tol}_{k+1,k} = \theta_{0} D_{0}(1 + \varepsilon R_{12}) \|H_{k+1,k}\|_2,
\end{align*}
\]
\( r_{c}(H_{k+1,k}) \) can be determined for all eligible \((k,t)\), provided that
\[
D_{0} \varepsilon \leq \frac{\delta_{k+1,k}}{\|H_{k+1,k}\|_2}
\]
is small enough.

Hence, the order of a minimal realization of \((s_{1})_{i=1}^{m}\) may be retrieved from \((s_{i})_{i=1}^{m}\) if \( D_{0} \) is small enough.

Similarly as in subsection 4.3.1, it follows that we should have
\[
(12) \quad D_{0} \varepsilon \max(C_{0}C_{1}) \ll 1.
\]
The algorithm may detect itself whether (12) is satisfied (see 4.3.1).

The number of operations is asymptotically \( n^{3} = \frac{1}{8}m^{3} \).

**Remark.**
The demand (12) is considerably stronger than (5). It is possible to retrieve \( n \) from \((s_{i})_{i=1}^{m}\) if \( D_{0} \max(C_{0}C_{1}) \ll 1 \) by computing the smallest singular value of \( X_{k+1,k} \) explicitly. However, this leads to a considerable increase of the total number of operations.
5.2. Approximation

5.2.1. Let \( \{s_i^n\}_{i=1}^\infty \) \( \in C \) such that \( \delta > 0 \) and a natural number \( N \) be given.

We shall discuss the problem to find a natural number \( r_N \) such that

\[ r_N = \min(n \mid n \text{ is the order of a minimal realization of } (s_i^n)_{i=1}^\infty, \text{ such that } |s_i - s_{i+N}| < \delta \text{ for } 1 \leq i \leq 2N). \]

We do not suppose that \( \{s_i^n\}_{i=1}^\infty \) is a realizable impulse response nor do we presuppose a relation between \( \{s_i^n\}_{i=1}^\infty \) and \( \delta \) However, we do assume that \( 1 \leq r_N \leq N \).

Remarks.

1. In the preceding section a noisy impulse response \( \{s_i^n\}_{i=1}^\infty \) was considered.

   It was shown that, the noise being small enough, the order of a minimal realization of the unperturbed \( \{s_i^n\}_{i=1}^\infty \) may be retrieved. Here, we do not presuppose such knowledge concerning the noise.

2. It seems a bit awkward to assume that any element of \( \{s_i^n\}_{i=1}^\infty \) may vary in an interval of the same size, disregarding the magnitude of the elements itself. However, this assumption enables us to apply the algorithm of chapter 6 be it in a modified form. An essential feature of this algorithm is that, in order that it be correct, for all \((k, \delta)\) with \( 1 \leq k \leq N, k + 1 \leq 2N \) (see subsection 4.1.1):

   \[
   r_c (N_{c,k}) = k \text{ should imply } r_c (N_{c,k+1}) = k \text{ as well as } r_c (N_{c,k+1}) \leq k.
   \]

   In chapter 4, formula (13) is a consequence of theorems 4.1 and 4.2 provided that \( c^{-1} \) is of the order of magnitude of the maximal condition number of all Hankel blocks \( N_{c,k} \) with \( 1 \leq k \leq t, k \leq 2N \) and \( r(N_{c,k}) = k \). When nothing is known concerning the magnitude of \( c \), formula (13) does not hold.

3. If the magnitude of the elements of \( \{s_i^n\}_{i=1}^\infty \) varies much, one might scale \( \{s_i^n\}_{i=1}^\infty \) and pose the problem for the scaled impulse response.
We do not use the Euclidean matrix norm but the H"ankel norm

\[(14) \quad \| H_{k+1,k} \|_2 = \max_{\| z \|_2 - 1} \| z \|_2.\]

We recall that, given \( \delta > 0 \) and \( H_{k+1,k} \),
\[
\rho(H_{k+1,k}, \delta) = \rho \text{ if and only if } \gamma_{p+1}(H_{k+1,k}) < \delta \leq \gamma_p(H_{k+1,k})
\]
and that, for some \( \varepsilon > 0 \),
\[
\rho(H_{k+1,k}, \delta) = \varepsilon \text{ stable if and only if } \gamma_{p+1}(H_{k+1,k}) < \delta \leq \gamma_p(H_{k+1,k}) - \varepsilon.
\]
The critical values \( \gamma_i(H_{k+1,k}) \) are defined by
\[
\gamma_i(H_{k+1,k}) = \sup(\alpha \mid \| H_{k+1,k} - H_{k+1,k} \|_2 < \alpha, \rho(H_{k+1,k}) \geq i),
\]
\[\text{ for } 1 \leq i \leq k + 1.\]

**Lemma 1.**
Let \( \delta > 0 \) and let \( H_{k+1,k} \) be some H"ankel block with \( 1 \leq k \leq l, k + 1 \leq 2k. \)
Then
\[
\rho(H_{k+1,k}, \delta) \geq \rho(H_{k+1,k}, \delta'), \quad \rho(H_{k+1,k}, \delta') \geq \rho(H_{k+1,k}, \delta), \text{ for all } \delta > 0.
\]

So, \( \varepsilon(*) \) has, with respect to extension of the H"ankel block by rows or columns, the same properties as the rank. The lemma also holds in another norm, but then it has no practical significance. If, for instance, \( \| H \|_2 \) is used as norm, then we have
\[
\rho(H_{k+1,k}, \delta') = k \implies \rho(H_{k+1,k}, \delta') = k.\]

However,
\[
\| H_{k+1,k} - H_{k+1,k} \|_2 < \delta \text{ does not imply } \| H_{k+1,k} - H_{k+1,k} \|_2 < \delta'.\]
Let the algorithm of chapter 4 be modified as follows.

1. Determine \( k, \delta \) such that \( 1 \leq k \leq t \), \( k + \delta \leq 2N \) and \( \varnothing(H_{k+1,t}) = \varnothing(H_{k+1,t+1}) = k \).

2. \( m := k + \delta \).

3. While \( m < 2 \times N \) do

4. Begin \( t := t + 1 \); \( m := m + 1 \); (decompose \( H_{k+1,t} \)).

5. While \( \varnothing(H_{k+1,t+1}) = k + 1 \) do

6. Begin \( k := k + 1 \); \( m := m + 1 \); (decompose \( H_{k+1,t} \)) end

7. End

Let us suppose that \( \varnothing(H_{k+1,t+1}) \) can be determined from the decomposition of \( H_{k+1,t} \) for all \((k,t)\) satisfying \( 1 \leq k \leq t \) and \( k + t \leq 2N \).

From the definition of \( r_N \) and the supposition that \( r_N \leq N \) we obtain

\[
\varnothing(H_{r_N+1,2N-t+\delta}) = \varnothing(H_{r_N+1,2N-t+\delta}) = r_N + \delta
\]

Let \( t \) be the smallest integer such that \( \varnothing(H_{r_N+1,t+\delta}) = 1 \). Since \( \varnothing(H_{r_N+1,t+\delta}) = 1 \), we have \( t \leq r_N \). Let \( k \) be the smallest integer such that \( \varnothing(H_{k+1,t+\delta}) = k \) (\( k \) exists and \( k \leq \delta \)). We have \( k \leq t \leq r_N \) and thus \( k + t \leq 2r_N \leq 2N \). Hence, we can determine an initial pair \((k,\delta)\) such that

\[
\varnothing(H_{k+1,t+\delta}) = \varnothing(H_{k+1,t+\delta}) = k \text{ and } k \leq \delta, \ k + \delta \leq 2N
\]

The finiteness of the computational process is proved similarly as in 3.1.

Let us suppose that just before the execution of the while-statement in line 5 we have a pair \((k,\delta)\) such that

\[
\varnothing(H_{k+1,t+\delta}) = \varnothing(H_{k+1,t+1}) = k \text{ and } k \leq \delta, \ k + \delta \leq 2N
\]

From lemma 3 we obtain that then

142
\[ \sigma(R_{k+1}^{\varepsilon, \delta}) \geq k. \]

If \( \sigma(R_{k+1}^{\varepsilon, \delta}) = k \), then (15) holds after the execution of the while-statement in line 5.

If \( \sigma(R_{k+1}^{\varepsilon, \delta}) = k+1 \), then obviously \( k+1 \leq r_N \).
Since \( \sigma(R_{k+1}^{\varepsilon, \delta}) < \sigma(R_{k+1}^{\varepsilon, \delta}) \) we also have \( k \leq r_N \).
Hence, after \( k := k + 1 \) (line 6) it holds that
\[ \sigma(R_{k}^{\varepsilon, \delta}) = k \text{ with } k \leq \delta \leq r_N. \]

It is easily seen that hereafter this relation cannot be changed by the while-statement in line 5. Hence, (15) holds after the execution.

Formula (15) holds just before the execution of line 3.

If \( m = k + 1 = 2^n \), then (15) holds at the end of the computational process.

If \( m = k + 1 < 2^n \), then \( k := k + 1 \) (line 4) and, thus, \( k \leq \delta \), \( k + 1 \leq 2N \).
Because of (15) we have just before the execution of line 5
\[ \sigma(R_{k}^{\varepsilon, \delta}) = \sigma(R_{k+1}^{\varepsilon, \delta}) = k \text{ and } k \leq \delta \leq k + 1 \leq 2N. \]

From the preceding paragraph it follows that (15) holds just before line 7.

Consequently, (15) is an invariant relation for the while-statement in line 3. Hence, at the end of the computational process we have a pair \((k, \varepsilon)\) such that (15) holds with \( k + 1 = 2N \).

It follows that \( \{ a_{k+1}^{[k+1]} \} \) exists such that
\[ \tau(R_{k}^{\varepsilon, \delta}) = \tau(R_{k+1}^{\varepsilon, \delta}) = k \text{ and } H_k^{1, \varepsilon, \delta} - H_{k+1}^{1, \varepsilon, \delta} < \delta. \]

Secondly, it follows that for all Hankel blocks \( H_k^{1, \varepsilon, \delta} \)
\[ \| H_{k+1}^{1, \varepsilon, \delta} - H_{k+1}^{1, \varepsilon, \delta} \| < \delta \text{ implies } \tau(R_{k}^{1, \varepsilon, \delta}) \geq k. \]

\( \{ a_{k}^{[k+1]} \} \) admits a partial minimal realization of order \( k \) and may be extended to \( \{ a_{k}^{[k+1]} \} \) such that the infinite sequence admits a complete minimal realization of order \( k \).
Because in a \( \delta \)-neighbourhood of \( \{ a_{k}^{[k+1]} \} \), no sequence occurs that admits a minimal realization of lower order than \( k \), we must have \( r_H = k \).
In chapter 2 we argued that it is advisable to determine not only the value of \( \rho(H_{k+1,1,\epsilon}) \) but also whether \( \rho(H_{k+1,1,\epsilon}^\epsilon) \) is \( \epsilon \)-stable for some \( \epsilon > 0 \).

When finally, \( \rho(H_{k+1,1,\epsilon}^\epsilon) \) is \( \epsilon \)-stable, then an increase of \( \epsilon \) with \( \epsilon \) will supply the same \( \gamma_{k+1} \). If \( \rho(H_{k+1,1,\epsilon}^\epsilon) \) is not \( \epsilon \)-stable than an increase of \( \epsilon \) with \( \epsilon \) will supply a smaller \( \gamma_{k+1} \) (we would prefer the \( \epsilon \)-stable rank were it not that lemma 3 does not apply for the \( \epsilon \)-stable rank).

The determination of \( \rho(H_{k+1,1,\epsilon}^\epsilon) \).

Let \( \epsilon > 0 \) (\( \epsilon = \frac{1}{6} \) say).

Whenever \( \rho(H_{k+1,1,\epsilon}^\epsilon) \) is determined, it holds that \( \rho(H_{k+1,1,\epsilon}^\epsilon) = k \).

Because \( \rho(H_{k+1,1,\epsilon}^\epsilon) = k \) (lemma 3) we have
\[
\rho(H_{k+1,1,\epsilon}^\epsilon) = k + 1 \text{ if } \epsilon \leq \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon),
\]
in which case \( \rho(\ast, \ast) \) is \( \epsilon \)-stable when \( \epsilon + \epsilon \leq \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \), or
\[
\rho(H_{k+1,1,\epsilon}^\epsilon) = k \text{ if } \epsilon > \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon),
\]
in which case \( \rho(\ast, \ast) \) is \( \epsilon \)-stable when \( \epsilon + \epsilon \geq \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \).

Consequently, first of all from the decomposition of \( H_{k+1,1,\epsilon}^\epsilon \) a lower bound for \( \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \) is determined. Then, if it cannot be decided what \( \rho(\ast, \ast) \) is, also an upper bound for \( \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \) is determined as well as a lower bound for \( \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \). Because \( \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \leq \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \) (lemma 3) a lower bound for \( \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \) is already available. If even then it cannot be decided what \( \rho(\ast, \ast) \) is, the algorithm has failed.

The determination of \( \gamma_{k+1}(H_{k+1,1,\epsilon}^\epsilon) \).

We have to know in which neighbourhood of \( H_{k+1,1,\epsilon}^\epsilon \) a Hankel block of lower rank occurs and in which neighbourhood of \( H_{k+1,1,\epsilon}^\epsilon \) all Hankel blocks have the same rank as \( H_{k+1,1,\epsilon}^\epsilon \).
Theorem 6.

Let $H_{k+1,L}$ be a Hankel block ($k+1 \leq L$). Let $H^L_{k+1,k}$ be the nearest Hankel block to $H_{k+1,k}$ such that $r(H^L_{k+1,k}) < k+1$. Then

$$
\| H_{k+1,k} - H^L_{k+1,k} \|_2 = \min_{X \in \mathbb{R}^{k+1 \setminus \{0\}}} \left\{ \| X^T H_{k+1,k} X \|_2 \right\}
$$

where $X$ is the $k \times (2+k)$ matrix, derived from the vector $x$:

$$
X = \begin{bmatrix}
x_1 & \cdots & x_k \\
\vdots & \ddots & \vdots \\
x_k & \cdots & x_1 \\
0 & \cdots & 0
\end{bmatrix}
$$

In addition, if $H_{k+1,k} = (e_{1+2\cdots j} \cdots e_{1+k})$, $H^L_{k+1,k} = (e_{1+2\cdots j} \cdots e_{1+k})^T$ and $x$ minimizes (16), then

$$
(s, \ldots, s^T) = (I - XX^T)(e_1, \ldots, e_{k+1})^T.
$$

Proof.

If $r(H_{k+1,k}) < k+1$, the theorem is trivial. So, suppose that $r(H_{k+1,k}) = k+1$. Let $x \in \mathbb{R}^{k+1}$ be non-zero and let $\tilde{H}_{k+1,k}$ be a Hankel block such that $x^T \tilde{H}_{k+1,k} = 0^T$ (such a Hankel block does exist; its rank is necessarily less than $k+1$).

Let $F_{k+1,k} := (e_{1+2\cdots j} \cdots e_{1+k}) - H_{k+1,k} - \tilde{H}_{k+1,k}$.

We have

$$
x^T F_{k+1,k} = x^T \tilde{H}_{k+1,k}.
$$

Hence,

$$
X \tilde{X} = Xs,
$$

where $X$ is defined by (17), $s = (e_1, \ldots, e_{k+1})^T$, $s = (s_1, \ldots, s_{k+1})^T$.

Considering (19) as a system of linear equations in $s$, the solution of (19) having minimal Euclidean length is given by

$$
\bar{s} - x^T \bar{x} = X^T H_{k+1,k} x.
$$
Minimizing $f$ in (20) over all non-zero $x \in \mathbb{R}^{k+1}$ supplies the nearest Hankel block $H^*_{k+1,k+1}$ with $\mathcal{R}(H^*_{k+1,k+1}) \subset k+1$. The assertions (16) and (18) are now easily obtained.

Note that $s^*$ is obtained by projecting $s$ on the orthogonal complement of the row space of $X$ (the kernel of $X$).

**Corollary.**
If $k+1 = 2$, then

\[H^*_{k+1,k+1} = H^*_{k+1,k+1} \succeq \sigma_{k+1}(H^*_{k+1,k+1}) \cdot \]

**Proof.**

$H^*_{k+1,k+1}$ has an eigenvector $x$ and corresponding eigenvalue $\lambda$ such that $|\lambda| = \sigma_{k+1}(H^*_{k+1,k+1})$. Employing (16) it shows that

\[\|H^*_{k+1,k+1} - H^*_{k+1,k+1}\|_2 \leq \sigma_{k+1}(H^*_{k+1,k+1}) \|x\|_2,\]

where $x$ is the $(k+1) \times (2k+1)$ matrix,

\[X = \begin{bmatrix} x_1 & \cdots & x_{k+1} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & x_1 & x_{k+1} \end{bmatrix} .\]

Since $X$ has full row rank, we have $X^* = X^*(XX^*)^{-1}$ and $\|X^*x\|_2^2 = x^*(XX^*)^{-1}x$.

The matrix $XX^T$ is a symmetric Toeplitz matrix, i.e. there exist constants $\xi_i$ $(0 \leq i \leq k)$ such that

\[(XX^T)_{i,j} = \xi_{|i-j|} \quad (1 \leq i \leq k+1; \ 1 \leq j \leq k+1) .\]

Hence, if $P_1$ is the $(k+1) \times (k+1)$ permutation matrix such that

\[(1, \ldots, k, k+1)P_1 = (k+1, k, \ldots, 1)\]

then

\[XX^T = P_1^T(XX^T)P_1 = (P_1^T X P_1^T)^T .\]
If $P_2$ is the $(2k+1) \times (2k+1)$ permutation matrix such that 
\[(1, \ldots, 2k, 2k+1)P_2 = (2k+1, 2k, \ldots, 1)\]

then 
\[(P_2^T)(P_2^T)^T = (P_2^T)P_2P_2^T(P_2^T)^T = (P_2^T)(P_2^T)^T.
\]

We have 
\[
\bar{x}_i = \bar{x}_{i+1} \begin{bmatrix} x_{k+1} & \cdots & x_1 & 0 \\
0 & \ddots & \vdots & \vdots \\
& \ddots & x_{k+1} & \cdots \end{bmatrix}
\]

and 
\[
\|x\|^2 = x^T(x^T)^{-1}x = x^T(x^T)^{-1}x = \|\bar{x}\|^2.
\]

$\bar{x}$ is the solution of $\bar{x}y = x$ with minimal length. 
$y = e_{k+1}$ is a solution of $\bar{x}y = x$.

Therefore, 
\[
x^T(x^T)^{-1}x = \|\bar{x}\|^2 \leq \|e_{k+1}\|^2 = 1
\]

and the assertion follows.

Remark.
If $k+1 \neq t$, then as can be shown by examples, $\|H_{k+1,s}^{-1} - H_{k+1,s}^x\|_2$ may be greater than $\|e_{k+1}\|_{k+1}$.

We return to the determination of a lower and an upper bound for the critical value $\gamma_{k+1}(H_{k+1,s})$. Let us suppose that $\tau(H_{k+1,s}) = k+1$.

Let $H_{k+1,s}$ be defined as in theorem 4 and let $\bar{H}_{k+1,s}$ be the Hankel block with lower rank in the $\|\cdot\|_2$ norm. We have 
\[
\gamma_{k+1}(H_{k+1,s}) = \|H_{k+1,s} - \bar{H}_{k+1,s}\|_2 = \|H_{k+1,s} - H_{k+1,s}^x\|_2 = \|H_{k+1,s} - H_{k+1,s}^x\|_2.
\]

Hence, (16) provides a possibility to give upper bounds for $\gamma_{k+1}(H_{k+1,s})$. 

147
Let $H_{k+1, t}$ be a Hankel block. We have for all $\Delta H_{k+1, t}$

$$\|\Delta H_{k+1, t}\|_2 < u_{k+1}(H_{k+1, t})$$
implies $\nu(H_{k+1, t} + \Delta H_{k+1, t}) = k+1$.

Because $\|y\|_2 \leq \sqrt{k+1} \|\Delta y\|_2 \leq \sqrt{k+1} \|\Delta H_{k+1, t}\|_2$, it follows that for all $\Delta H_{k+1, t}$

$$(22) \|\Delta H_{k+1, t}\|_2 < \frac{u_{k+1}(H_{k+1, t})}{\sqrt{(k+1)(k+2)}}$$
implies $\nu(H_{k+1, t} + \Delta H_{k+1, t}) = k+1$.

So, the right-hand side of (22) provides a lower bound for $\nu(H_{k+1, t})$.

Let $H_{k+1, t}$ have a decomposition

$$(23) \quad M_{k+1, t} H_{k+1, t} = R_{k+1, t},$$

where $M_{k+1, t}$ is orthogonal, $P$ is a permutation matrix and $R_{k+1, t}$ has upper trapezoidal form with non-zero diagonal.

It follows from (22) that a lower bound for $\nu(H_{k+1, t})$ may be obtained by computing a lower bound for $\nu(M_{k+1, t} H_{k+1, t}) = \nu(R_{k+1, t})$ (see chapter 2).

When $k+1 = t$, (21) shows that an upper bound for $\nu(H_{k+1, t})$ may be obtained by computing an upper bound for $\nu(M_{k+1, t} H_{k+1, t})$ (see chapter 2).

When $k+1 \neq t$, an upper bound for $\nu(H_{k+1, t})$ has to be found in another way. This is done as follows. Taking for $\mathbf{x}$ the last row of $H_{k+1, t}$ according to theorem 4 an upper bound for $\nu(H_{k+1, t})$ is $\|\mathbf{x}^T H_{k+1, t} \mathbf{x}\|$; we expect this to be quite a realistic upper bound (the last row of $H_{k+1, t}$ is close to a singular matrix). This technique is also applicable if $k+1 = t$.

Remark.

Once more we point out that the algorithm fails whenever for some pair $(k, t)$ it cannot be decided what the value of $\nu(H_{k+1, t})$ is.

5.2.2.

Let $(r, m)$ and two natural numbers $(r, m)$ with $2r < m$ be given.

We study the problem of finding an impulse response $\{g_r^{(m)}(t)\}$ such that

$$\{g_r^{(m)}(t)\}_{t=0}^{m}$$

admits a minimal realization of order $r$ and

$$\left(\frac{m}{2r} \right) \left( s_1 - \frac{m}{2r} \right)^{-\frac{r}{2}}$$
is minimal.
We consider the associated Hankel block \( \tilde{H}_{r+1, m-r} \). Let \( \tilde{H}_{r+1, m-r} \) be the Hankel block that is the nearest to \( \tilde{H}_{r+1, m-r} \) in the \( \| \cdot \|_2 \) norm satisfying

\[
| r(\tilde{H}_{r+1, m-r}^H - \tilde{H}_{r+1, m-r}) |^2 = \sum_{i=1}^s \left( s_i - \tilde{s}_i \right)^2 .
\]

If we can determine \( \tilde{H}_{r+1, m-r} \) and if \( r(\tilde{H}_{r+1, m-r}) = r \), then the problem is solved, because \( r(\tilde{H}_{r+1, m-r}) = r(\tilde{H}_{r+1, m-r}) = r \) implies that \( \{ \tilde{s}_i \} \) admits a partial minimal realization of order \( r \) (Theorem 0.9).

According to Theorem 4 we have

\[
\| \tilde{H}_{r+1, m-r} - \tilde{H}_{r+1, m-r} \|_2^2 = \min_{x \in \mathbb{C}^{l \times (l+1)}} g(x) ,
\]

where \( g(x) = \| \tilde{x}^T X \tilde{x} \|_2^2 = s \tilde{x}^T (XX^T)^{-1} \tilde{x} \) and where the matrix \( X \) is defined as in the theorem. If \( \tilde{x} \) denotes the optimal \( x \), then \( \tilde{x} = (I - \tilde{x}^T \tilde{x}) \tilde{x} \).

\( g(x) \) is a known differentiable function of \( x \).

We have

\[
7g(x) = 2\tilde{x}^T \tilde{x} - u ,
\]

where \( \tilde{H}_{r+1, m-r} \) is a Hankel block associated with \( \tilde{x} = (I - \tilde{x}^T \tilde{x}) \tilde{x} \) and \( u = (\tilde{x}^T \tilde{x}) \tilde{x} \).

Applying one of the many minimization procedures available in the literature, a good choice of the starting vector \( x_0 \) (preferably chosen to be the last row of \( \tilde{H}_{r+1, m-r} \)) in a decomposition as \( \text{diag}(s) \tilde{x} \tilde{x}^T \tilde{x} \tilde{x}^T \) will supply the optimal \( \tilde{x} \).

If the last component of \( \tilde{x} \) is non-zero then \( r(\tilde{H}_{r+1, m-r}) = r(\tilde{H}_{r+1, m-r}) = r \) and the problem is solved.

If the last component of \( \tilde{x} \) is zero then \( r(\tilde{H}_{r+1, m-r}) = r - 1 \) and \( r(\tilde{H}_{r+1, m-r}) = r \).

Let \( \tilde{z} \) be the first row of \( \tilde{H}_{r+1, m-r} \) that is linearly dependent on the foregoing rows. We have

\[
r(\tilde{H}_{r+1, m-r}) = r(\tilde{H}_{r+1, m-r}) = r .
\]

It easily follows that then

\[
r(\tilde{H}_{r+1, m-r}) = r + 1 .
\]
Consequently, whatever choice is made for $\tilde{z}_{m-1}, \ldots, \tilde{z}_{2m-2r+1}$, it holds that (independent of the particular value of $t$)

$$r(\tilde{z}_{m-1}, \ldots, \tilde{z}_{m-r+1}) = n - r + 1.$$ 

We therefore advise, in case the last component of $\tilde{x}$ is zero, to take either a lower value of $r$ or a higher value of $m$.

Remarks.

1. The obtained $\tilde{z}$ depends on the value of $m$, even if it is known that the given impulse response $\{s_{k}\}_{k=-\infty}^{\infty}$ admits a minimal realization of order $n$ with $r < n$ and $2n \leq m$. This can be shown by examples.

We shall not investigate the behavior of

$$\frac{\sum_{l=1}^{m} (s_{l} - \tilde{s}_{1})^{2}}{\sum_{l=1}^{m} s_{l}}$$

as $m$ approaches infinity. We expect that this quantity, whether $\{s_{k}\}_{k=-\infty}^{\infty}$ is the impulse response of an asymptotically stable or unstable system, tends to a finite limit.

2. It requires further investigation to see how the methods of this chapter generalize to systems with multiple input and output terminals.

3. In [3], Zeiger and McEwen propose a method to compute approximate linear realizations of given dimension via a singular value decomposition of the Hankel block $H_{n,m}$. Suppose this decomposition is

$$H_{n,m} = U \Sigma V^{T},$$

with $U$ and $V$ orthogonal, $\Sigma$ a diagonal matrix with the singular values of $H_{n,m}$ as diagonal elements.

Replacing $\Sigma$ by $\Sigma'$ that is obtained by setting the $(n-r)$ smaller diagonal elements of $\Sigma$ to zero, the matrix $U \Sigma' V^{T}$ is a matrix of rank $r$ that is as close as possible to $H_{n,m}$ in the $\| \cdot \|_2$ norm or the $\| \cdot \|_{H}$ norm. Pretending that $U \Sigma' V^{T}$ is a Hankel matrix (which it is not), a minimal realization triple of order $r$ is constructed that is expected to have an impulse response close to the given one.
4. In [2], Applewhip attacks the approximation problem from a different angle. Although his results may be of theoretical interest, the proposed algorithm has no practical significance.
REFERENCES


Chapter 1


real symmetric matrix. Oak Ridge (Tenn.), Oak Ridge National
Laboratory, Rep. ORNL-1574.


Springer-Verlag, 1972.

217-232.


Chapter 2


[2] Eckart, C. and G. Young: The approximation of one matrix by another of

[3] Fan, K.: Maximum properties and inequalities for the eigenvalues of
(1951), 760-766.


Chapter 3


Chapter 4


Chapter 3


SANERVATTING

Een constante lineair discreet dynamisch systeem is een (fysieke) entiteit die signalen ontvangt en uitspunt. De uitgaande signalen zijn afhankelijk van de toestand waarin het systeem verkeert en van de binnenkomende signa- len. Een dergelijk systeem wordt beschreven door de differentievergelij- kingen

\[
\begin{aligned}
    x(t+1) &= Ax(t) + bu(t) \\
    y(t) &= c^Tx(t) .
\end{aligned}
\]

De parameter \( t \) (de tijd) neemt waarden aan uit de verzameling \( \{\ldots,-1,0,1,\ldots\} \). De vector \( x(t) \) bepaalt de toestand van het systeem op de tijd \( t \). De parameter \( u(t) \) representeert het op de tijd \( t \) binnenkomende signaal en de parameter \( y(t) \) het op de tijd \( t \) uitgaande signaal.

\( A \) is een constante matrix; \( b \) en \( c \) zijn constante vectoren (in principe kunnen meerdere signalen tegelijkertijd binnenkomen en uitgezonden worden; in dat geval zijn \( u(t) \) en \( y(t) \) vectoren, en \( b \) en \( c \) matrices; deze mogelijkheid wordt in dit proefschrift buiten beschouwing gelaten).

Als op een tijdstip \( t = t_0 \) een impuls ontvangen wordt (dus \( u(t) = 0 \) voor \( t \neq t_0 \) en \( u(t_0) = 1 \)), terwijl het systeem in rust is (dus \( x(t_0) = 0 \)), dan worden de volgende signalen uitgezonden:

\[
y(t_0 + 1) = c^Tb, \quad y(t_0 + 2) = c^TAb, \quad y(t_0 + 3) = c^TA^2b, \ldots .
\]

De rij \( (c^TA^{t-1}b)_{t=1}^{\infty} \) wordt de impuls responsie genoemd.

Het realisatieprobleem is nu om bij gegeven impuls responsie een triple \((A,b,c)\) te bepalen dat in staat is de gegeven impuls responsie te genereren, redelijk dat \( A \) een matrix is van zo klein mogelijke afmetingen. Zo'n triple \((A,b,c)\) heet een minimale realisatie. Dit probleem is klassiek bij het in andere formuleringen, bijvoorbeeld:

als de getallen \( c^Tb, c^TAb, \ldots \) gegeven zijn, bepaal dan de eigenwaarden van de matrix \( A \). De klassieke algoritmen waarmee dit probleem kan worden opgelost werken alleen voor speciale rijen \((c^TA^{t-1}b)_{t=1}^{\infty}\) of zijn numeriek instabiel.
In 1965 is de eerste algoritme gepubliceerd waarmee het realisatieprobleem in de algemene formulering (meer-dimensional ingang en uitgang) kan worden opgelost (Ho, Kalman). Ondertussen dezelfde tijd zijn onafhankelijk hiervan door Silverman en Youla vervante algoritmen gepubliceerd. Riissanen heeft in 1971 een algoritme voorgesteld waarmee het realisatieprobleem, indien het een oplossing heeft, recursief wordt opgelost. De algoritme berekent partiële minimale realisaties van opeenvolgende, steeds groter vordende, eindige stukken van de impuls responsie. Iedere nieuwe partiële realisatie kan met weinig moeite uit de voorafgaande berekend worden. Uiteindelijk wordt een partiële minimale realisatie verkregen die tevens een volledige minimale realisatie is.

Het voordelen van een recursieve algoritme is dat daarmee het partiële realisatieprobleem (slechts een eindig stuk van de impuls responsie is gegeven) kan worden opgelost en dat hij gebruikt kan worden als de impuls responsie ruim bevat.

Het nadeel van de algoritme van Riissanen is dat hij numeriek instabiel is. Het is onmogelijk om bovengrenzen te geven voor de doorwerking van afrondfouten. In dit proefschrift wordt een variant op de algoritme van Riissanen besproken die wel numeriek stabiel is.

Daartoe wordt allereerst in de hoofdstukken 2 en 3 ingegaan op de begrippen numerieke stabilitiet en numerieke rang. Er wordt gepoogd een formele definitie van deze begrippen te geven.

Vervolgens worden in hoofdstuk 3 een aantal recursieve algoritmen onderzocht, die numeriek instabiel blijken te zijn. De redenen hiervoor worden opgespoord en leiden tot een nieuwe realisatie-algoritme (hoofdstuk 4) die numeriek stabiel is.

Ten slotte wordt in hoofdstuk 5 aandacht besteed aan de meer met de praktijk overeenkomende situatie dat de gegeven impuls responsie meetrui bevat. In het proefschrift worden alleen systemen met één ingang en één uitgang beschouwd. Hoewel systemen met meerdere in- en uitgangen zeker niet triviale generalisaties hiervan zijn, menen we dat de verschafte informatie voldoende is voor het construeren van een numeriek stabiele algoritme voor het algemene geval.
CURRICULUM VITAE

STELLINGEN

I

Een $n$-graads polynoom $P(x)$ en al zijn genormaliseerde afgeleiden $p^{(i)}(x)/i!$ ($1 \leq i \leq n$) kunnen worden berekend in

$$2n + \left\lceil \frac{n+1}{2} \right\rceil = 2$$

vermenigvuldigingen/delingen.

Opmerking: Horner’s algoritme heeft $n(n+1)$ vermenigvuldigingen nodig.
Shaw/Traub: het kan in $3n-2$ vermenigvuldigingen/delingen.
van Leunen: het kan in $3n-3$ vermenigvuldigingen/delingen.


II

Indien het minmax probleem

$$\min \|Ax - b\|_\infty$$

wordt opgelost door het Simplex algoritme toe te passen op het lineaire programmeringsprobleem

maximaliseren $f = (b^T, -b^T)\mathbf{w}$ onder de voorwaarden:

$$\begin{bmatrix}
A^T & -A^T \\
A^T & -\mathbf{a}^T
\end{bmatrix} \mathbf{w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$(e^T = (1, \ldots, 1)),$$

$\mathbf{w} \geq 0,$

dan is het essentieel voor het vermijden van "cycling" dat aan de actuele basis steeds die kolom wordt toegevoegd die de objectfunctie $f$ maximaal doet toenemen.

Stel $n$ en $N$ zijn natuurlijke getallen en $D$ is de verzameling

$$D = \{(C_{-N}^n, C_{-N+1}^n, \ldots, C_n^N) \mid C_i \in \mathbb{R}^n_n \ (N \leq i \leq N)\}.$$ 

We beschouwen een afbeelding $F: D \to \mathbb{R}$ die gedefinieerd is door

$$F(C_{-N}^n, \ldots, C_n^N) = \sum_{k=-N}^N \left\| C_i A_k \right\|_2.$$ 

Hierbij zijn de $A_k$'s bekende $n \times n$ matrices ($-N < k < N$) en we veronderstellen dat $A_0 = I$ en dat $\sum_{k=-N}^N \left\| A_k \right\|_2 < 1$.

Indien $F$ beperkt wordt tot de deelverzameling van $D$ waarvoor $\sum_{i=-N}^N C_i A_k^{-1} = I$, dan neemt $F$ op die verzameling een minimum aan. In het minimum geldt

$$\sum_{i=-N}^N C_i A_k^{-1} = 0 \quad (k = -N, \ldots, -1, 1, \ldots, N)$$

($\| . \|$ is een matrixnorm waarvoor geldt $\| I \| = 1$).

IV

Het berekenen van een product van een groot aantal vlakke rotatiemattices in een strikte volgorde door gebruik te maken van recurrente betrekkingen die bestaan tussen de elementen van de productmatrix, weegt qua efficiëntie niet op tegen het rechtstreekse uitersteken van de productmatrix indien daarbij gebruik gemaakt wordt van de verzellingsmethodiek van Gentleman en Hammarling.


V
Stel \( P(z) = a_0 z^n + \ldots + a_{n-1} z + a_n \) waarbij \( |a_0| = |a_n| \neq 0 \). De toegevoegd complexe van een getal \( z \) noteren we door \( \overline{z} \).
Stel \( P^*(z) := z^n P(z^{-1}) \) en \( TP(z) := \overline{a_0} P(z) - a_0 z^n P^*(z) \).
Indien op de eenheidscirkel \( C_1 \) geldt dat \( TP(z) \neq 0 \), dan heeft \( TP(z) \) buiten \( C_1 \) evenveel nolpunten als \( P(z) \).

VI
Stel \( P_n(z) = x_0 z^n + \ldots + x_{n-1} z + x_n \) en \( P_y(z) = y_0 z^n + \ldots + y_{n-1} z + y_n \).
Stel\[
X = \begin{bmatrix}
x_0 & \cdots & x_n & 0 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & x_{n-1} & x_n
\end{bmatrix} \quad \text{en} \quad Y = \begin{bmatrix}
y_0 & \cdots & y_n & 0 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & y_{n-1} & y_n
\end{bmatrix}.
\]
Dan geldt
\[
\text{rang} \begin{bmatrix} X \\ Y \end{bmatrix} = 2n + 1 - \text{graad} (\gcd(P_n, P_y)).
\]

VII
Stel \( H \) is een reguliere \( n \times n \) Hankel matrix (dus \((H)_{i,j} = s_{i+j-1}\) voor \(1 \leq i, j \leq n\)).
Stel \( H \) is een \( n \times n \) Hankel matrix met \( r(H) < n \) en \( \frac{2n-1}{3} \sum_{i=1}^{2n-1} (s_i - \overline{s_i})^2 \) minimaal.
Dan is
\[
P(z) := (s_1 - \overline{s_1}) z^{2n-2} + (s_2 - \overline{s_2}) z^{2n-3} + \ldots + (s_{2n-1} - \overline{s_{2n-1}})
\]
het kwadraat van een polynoom \( Q(z) \).

VIII
Zij gegeven de rij \( \{s_i\}_{i=1}^{m} \) met \( s_i \in \mathbb{C} \).
We definieeren voor alle \( k \geq 1 \) en \( t \geq 1 \) de gegeneraliseerde Hankel matrices
\[
E_{k,t} = \begin{bmatrix}
s_1 & \cdots & s_k \\
\vdots & \ddots & \vdots \\
s_k & \cdots & s_{k+t-1}
\end{bmatrix}.
\]
Stel het paar

\[(\bar{a}, a) := \min (\{i, n\} \mid \text{rang}(H_{i,n}) = \text{rang}(H_{\bar{a}, n}) \text{ voor } i \geq 0, j \geq 0)\]

bestaat.
Indien \((k, l)\) een paar is dat voldoet aan

\[k \leq \bar{a}, \ t \leq a\]

en indien \(t\) voldoet aan

\[\text{rang}(H_{k,t}) = \text{rang}(H_{k+1,t}) \text{ voor } k \leq j \leq t\]

en

\[\text{rang}(H_{k,1}) < \text{rang}(H_{k+1,1})\]

dan volgt

(i) \[a \geq t\]

(ii) \[b \geq k + 2 \text{ (als } t - 1 > 0) \text{ of } b \geq k \text{ (als } t - 2 \leq 1)\].

(Dit proefschrift, stelling 0.9.)

IX

Voordat met de bouw van grote gebouwencomplexen wordt begonnen, dienen de aerodynamische eigenschappen onderzocht te worden met het oog op het gemak waarmee de gebouwen bereikt en gepasseerd kunnen worden door voetgangers en fietseren.

X

Het maakt elkaar voorkomen van de woorden kennis en macht in de leuze "spreiding van kennis en macht" doet de gedachte opkomen dat het hebben van macht iets anders is dan het hebben van kennis.

Eindhoven, 1 juli 1975

L.S. de Jong