Euclid, Padé and Lanczos, another golden braid

Adhemar Bultheel and Marc Van Barel

Report TW 188, April 1993

Department of Computer Science, K.U. Leuven

Abstract

We describe how the Euclidean algorithm can be interpreted as a method to solve Padé approximation problems in a table with singular blocks and minimal partial realization problems of linear system theory. Its recursive structure also relates it to continued fraction expansions of formal series and generalizations of three term recurrence relations of formal (block) orthogonal polynomials. Hence, it can also be considered as a fast Gram-Schmidt procedure for orthogonalization with respect to an indefinite Gram matrix with Hankel structure and a fast Hankel factorizer in triangular factors. Finally, the link with the Lanczos algorithm for nonsymmetric matrices is made.

Keywords : Krylov subspace methods, formal orthogonal polynomials, Padé approximation, Hankel matrices, Toeplitz matrices, Lanczos method, linear systems, eigenvalues

AMS(MOS) Classification : Primary : 65F10 Secondary : 41A21 65F25 13F07 93B20

EUCLID, PADÉ AND LANCZOS, ANOTHER GOLDEN BRAID

ADHEMAR BULTHEEL AND MARC VAN BAREL *

Abstract. We describe how the Euclidean algorithm can be interpreted as a method to solve Padé approximation problems in a table with singular blocks and minimal partial realization problems of linear system theory. Its recursive structure also relates it to continued fraction expansions of formal series and generalizations of three term recurrence relations of formal (block) orthogonal polynomials. Hence, it can also be considered as a fast Gram-Schmidt procedure for orthogonalization with respect to an indefinite Gram matrix with Hankel structure and a fast Hankel factorizer in triangular factors. Finally, the link with the Lanczos algorithm for nonsymmetric matrices is made.

Key words. Krylov subspace methods, formal orthogonal polynomials, Padé approximation, Hankel matrices, Toeplitz matrices, Lanczos method, linear systems, eigenvalues

AMS subject classifications. 65F10 41A21 65F25 13F07 93B20

1. Abstraction of the Euclidean algorithm. The Euclidean algorithm is described in the Elements of Euclid, book 7, theorems 1 and 2. The Elements date back to ca 300 BC, but traces of the Euclidean algorithm can be found in the work of other Greek mathematicians [5, p.6] and some even go back to the Babylonian period [1, p.27]. It is certainly one of the oldest nontrivial algorithms and it is still used in many modern applications.

The description of the algorithm lies in the proof of the two theorems mentioned before. They deal with, what we now call the greatest common divisor (GCD) of two positive integers. The second theorem considers the case of relative prime numbers with greatest common divisor equal to 1. This is treated separately because, according to the Greek, 1 is not considered as a number. The proof of these theorems has a geometrical flavor. Suppose we have two bars, call them L_{-1} and L_0 and let the length of L_{-1} be at least equal to the length of L_0. The problem is to find a bar of common "unit" length. That is a bar of maximal length such that the length of L_{-1} and L_0 is an integer multiple of the length of the unit bar. The problem was solved as follows: Place L_0 along L_{-1} and cut from L_{-1} a piece exactly as long as L_0. Call L_1 the remaining part of L_{-1}. Then we repeat this procedure, placing L_0 along L_1 and cutting off until L_0 is longer than what remains of L_{-1}. Let us call this remainder L_k (it is obtained in k steps). Then we reverse the role of L_k and L_0 and start cutting pieces from L_0 with the length of L_k. This procedure goes on and on, each time a bar becomes shorter than the other, the role of the two bars is interchanged. In the end, one will hopefully find a situation when, placing the measuring bar against the remainder of the longer one, it turns out that they have exactly the same length. This length is then the common unit bar we tried to find in the beginning.

Reformulating this procedure with positive integers, we get the following. Given two positive integers, subtrahend the smaller from the larger, giving again two numbers: the remainder of the minuend and the subtrahend. With these two numbers, repeat the procedure, until you get two equal numbers. This will then be the GCD of the two original numbers. Take for example $r_{-1} = 8$ and $r_0 = 6$. Subtract:

* Department of Computing Science, K.U.Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium.
This research was supported by the National Fund for Scientific Research (NFWO), project Lanczos, grant #2.0042.93

$8 - 6 = 2 < 6$, then subtract 2 from 6 and you get 4, subtract again and you get 2. The remainder and the subtrahend are both equal to 2 and thus is $GCD(8,6) = 2$. The action of subtracting several times the same number can be compressed in an integer division. For example the subtraction $8 - 6 = 2$ can be represented by the division

$$\frac{8}{6} = 1 + \frac{2}{6} \quad \Rightarrow \quad \text{quotient 1, remainder 2,}$$

meaning that you have to subtract 6 from 8 only once and the remainder is 2. The next step results in

$$\frac{6}{2} = 3 + \frac{0}{2} \quad \Rightarrow \quad \text{quotient 3, remainder 0.}$$

Thus you can subtract 2 from 6 exactly 3 times and the remainder is 0, which certifies that 2 is the GCD of 8 and 6. The most important numbers in this procedure are the remainders. Thus we use the modulo-notation to stress the remainder operation in the previous divisons as follows

$$8 \bmod 6 = 2 \quad \text{and} \quad 6 \bmod 2 = 0$$

Thus, the Euclidean algorithm can now be formulated as

$GCD(a,b)$
$r_{-1} = a, r_0 = b$
$k = 1$
while $r_{k-1} \neq 0$
  set $r_k = r_{k-2} \bmod r_{k-1}$
  $k = k + 1$
end while
$GCD(a,b) = r_{k-2}$

This formulation for positive integers needs only a reinterpretation of the notation to work also for polynomials. For example, to find the GCD of the polynomials

$$r_{-1} = z^4 - z^3 - z^2 + 3z - 2$$
$$r_0 = z^3 + z^2 - z - 1$$

we can again divide and retain the remainder, which will be indicated by the same notation:

$$r_1 = r_{-1} \bmod r_0 = 2z^2 + 2z + 4$$
$$r_2 = r_0 \bmod r_1 = z - 1$$
$$r_3 = r_1 \bmod r_2 = 0.$$

Hence $GCD(r_{-1}, r_0) = z - 1$. The algorithm can be used without any change, we only need to know the meaning of the mod-operation; and to be certain that the algorithm will end, we have to be sure that the situation $r_k = 0$ will eventually occur. In the polynomial case, this will certainly happen, because the degree of r_k will be strictly decreasing with k, and if we assume by convention that 0 has a negative degree, then

we shall always end up with this situation. For the positive integers, we have a similar situation. Just associate with the integer its value, then this will always decrease, down to zero and the algorithm will end.

The GCD is not unique though. For the polynomial case, we can always multiply the GCD with a nonzero constant. For integers, if we don't restrict ourselves to positive integers, we can multiply a GCD with plus or minus 1 and still get a GCD. Such an element that can be used freely as a factor is called a unit. A unit is a (nonzero) element whose inverse is again in the same set. For the integers, the only units are ± 1 and for the polynomials, the units are all nonzero constants. In the Euclidean algorithm, which is based on the chain of equalities $GCD(r_{-1}, r_0) = GCD(r_0, r_1) = GCD(r_1, r_2) = \cdots$, we can at every stage multiply each of the two elements $s_k = r_{k-1}$ and r_k with such a unit. Because $GCD(s_k, r_k) = GCD(s_k x_k, r_k c_k)$ when x_k and c_k are units, we can insert these units in every iteration step, to ensure that the dividend and divisor have a certain normalization, for example we make them positive for the integers or monic in the polynomial case. We could also use this freedom to normalize the remainders in the iterations. Indeed, $r_k = (s_{k-1}x_k) \bmod (r_{k-1}c_k) = x_k c_k^{-1}(s_{k-1} \bmod r_{k-1})$ so that $x_k c_k^{-1}$ can force a normalized remainder r_k. In this case the successive remainders r_k, and hence also the GCD, will be normalized. So, we can always pick a representant of the equivalence class of GCDs with a certain normalization and consider a GCD as uniquely defined, so that it is appropriate to speak about "the" GCD. Whatever the normalization for the GCD, we can compute it by the Euclidean algorithm.

We have now all the elements to give an abstract definition of the Euclidean algorithm in a Euclidean domain. The latter is an integral domain D with a valuation $\partial : D \to \mathbb{N}$ which satisfies $\partial(a) = 0$ iff $a = 0$, $\partial(ab) \geq \partial(a)$, for all $a, b \neq 0$ and there should be some division property or modulo calculus possible, i.e., for any two elements $a, b \in D$, $b \neq 0$, there exist elements q (quotient) and r (remainder) such that $a = bq + r$ with $\partial(r) < \partial(b)$. We denote r as $a \bmod b$ in this division property, then the Euclidean algorithm will still give you a GCD exactly as it did before. It will certainly end after a finite number of steps because the valuation $\partial(r_k)$ is a strictly decreasing sequence of integers that will therefore end up being zero, which corresponds to the zero element and then the algorithm stops.

Note that in general the quotient need not be uniquely defined. In $\mathbb{Z}$ for example, we may write $8 = 1 \cdot 6 + 2$ but also $8 = 2 \cdot 6 - 4$. With the valuation $\partial(a) = |a|$, both relations are possible since $|2| < 6$ as well as $|-4| < 6$. Both choices lead to a GCD that is in the same equivalence class of greatest common divisors in the end. By appropriate normalizations in the iterations, we may get a uniquely defined GCD.

2. The extended Euclidean algorithm and continued fractions. The successive divisions that are implicit in the Euclidean algorithm can be collected in a continued fraction expansion of the ratio of the two elements. For example, for the numbers 8 and 6, the successive steps are reflected in the continued fraction

$$\frac{8}{6} = 1 + \frac{2}{6} = 1 + \frac{1}{\frac{6}{2}} = 1 + \cfrac{1}{3 + \frac{0}{2}}$$

or for the polynomial example, this becomes

$$\frac{r_{-1}}{r_0} = z - 2 + \frac{r_1}{r_0} = z - 2 + \cfrac{1}{\frac{r_0}{r_1}}$$

$$= z - 2 + \cfrac{1}{\cfrac{1}{2}z + \cfrac{r_2}{r_1}} = z - 2 + \cfrac{1}{\cfrac{1}{2}z + \cfrac{1}{\cfrac{r_1}{r_2}}}$$

$$= z - 2 + \cfrac{1}{\cfrac{1}{2}z + \cfrac{1}{2z + 4 + \cfrac{0}{z-1}}}.$$

Each step of this expansion makes use of

$$\frac{s_{k-1}}{r_{k-1}} = q_k + \frac{r_k}{s_k}$$

where we have set $s_k = r_{k-1}$, or

$$[s_k \;\; r_k] = [s_{k-1} \;\; r_{k-1}]V_k; \quad V_k = \begin{bmatrix} 0 & 1 \\ 1 & -q_k \end{bmatrix}.$$

If we introduce the freedom we have to choose some units x_k and c_k as factors in each step, we can replace V_k by

$$V_k = \begin{bmatrix} 0 & 1 \\ 1 & -q_k \end{bmatrix} \begin{bmatrix} x_k & 0 \\ 0 & c_k \end{bmatrix} \equiv \begin{bmatrix} y_k & c_k \\ x_k & a_k \end{bmatrix}$$

with

$$a_k = -q_k c_k; \quad y_k = 0; \quad q_k = \text{quotient of } s_{k-1}/r_{k-1}.$$

The units x_k and c_k are used to get proper normalizations for s_k and r_k. If we start with $s_0 = s$ and $r_0 = r$, then it may be that these do not have the same normalization as all the other s_k, r_k, $k = 1, 2, \ldots$. To ensure that also s_0 and r_0 have the same normalization, we introduce an extra V_0 which will be of the from

$$V_0 = \begin{bmatrix} y_0 & c_0 \\ x_0 & a_0 \end{bmatrix} \quad \text{with } x_0 = c_0 = 0 \text{ and } y_0 \text{ and } a_0 \text{ units.}$$

The given elements s and r are then first normalized in step 0, to give $s_0 = sy_0$ and $r_0 = ra_0$.
The original continued fraction expansion which had the form

$$\frac{r_{-1}}{r_0} = q_1 + \left\lceil \frac{1}{q_2} \right\rfloor + \cdots + \left\lceil \frac{1}{q_k} \right\rfloor + \left\lceil \frac{r_k}{r_{k-1}} \right\rfloor$$

could now be replaced by the equivalent one

$$-\frac{r}{s} = \frac{y_0}{a_0}\left(\left\lceil \frac{c_1}{a_1} \right\rfloor + x_1 \left\lceil \frac{c_2}{a_2} \right\rfloor + \cdots + x_{k-1} \left\lceil \frac{c_k}{a_k} \right\rfloor + x_k \left\lceil \frac{r_k}{s_k} \right\rfloor \right).$$

Cutting the continued fraction after a certain number of steps gives approximants of the ratio we are expanding. For example,

$$-\frac{r}{s} \approx \frac{c_{0k}}{a_{0k}} = \frac{y_0}{a_0}\left(\left\lceil \frac{c_1}{a_1} \right\rfloor + x_1 \left\lceil \frac{c_2}{a_2} \right\rfloor + \cdots + x_{k-1} \left\lceil \frac{c_k}{a_k} \right\rfloor \right), \; k \geq 1 \text{ and } \frac{c_{00}}{a_{00}} = \frac{0}{a_0}$$

where the numerator and denominator satisfy the recurrence

$$(1)\qquad \begin{bmatrix} y_{0k} & c_{0k} \\ x_{0k} & a_{0k} \end{bmatrix} = \begin{bmatrix} y_{0,k-1} & c_{0,k-1} \\ x_{0,k-1} & a_{0,k-1} \end{bmatrix} V_k; \quad \begin{bmatrix} x_{0,-1} & c_{0,-1} \\ y_{0,-1} & a_{0,-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The extended Euclidean algorithm keeps track of these elements too. Thus, it can be condensed in the formula

$$G_k = G_{-1}V_0V_1\dots V_k = \begin{bmatrix} y_{0k} & c_{0k} \\ x_{0k} & a_{0k} \\ s_k & r_k \end{bmatrix}; \quad G_{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ s & r \end{bmatrix},$$

with V_k, $k \geq 0$ as before. Note also that when $r_k = 0$, then it follows from

$$[s \;\; r]V_{0k} = [s_k \;\; r_k] \quad \text{with} \quad V_{0k} = V_0V_1\dots V_k$$

that

$$\begin{cases} sc_{0k} + ra_{0k} = 0 \\ sx_{0k} + ry_{0k} = s_k. \end{cases}$$

Thus we get the well known property saying that $GCD(s,r) = s_k$ can be written as a combination of s and r. The factors we need in this combination are precisely given by x_{0k} and y_{0k}.

3. Approximation and formal series. As far as approximation is concerned, it does not matter whether the Euclidean algorithm terminates or not. If we started from a real nonrational number, then we know that its continued fraction expansion would never end. For example

$$\begin{aligned} 3.1415926\dots &= 3 + \frac{1}{(0.1415926\dots)^{-1}} = 3 + \frac{1}{7.0625133\dots} \\ &= 3 + \frac{1}{\lfloor 7 \rfloor} + \frac{1}{\lfloor (0.0625133\dots)^{-1} \rfloor} = 3 + \frac{1}{\lfloor 7 \rfloor} + \frac{1}{\lfloor 15.9965\dots \rfloor} \\ &= \dots = 3 + \frac{1}{\lfloor 7 \rfloor} + \frac{1}{\lfloor 15 \rfloor} + \frac{1}{\lfloor 1 \rfloor} + \frac{1}{\lfloor 292.63459\dots \rfloor} = \dots \end{aligned}$$

The same is true when we work with (formal) series instead of ratios of polynomials. For example, consider the formal Laurent series $\pi(z) \in \mathbb{F}\langle z^{-1} \rangle$ over some field $\mathbb{F}$ with finitely many terms with positive exponents, say

$$\pi(z) = 3z^0 + 1z^{-1} + 4z^{-2} + 1z^{-3} + \dots \in \mathbb{F}\langle z \rangle,$$

then we will get a never ending continued fraction

$$\pi(z) = 3 + \cfrac{1}{z - 4 + \cfrac{1}{\cfrac{z^{-1} + 4z^{-2} + z^{-3} + 5z^{-4} + \dots}{15z^{-2} - z^{-3} + 11z^{-4} + 9z^{-5} + \dots}}}$$

We shall get approximants of the given quantity $\pi(z)$ when the continued fraction is truncated. Except for the lack of the termination property, the Euclidean algorithm

is still applicable with a minor redefinition of what we mean now by quotient and remainder. For the case of a real number, the quotient is the integer part and the remainder is the decimal part; for the Laurent series, the quotient is the polynomial part of the series and the remainder is the part with the negative exponents for z.

A similar algorithm can be formulated when z is replaced by z^{-1}. We then get approximants for formal Laurent series (or power series) from $\mathbb{F}\langle z\rangle$ having only a finite number of terms with a negative exponent. In fact, the variant of the Euclidean algorithm, adapted to the latter situation is known as Arne Magnus' P-fraction algorithm. The P- stands for principal part, since the quotient corresponds here to the principal part (including the constant term) of the series $\mathbb{F}\langle z\rangle$. The approximant one gets then is of Padé type. This can be seen as follows. Suppose that $r \in \mathbb{F}\langle z^{-1}\rangle$ is strictly proper, i.e., it has only terms with negative exponents. By choosing $s = -1$ we get that $f = -r/s = r$. Truncating the previous continued fraction expansion for f after k terms, gives an approximant c_{0k}/a_{0k} satisfying the relation

$$f = -\frac{r}{s} = \frac{c_{0k}}{a_{0k}} - \frac{r_k}{s a_{0k}} = \frac{c_{0k}}{a_{0k}} + \frac{r_k}{a_{0k}} \tag{2}$$

where r_k/a_{0k} is the approximation error. Recall that a_k is up to a unit (which is here a nonzero element from the field $\mathbb{F}$) equal to the quotient. Suppose that the quotient, and therefore also a_k, is a polynomial of degree α_k, then it follows from the recurrence (1) that a_{0k} will be a polynomial of degree $\alpha_{0k} = \alpha_1 + \cdots + \alpha_k$, $(\alpha_0 = 0)$, and the "degree" of the formal series $f - c_{0k}/a_{0k} = r_k/a_{0k}$ will be $-2\alpha_{0k} - \alpha_{k+1}$, $\alpha_{k+1} \geq 1$. By the degree of a formal series $\in \mathbb{F}\langle z^{-1}\rangle$, we mean the largest exponent of the terms with nonzero coefficient. Because also the degree of c_{0k} does not exceed the degree of a_{0k}, we see that c_{0k}/a_{0k} is a genuine Padé approximant (at ∞) for the given ratio $f = -r/s$. The definition of Padé approximant is given below. Moreover, it can be shown that among all strictly proper rational approximants for f at ∞, which approximate f up to a degree at least $-2\alpha_{0k} - 1$, the approximant c_{0k}/a_{0k} will have minimal degree. We have solved here a minimal partial realization problem for the impulse response $f \in \mathbb{F}\langle z^{-1}\rangle$. The definition of this standard problem in linear system theory is also given below. In the latter context, the Euclidean algorithm is often known as the Berlekamp-Massey algorithm. This algorithm was originally designed for the realization of linear predictive codes [3, Chap.9],[26].

When we replace z by z^{-1}, and divide out one power of z, the strictly proper Laurent series in $\mathbb{F}\langle z^{-1}\rangle$ transforms into a power series in $\mathbb{F}[[z]]$. By the transformation $z \to z^{-1}$, the Euclidean algorithm will produce an approximant $c_{0k}(z^{-1})/a_{0k}(z^{-1})$ where numerator and denominator are now polynomials in z^{-1} with $\deg c_{0k}(z) < \deg a_{0k}(z) = \alpha_{0k}$. After dividing out one power of z, we get an approximant of the form $z^{-1}c_{0k}(z^{-1})/a_{0k}(z^{-1})$ which is equivalent with a rational function c^*_{0k}/a^*_{0k} with $c^*_{0k}(z) = z^{-1}(z^{\alpha_{0k}}c^*_{0k}(z^{-1}))$ and $a^*_{0k}(z) = z^{\alpha_{0k}}a^*_{0k}(z^{-1})$ so that the degree of c^*_{0k} is at most $\alpha_{0k} - 1$ and the degree of a^*_{0k} is at most α_{0k}, while the order of approximation in (2) is $\operatorname{ord}(r_k/a_{0k}) = 2\alpha_{0k} + \alpha_{k+1} - 1$, $\alpha_{k+1} \geq 1$. This kind of approximant is called a Padé approximant of f (around zero).

An (m, n) Padé approximant is usually defined for a power series $f(z)$ at $z = 0$. First define a nontrivial rational form $P(z)/Q(z)$ with $\deg P \leq m$, $\deg Q \leq n$ and such that the order of the series $fQ - P$ is at least $m+n+1$. The Padé (m, n) approximant is then the rational function defined by $R_{mn}(z) = P(z)/Q(z)$. These approximants are arranged in a table, called Padé table. The m-axis goes vertically down and the

n-axis goes horizontally to the right and the (m,n) entry contains an (m,n) Padé approximant for the given series. This Padé table has a particular block structure [15]. When the series f starts with zero coefficients: $f(z) = c_p z^p + c_{p+1} z^{p+1} + \cdots$, then the Padé table has a band of entries (m,n), $m = 0,\ldots,p-1$, $n = 0,1,2,\ldots$ that are all equal to zero. The rest of the table will in general contain square blocks where all the entries can be reduced to the left top element of that block. A Padé table where all these blocks are 1×1, and hence, where all the entries are different, is called a normal Padé table, otherwise, it is called singular.

As it was explained before, the Euclidean algorithm will generate the entries of a Padé table for f located on a diagonal $(n-1,n)$, $n = 1,2,\ldots$. When in the Euclidean algorithm all the a_k are of degree 1, then the diagonal path followed by the algorithm will not hit a singular block. If some a_k has a higher degree, this will correspond to a jump made by the algorithm to cross a singular block that is met on the diagonal.

4. Linear algebra and block orthogonal polynomials. The equations defining a Padé approximant at ∞ for a formal Laurent series $f(z) = \sum_{k=1}^{\infty} f_k z^{-k} \in \mathbb{F}\langle z^{-1}\rangle$, can be written in the form of an infinite set of linear equations. Define the Hankel matrix with symbol f: $H = H(f)$ as $H = [f_{i+j+1}]$, $i,j = 0,1,\ldots$, and let $\mathbf{a}_{0k}$ be the (infinite) column, stacking the $\alpha_{0k}+1$ coefficients of the polynomial $a_{0k}(z)$, extended with infinitely many zeros, thus $a_{0k}(z) = [1 \;\; z \;\; z^2 \;\; \ldots]\mathbf{a}_{0k}$. Similarly, we shall use the bold font for stacking vectors of series, e.g., $r_k(z) = [z^{-1} \;\; z^{-2} \;\; \ldots]\mathbf{r}_k$ etc. Then the relation

$$f(z)a_{0k}(z) = c_{0k}(z) + r_k(z)$$

truncated to the negative powers of z can be written as

$$H\mathbf{a}_{0k} = \mathbf{r}_k.$$

Clearly, when Z denotes a down shift operator: $Z = [\delta_{i,j+1}]$, then $Z^T H = HZ$ is a characteristic property of a Hankel operator and it thus holds that

$$H[Z\mathbf{a}_{0k}] = Z^T\mathbf{r}_k$$

and more generally

$$H[Z^i\mathbf{a}_{0k}] = [Z^T]^i\mathbf{r}_k, \quad i = 0,1,\ldots,\alpha_{k+1}-1.$$

We can arrange all these relations for $k = 0,1,\ldots$ and $i = 0,1,\ldots,\alpha_{k+1}-1$ as follows

$$H[A_{\cdot 0}|A_{\cdot 1}|\ldots] = [R_{\cdot 0}|R_{\cdot 1}|\ldots]$$

where

$$A_{\cdot k} = [\mathbf{a}_{0k}|Z\mathbf{a}_{0k}|\ldots|Z^{\alpha_{k+1}-1}\mathbf{a}_{0k}]$$

and

$$R_{\cdot k} = [\mathbf{r}_{0k}|Z^T\mathbf{r}_{0k}|\ldots|(Z^T)^{\alpha_{k+1}-1}\mathbf{r}_{0k}]$$

The matrix $A = [A_{\cdot 0}|A_{\cdot 1}|\ldots]$ is upper triangular with each block $A_{\cdot k}$ being a Toeplitz stripe, while, $R = [R_{\cdot 0}|R_{\cdot 1}|\ldots]$ is block lower triangular and Hankel striped. This means that by symmetry arguments, $A^T R = A^T H A = R^T A$ is block diagonal,

$$A^T H A = \operatorname{diag}(D_{00}, D_{11}, \ldots) = D \tag{3}$$

where one can easily check that each diagonal block D_{kk} has a lower triangular Hankel structure (with zeros above the main antidiagonal, and a nonzero element on that diagonal). Thus $H = A^{-T}DA^{-1}$ and $H^{-1} = A^T D^{-1} A$ are symmetric block LDR factorizations of the Hankel matrix and its inverse. The size of the k-th block is α_{k+1}. We shall call these indices the Euclidean indices. Their sums $\alpha_{0k} = \alpha_1 + \cdots + \alpha_k$ are generally known as the Kronecker indices. It is also possible to write the recurrence relation (1) in terms of matrices. We have from (1)

$$a_{0k}(z) = a_{0,k-2}(z)x_{k-1}c_k + a_{0,k-1}(z)a_k(z) \tag{4}$$

where x_{k-1} and c_k are units and $a_k(z) = -q_k(z)c_k$ with $q_k(z)$ the "quotient" of $s_{k-1}(z)$ divided by $r_{k-1}(z)$. Within a block $A_{.k}$, the columns are shifted one position down each time you move one place to the right. The transition from one block to the next one is given by the previous recurrence. This leads to the following relation

$$ZA_{.,k-1} = A[0 \ldots 0 | T^T_{k-2,k-1} | T^T_{k-1,k-1} | T^T_{k,k-1} | 0 \ldots]^T$$

where $T_{k-2,k-1} \in \mathbb{F}^{\alpha_{k-1} \times \alpha_k}$, $T_{k-1,k-1} \in \mathbb{F}^{\alpha_k \times \alpha_k}$, $T_{k,k-1} \in \mathbb{F}^{\alpha_{k+1} \times \alpha_k}$ have the forms

$$T_{k-2,k-1} = \begin{bmatrix} 0 & \ldots & 0 & -\beta_k \\ 0 & \ldots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \end{bmatrix}, \quad \beta_k = x_{k-1}c_k,$$

$$T_{k-1,k-1} = F(a_k) = \begin{bmatrix} 0 & \ldots & 0 & -a_0^{(k)} \\ 1 & \ldots & 0 & -a_1^{(k)} \\ & \ddots & \vdots & \vdots \\ & & 1 & -a_{\alpha_k-1}^{(k)} \end{bmatrix}, \quad a_k(z) = \sum_{i=0}^{\alpha_k} a_i^{(k)} z^i, \quad a_{\alpha_k}^{(k)} = 1$$

$$T_{k,k-1} = \begin{bmatrix} 0 & \ldots & 0 & 1 \\ 0 & \ldots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \end{bmatrix}.$$

Note that $T_{k-1,k-1} = F(a_k)$ is the Frobenius or companion matrix associated with the polynomial $a_k(z)$, which is supposed to be monically normalized by an appropriate choice of the units. Collecting all these relations for $k = 0, 1, \ldots$, we get the following relation

$$ZA = AT$$

where T is a block tridiagonal matrix:

$$T = \text{tridiag} \begin{pmatrix} T_{01}, & T_{12}, & \ldots \\ T_{00}, & T_{11}, & \ldots \\ T_{10}, & T_{21}, & \ldots \end{pmatrix}.$$

These matrix relations were obtained e.g., in [16] where you can find more details. Now it is possible to make another connection. If the Padé table of $f(z)$ is normal, then all the blocks are of size $\alpha_k = 1$ and the relation

$$A^T H A = D$$

expresses the orthogonality of the polynomials $a_{0k}(z)$ with respect to the inner product whose Gram matrix is given by

$$H = \left[\left\langle z^k, z^l \right\rangle\right] = [f_{k+l+1}]$$

and the diagonal elements in D are the squared norms of the polynomials

$$\langle a_{0k}(z), a_{0l}(z)\rangle = \delta_{kl} D_{kk}.$$

The relation $ZA = AT$ can be interpreted as the well known three term recurrence ralation with T being the Jacobi matrix which contains the recurrence coefficients. It is the matrix expression for the relation (1).

Note that the inner product is not necessarily positive definite, since the Gram matrix H is not necessarily positive definite. It is in the normal case non degenerate, meaning that none of the polynomials $a_{0k}(z)$ is orthogonal to itself, because all the D_{kk} are nonzero. This corresponds to the Gram matrix being strongly nonsingular, i.e., all its leading submatrices are regular.

The general situation where the entries of the factorization are blocks of size larger than or equal to 1 corresponds to a generalization. The polynomials are now organized in blocks. Each polynomial in a block

$$A_{\cdot k}(z) = [a_{0k}(z), z a_{0k}(z), \ldots, z^{\alpha_{k+1}-1} a_{0k}(z)]$$

is orthogonal to all the polynomials in previous blocks but not necessarily to the polynomials within the same block. We shall call these block orthogonal polynomials. The size of the blocks are dictated by the values of the Euclidean indices α_{k+1}. The Kronecker indices α_{0k} define precisely the size α_{0k} of the nonsingular leading submatrices in the Gram matrix H. The leading submatrices in between are all singular.

As a conclusion we could say that the Euclidean algorithm provides a fast method for performing a Gram-Schmidt orthogonalization in an inner product space with a metric that can be indefinite and/or degenerate. It is essential however that the Gram matrix is Hankel, i.e., that the inner product satisfies $\left\langle z^k, z^l \right\rangle = \left\langle z^{k+l}, 1 \right\rangle$. This is usually expressed by calling them orthogonal with respect to a linear functional defined by $\mathcal{L}\{z^k\} = f_{k+1}$. Such formal orthogonal polynomials were studied by Brezinski [4] and Draux [9].

When we cut the infinite matrices to finite sections, the previous relations transform into

$$F(a_{0,k+1}) A_k = A_k T_k \tag{5}$$

with $F(a_{0,k+1})$ the Frobenius matrix of $a_{0,k+1}(z)$, which is the restriction of the shift operator to the finite subspace of dimension $\alpha_{0,k+1}$, while A_k and T_k are the leading submatrices of size $\alpha_{0,k+1}$ of the matrices A and T respectively.

Before concluding this section, we introduce one more matrix relation which will be important in the sequel of this paper. We have seen that the denominators of Padé approximants may be considered as formal block orthogonal polynomials. The recurrence relation computes these polynomials for a diagonal in the Padé table. One may consider other, adjacent diagonals. These give rise to a related sequence of block orthogonal polynomials and it will be no surprise that these adjacent sequences are related in one way or another. In terms of linear algebra, one then considers a shifted Hankel matrix as Gram matrix, i.e., the matrix with symbol $zf(z)$, instead of $f(z)$. Using $H(zf) = Z^T H(f) = H(f)Z$ and $A^T H(f)A = D$ and $ZA = AT$, we get

$$A^T H(zf)A = DT = J, \tag{6}$$

where the right hand side $J = DT$ is block tridiagonal because T is block tridiagonal and D is block diagonal. It turns out that J has a special Hankel block structure [16]. For example with $k = 2$, $\alpha_1 = 2$, $\alpha_2 = 3$ and $\alpha_3 = 4$, we get for a finite section

$$J_2 = D_2 T_2 = \left[\begin{array}{cc|ccc|cccc} \times & \times & & & & & & & \\ \times & \times & & & \times & & & & \\ \hline & & 0 & \times & \times & & & & \\ & & \times & \times & \times & & & & \\ & \times & \times & \times & \times & & & & \times \\ \hline & & & & & 0 & 0 & \times & \times \\ & & & & & 0 & \times & \times & \times \\ & & & & & \times & \times & \times & \times \\ & & & & \times & \times & \times & \times & \times \end{array}\right].$$

The nonzero elements are indicated by $\times$ and they form lower triangular Hankel matrices.

5. Linear system theory. Consider a discrete, linear, time invariant, strictly causal system which is described by the state space equations

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, & A &\in \mathbb{F}^{n\times n}, B \in \mathbb{F}^{n\times 1} \\ y_k &= C^T x_k & C &\in \mathbb{F}^{n\times 1} \end{aligned}$$

$x_k \in \mathbb{F}^{n\times 1}$ is the state, $u_k \in \mathbb{F}$ is the input and $y_k \in \mathbb{F}$ is the output at time instance k. The matrix A is called the state transition matrix and n is the dimension of the state space which we suppose to be finite. The system is discrete because the time index is discrete. It is linear because the relation between input and output is linear. It is time invariant because A, B and C do not depend on the time index and it is strictly causal because an input at time k will only influence the output at time $k+1$ and later. The triple (A, B, C) is called a state space realization of the system. A state space triple is not unique. For any nonsingular matrix F, the triple $(F^{-1}AF, F^{-1}B, F^T C)$ will generate the same input-output behavior. This is easily checked using the expression for the transfer function which is given below. Such triples are called equivalent. The state space is the space generated by the state vectors x_k. When the dimension of the state space is less than n, it is possible to describe the same input-output relation with state vectors, and hence also A, of smaller size; hence to produce a realization with a smaller n. The realization is called minimal if n is the smallest possible one.

Taking z-transforms of all the sequences involved, e.g., $\hat{u}(z) = \sum_k u_k z^{-k}$ etc., we may write

$$z\hat{x}(z) = A\hat{x}(z) + B\hat{u}(z)$$
$$\hat{y}(z) = C^T\hat{x}(z)$$

thus we find for the input-output relation

$$\hat{y}(z) = C^T(zI - A)^{-1}B\hat{u}(z) = f(z)\hat{u}(z).$$

The rational function $f(z) = C^T(zI - A)^{-1}B$ is called the transfer function of the system. Suppose it has the (formal) expansion

$$f(z) = \sum_{k=1}^{\infty} f_k z^{-k}$$

then the sequence (f_k) is called the impulse response and the $f_k = C^T A^{k-1}B$ are called Markov parameters. All you have to know about linear systems and much more can be found in Kailath's book [23].

Note that $f(z)$ is a strictly proper rational function of degree at most n. If n is minimal, then f has strict degree n and this minimal n is called the McMillan degree of the system. It equals the degree of the denominator of a coprime representation of the transfer function. The matrix $\mathcal{O} = [C\ A^TC\ (A^T)^2C\ \ldots]^T$ is called the observability matrix and $\mathcal{R} = [B\ AB\ A^2B\ \ldots]$ the reachability matrix. The realization is minimal iff $n = \operatorname{rank}\mathcal{R} = \operatorname{rank}\mathcal{O} = \operatorname{rank}\mathcal{O}\mathcal{R}$. The realization is said to be reachable if $\mathcal{R}$ has full rank and observable if $\mathcal{O}$ has full rank. Thus a minimal realization is both reachable and observable. Note that the matrix

$$H = \mathcal{O}\mathcal{R} = \left[C^TA^{i+j}B\right] = [f_{i+j+1}]$$

is Hankel.

The (minimal) partial realization problem is to find a (minimal) linear system realization whose impulse response has Markov parameters matching the initial Markov parameters of the given system. Thus in terms of the transfer function, we have to find a system with (rational) transfer function of minimal degree which will be a Padé approximant at ∞ of the given transfer function. As we have seen before, this problem can be solved essentially by the Euclidean algorithm. For example, if $c_{0k}(z)$ and $a_{0k}(z)$ are related to $f(z)$ by the Euclidean algorithm, then $c_{0k}(z)a_{0k}(z)^{-1}$ will be a minimal partial realization of McMillan degree α_{0k} and the degree of approximation is $-2\alpha_{0k} - \alpha_{k+1}$, meaning that $2\alpha_{0k} + \alpha_{k+1} - 1$ of the initial Markov parameters are fitted.

6. Iterative solution of linear systems. Suppose we want to solve a linear system $Mx = b$ with a positive definite symmetric matrix M of size m. We could do this by minimizing $h(x) = \frac{1}{2}\|r\|_{M^{-1}} = \frac{1}{2}r^TM^{-1}r$ with $r = b - Mx$. At every iteration step, we choose a search direction s_{k-1} and take a step λ_k in this direction to find an updated value

$$x_k = x_{k-1} + \lambda_k s_{k-1}. \tag{7}$$

An optimal value along this direction is obtained by solving $\frac{d}{d\lambda_k}h(x_k) = 0$, which results in $r_k^T s_{k-1} = 0$. The residual r_k in x_k is given by

$$r_k = b - Mx_k = r_{k-1} - \lambda_k M s_{k-1}. \tag{8}$$

It is equal to the negative gradient of $h(x)$ in $x = x_k$. Hence

$$\lambda_k = \frac{r_{k-1}^T s_{k-1}}{s_{k-1}^T M s_{k-1}}.$$

A possible choice for s_{k-1} is to take the steepest descent direction for $h(x_{k-1})$, which is $s_{k-1} = r_{k-1}$. This results in a method suffering from the well known zig-zagging effect, meaning that, e.g., for $m = 2$, the x_k move towards the minimum alternating between two directions intersecting in the minimum, so that the latter will only be reached after an infinite number of steps. It is better to use independent search directions orthogonal with respect to M, i.e., conjugate directions satisfying $s_k^T M s_{k-1} = 0$. This guarantees in principle a solution after at most m steps. A choice for these conjugate directions which is optimal in the sense that they stay as close as possible to the negative gradients, i.e., the residuals r_k, is to choose them such that

$$s_k \in \operatorname{span}\{r_0, Mr_0, \ldots, M^k r_0\} \setminus \operatorname{span}\{r_0, Mr_0, \ldots, M^{k-1} r_0\}.$$

This is obtained by choosing

$$s_k = r_k + \mu_k s_{k-1}, \quad \mu_k = -\frac{r_k^T M s_{k-1}}{s_{k-1}^T M s_{k-1}} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

with initial conditions $s_0 = r_0 = b - Mx_0$ [14, 22]. By induction, it is easy to prove that

$$s_k, r_k \in \operatorname{span}\{r_0, Mr_0, \ldots, M^k r_0\}$$

while

$$r_k^T r_l = 0, \quad k \neq l. \tag{9}$$

This is called the conjugate gradient (CG) method which appears in the original paper of Hestenes and Stiefel (1952) [21]. A survey of the history of this method and its relation to the Lanczos method can be found in [13]. The space

$$\mathcal{K}_k(r_0, M) = \operatorname{span}\{r_0, Mr_0, \ldots, M^k r_0\}$$

is called a Krylov subspace generated by r_0 and M. From (9), it follows that the CG algorithm generates residuals which form an orthogonal basis for $\mathcal{K}_k(r_0, M)$. Because $r_k \in \mathcal{K}_k(r_0, M)$, there must be a polynomial $\varphi_k(z)$ of degree k such that $r_k = \varphi_k(M)r_0$. It turns out that φ_k is comonic, i.e., $\varphi_k(0) = 1$. We can now define an inner product for the space of polynomials of degree at most m by defining the Gram matrix for the natural basis functions:

$$H_n = H_n(f) = [f_{i+j+1}]_{i,j=0,\ldots,n}; \quad \left\langle z^i, z^j \right\rangle_H = f_{i+j+1} = r_0^T M^i M^j r_0.$$

This is a Hankel matrix of size $(n+1)$. Suppose that the polynomials $\varphi_k(z)$, which we introduced above, can be written as $\varphi_k(z) = \sum_{i=0}^{k} a_i^{(k)} z^i$, then for $0 \leq k, l \leq n < m$:

$$\langle \varphi_k, \varphi_l \rangle_H = r_k^T r_l = [a_0^{(k)}, \ldots, a_k^{(k)}, 0 \ldots, 0] H_n [a_0^{(l)}, \ldots, a_l^{(l)}, 0 \ldots, 0]^T$$

When we write

$$Y_n = [y_0 | y_1 | \ldots | y_n], \quad y_0 = r_0, \quad y_k = M y_{k-1}, \quad k = 1, \ldots, n,$$

then

$$H_n = H_n(f) = Y_n^T Y_n.$$

A direct consequence is that

$$\begin{aligned} \|r_n\|^2 = r_n^T r_n &= r_0^T \varphi_n(M)^T \varphi_n(M) r_0 \\ &= [a_0^{(n)}, \ldots, a_n^{(n)}] H_n [a_0^{(n)}, \ldots, a_n^{(n)}]^T \\ &= \|\varphi_n\|_H^2 \end{aligned}$$

with $\|\varphi_n\|_H^2 = \langle \varphi_n, \varphi_n \rangle_H$ where the inner product is defined above. In other words, generating residuals that form an orthogonal basis for $\mathcal{K}_n(r_0, M)$ is equivalent with generating orthogonal comonic polynomials with respect to the inner product $\langle \cdot, \cdot \rangle_H$. It is an undergraduate exercise to find out that the n-th comonic orthogonal polynomial in a Hilbert space is the one that solves the minimization problem

$$\min\{\|\varphi(z)\|_H : \varphi \in \Pi_n \setminus \Pi_{n-1}; \varphi(0) = 1\}.$$

Thus equivalently, $r_n = \varphi_n(M) r_0$ will solve the problem

$$\min\{\|r\| : r \in r_0 + \mathcal{K}_{n-1}(y_1, M), y_1 = M r_0\}$$

and this minimal residual (least squares) problem is solved when we attempt to solve our original optimization problem

$$\min\{\|b - Mx\|_{M^{-1}} : x \in x_0 + \mathcal{K}_{n-1}(r_0, M)\}.$$

All these problems turn out to give the same (approximate) solution for the system $Mx = b$. Since we already know how to generate polynomials orthogonal with respect to a Hankel Gram matrix, which is in this case $H_n(f) = Y_n^T Y_n$, this is where all this hooks up with the pevious theory. We can indeed take $\varphi_n(z)$ proportional to $a_{0n}(z)$, where $a_{0n}(z)$ are as generated by the Euclidean algorithm for f. Since $\mathcal{K}_n(r_0, M)$ is the column space of Y_n, to find an orthogonal basis for it, we have to find $\hat{Y}_n = Y_n A_n$ with $\hat{Y}_n$ having mutually orthogonal columns and A_n upper triangular. This orthogonality means that we have

$$\hat{Y}_n^T \hat{Y}_n = A_n^T Y_n^T Y_n A_n = A_n^T H_n(f) A_n = D_n$$

with D_n diagonal. The reader will recognize this as a special case of the relation (3) for the normal situation and for a finite section. It is then also clear that we get as well (see (6)),

$$\hat{Y}_n M \hat{Y}_n = A_n^T H_n(zf) A_n = D_n T_n = J_n$$

with J_n tridiagonal. Moreover, see (5),

$$F(a_{0,n+1})A_n = A_n T_n$$

gives a simple three-term recurrence relation for the columns of A_n, which are the stacking vectors for the polynomials φ_k (up to a normalization).

But, what happens if M is not symmetric and/or not positive definite? It is clear that then $H_n = Y_n^T Y_n$ is not Hankel anymore because

$$y_i^T y_j = r_0^T (M^T)^i M^j r_0.$$

One method to solve this problem is to orthogonalize the matrix Y_n as $\hat{Y}_n = Y_n A_n$ with the columns of $\hat{Y}_n$ orthogonal and A_n upper triangular, thus

$$A_n^T H_n A_n = D_n$$

with D_n diagonal. However, the Gram-Schmidt procedure will find the k-th column of A_n as a linear combination of $Z\mathbf{a}_{0,k-1}$ and $\mathbf{a}_{0,k-1}, \mathbf{a}_{0,k-2}, \ldots, \mathbf{a}_{00}$, which leads to a relation of the form

$$F(a_{0,n+1})A_n = A_n J_n$$

with J_n an upper Hessenberg matrix. Thus we shall not find a short recurrence for the underlying orthogonal polynomials $a_{0k}(z) = [1, z, \ldots, z^k, 0, \ldots, 0]\mathbf{a}_{0k}$. This method is known as Arnoldi's method. The procedure GMRES of Saad and Schultz (1986) [30] is probably the best known exponent of this family.

We can however produce a Hankel matrix if we are prepared to generate two Krylov subspaces

$$\mathcal{K}_n(y_0, M) \quad \text{and} \quad \mathcal{K}_n(z_0, M^T)$$

with $y_0 = z_0 = r_0$. Setting

$$Y_n = [y_0|y_1|\ldots|y_n] \quad \text{and} \quad Z_n = [z_0|z_1|\ldots|z_n]$$

with $y_k = My_{k-1}$ and $z_k = M^T z_{k-1}$, $k = 1, 2, \ldots$, we get

$$Z_n^T Y_n = H_n(f)$$

where $H_n(f)$ is the Hankel matrix with entries

$$f_{i+j+1} = z_i^T y_j = r_0^T M^i M^j r_0.$$

Note that from our system theoretic interpretation $f(z) = z_0^T (zI - M)^{-1} y_0$ so that (M, y_0, z_0) is a state space realization of the transfer function $f(z)$. The matrix Z_∞^T is the observability matrix and Y_∞ is the reachability matrix for the system.

The Gram-Schmidt orthogonalization has to be adapted to an asymmetric, i.e., two-sided version to bi-orthogonalize both the matrices Z_n and Y_n, to generate $\hat{Z}_n$ and $\hat{Y}_n$ such that $\hat{Z}_n^T \hat{Y}_n = D_n$ with D_n a (block) diagonal matrix. We know from the previous sections that we can find a (block) upper triangular matrix A_n such that

$$A_n^T H_n(f) A_n = D_n.$$

Hence, the matrices $\hat{Y}_n$ and $\hat{Z}_n$ can be found from $\hat{Y}_n = Y_n A_n$ and $\hat{Z}_n = Z_n A_n$, because then

$$\hat{Z}_n^T \hat{Y}_n = A_n^T Z_n^T Y_n A_n = A_n^T H_n(f) A_n = D_n.$$

The advantage of this method is that we still have

$$\hat{Z}_n^T M \hat{Y}_n = A_n^T H_n(zf) A_n = J_n = D_n T_n$$

with T_n and J_n (block) tridiagonal. Thus the columns of A_n, and therefore also of $\hat{Y}_n$ and $\hat{Z}_n$ can be obtained by a short recurrence, meaning that we have a fast algorithm. With the notation introduced before (see (4)), we get for example

$$\hat{y}_{0k} = a_{0k}(M) y_0 = a_k(M) \hat{y}_{0,k-1} + \beta_k \hat{y}_{0,k-2}$$

while

$$\hat{z}_{0k} = a_{0k}(M^T) z_0 = a_k(M^T) \hat{z}_{0,k-1} + \beta_k \hat{z}_{0,k-2}$$

where $\beta_k = x_{k-1} c_k$. The vector $\hat{y}_{0k}$ (similar considerations hold for $\hat{z}_{0k}$) is the first column of the k-th block of $\hat{Y}_n$. It is column $\hat{y}_{\alpha_{0k}}$ with $\alpha_{0k} = \deg a_{0k}$ the k-th Kronecker index for $H_n(f)$.

The two-sided Gram-Schmidt procedure for two Krylov subspaces, which is basically provided by the Euclidean algorithm, is the essence of the Krylov subspace methods of Lanczos type. The first one in this family was proposed by Lanczos in 1950 in the context of eigenvalue problems [24] and in 1952 as an iterative method for the solution of linear systems [25]. Later his method was called the Biconjugate Gradient (BCG) method by Fletcher in 1976 [10].

This method was originally only successful for the nondegenerate case where all $a_k(z)$ are of degree 1, or equivalently $a_{0k}(z)$ of degree k, because we then have the simplest possible three-term recurrence relation. When the algorithm hits a singular block in the Padé table of $f(z)$, this simple recurrence breaks down because the Euclidean index will then be larger than 1. A cure for this was first proposed by Parlett, Taylor and Liu (1985) [28], who implemented a Look Ahead strategy in the Lanczos method. It basically introduces the idea of the Euclidean algorithm to jump over the singular blocks. Later, the idea was picked up again by Freund and Nachtigal (1991) when they proposed their Quasi Minimal Residual (QMR) algorithm [11] and by Brezinski and coworkers (1993) with their Method of Recursive Zoom (MRZ) [6]. Many of the connections explained in this paper were worked out in great detail by Gutknecht in a series of papers [19, 17, 18]

These methods all compute in one way or another the vectors $a_{0n}(M) r_0$, based on a recurrence relation for the polynomials $a_{0n}(z)$. The Euclidean algorithm progresses along a diagonal of the Padé table of $f(z)$ and is therefore somehow the most natural choice. There are however many other methods. Suppose that $b_{0n}(z)$ are the analogs of the polynomials $a_{0n}(z)$ but for the adjacent diagonal in the Padé table, i.e., they are related to the Gram matrix $H_n(zf)$ exactly like the $a_{0n}(z)$ are related to $H_n(f)$. It is then well known in the theory of Padé approximation that we do not only have recurrence relations between the successive denominators a_{0n} and the denominators b_{0n}, but these polynomials can also be combined in a coupled recurrence that progresses in a staircase-like fashion, alternating between the two adjacent diagonals. In

the simplest (no singular blocks) case, $a_{0,n+1}$ is constructed from a_{0n} and $a_{0,n-1}$ which leads to the Euclidean variant of the Lanczos algorithms we have mentioned before. It corresponds to the procedure now generally known under the name ORTHORES. But it is also possible to generate a coupled recurrence computing $a_{0,n+1}$ from a_{0n} and b_{0n} and simultaneously $b_{0,n+1}$ from b_{0n} and $b_{0,n-1}$, which leads to the ORTHODIR variant of the Lanczos algorithm. When computing $a_{0,n+1}$ from a_{0n} and b_{0n} and $b_{0,n+1}$ from $a_{0,n+1}$ and b_{0n}, we get the ORTHOMIN variant. The latter is the BCG method originally proposed by Lanczos [25]. But of course, many other combinations are possible depending on the recurrence used to compute $a_{0,n+1}$ and $b_{0,n+1}$ [2]. There also exist symmetrized versions of all these methods that get then similar names e.g., BIORES, BIODIR, BIOMIN etc. Their generalization to the case where singular blocks appear is somewhat tedious but not impossible. For example, the MRZ method is the ORTHODIR algorithm adapted to avoid breakdown caused by the presence of singular blocks.

In 1984, Sonneveld [31] proposed his Conjugate Gradient Squared (CGS) algorithm. This algorithm does not use the transposed matrix M^T. The basic idea is that a recurrence is used for the polynomials $(a_{0n}(z))^2$ and vectors $r_k = (a_{0n}(M))^2 r_0$ are generated. His variant is a squared BIOMIN version. Similarly, also squared versions of BIODIR, BIORES and all other possibilities can be obtained, using recurrences for $(a_{0n})^2$ and $(b_{0n})^2$. These methods however display a "wild" convergence behavior and yet other variants are designed to smoothen the convergence. The Bi-CGSTAB algorithm of Van der Vorst (1992) [33] is an example.

7. Eigenvalue problems and incurable breakdown. The relation $\hat{Z}_n^T M \hat{Y}_n = D_n T_n$ does not only provide the desirable short recurrence relation, it gives also the following unexpected result. Suppose for simplicity that $\alpha_{0,n+1} = m$, so that the block matrix $\hat{Y}_n$ is square (as will be all the other block matrices with index n). If not, we shall see below what will happen. Since

$$\hat{Z}_n^T \hat{Y}_n = D_n \quad \Rightarrow \quad \hat{Z}_n^T = D_n \hat{Y}_n^{-1}$$

we get

$$\hat{Y}_n^{-1} M \hat{Y}_n = T_n$$

which shows that M and T_n are similar matrices and thus, they have the same eigenvalues. Therefore, the Lanczos algorithm also reduces an eigenvalue problem for M to a much simpler eigenvalue problem for the (block) tridiagonal matrix T_n. This application was in fact one of the motivations of Lanczos to design his algorithm in [24]. The most recent book on this subject is [29]. This eigenvalue problem is a good framework to have a look at the breakdown behavior of the Lanczos algorithm. We already observed that $f(z) = z_0^T (zI - M)^{-1} y_0$, meaning that (M, y_0, z_0) is a state space realization of $f(z)$. This realization need not be minimal though and this can cause some problems. The following numbers are important

$$\begin{aligned} m &= \text{size of } M \\ r_y &= \operatorname{rank} Y_\infty \\ r_z &= \operatorname{rank} Z_\infty \\ r &= \operatorname{rank} H(f) = \operatorname{rank} Z_\infty^T Y_\infty = \deg f(z). \end{aligned}$$

Clearly, r is the McMillan degree of the system and $m \geq \max\{r_y, r_z\} \geq \min\{r_y, r_z\} \geq r$. The poles of $f(z)$ are eigenvalues of M, but M may have some spurious eigenvalues that cancel out in $f(z)$ when $r < m$, i.e., when the realization is not minimal. When seen as a Padé approximation method, we find that the Lanczos algorithm only uses information enclosed in the Markov parameters, i.e., in the coefficients of the series $f(z)$, thus we can not expect to recover all the information about the eigenvalues of M by this algorithm. So, what can be expected and what do we loose?

This is basically enclosed in the Mismatch Theorem of Taylor [32]. To understand this theorem, suppose (without loss of generality) that we start with M in its Jordan form. This can always be obtained with some equivalence transformation on the state space triple for $f(z)$. So let $M = \text{diag}(J_1, \ldots, J_I)$ with J_i simple Jordan blocks. Let the vectors $z_0 = [z_{(1)}^T | z_{(2)}^T | \ldots | z_{(I)}^T]^T$ and $y_0 = [y_{(1)}^T | y_{(2)}^T | \ldots | y_{(I)}^T]^T$ be subdivided correspondingly, then

$$f(z) = z_0^T (zI - M)^{-1} y_0 = \sum_{i=1}^{I} z_{(i)}^T (zI - J_i)^{-1} y_{(i)}.$$

If J_i has size $\kappa(i)$, then this may be written as a partial fraction decomposition of $f(z)$:

$$f(z) = \sum_{i=1}^{I} \sum_{j=1}^{\kappa(i)} \frac{\rho_{ij}}{(z - \lambda_i)^j}$$

where λ_i is the eigenvalue associated with J_i. Some of the λ_i may be equal though. If we group the terms with equal λ_i, we get something of the form

$$f(z) = \sum_{i=1}^{I'} \sum_{j=1}^{\kappa'(i)} \frac{\rho'_{ij}}{(z - \lambda'_i)^j}$$

where $\lambda'_0, \ldots, \lambda'_{I'}$ are all different and. The $\kappa'(i)$ are made as small as possible. They are obviously at most equal to the size $\kappa_{\max}(i)$ of the maximal Jordan block associated with λ'_i. It can be smaller when $\rho'_{ij} = 0$ for $j = \kappa'(i) + 1, \ldots, \kappa_{\max}(i)$. This shows first of all that the best we can hope for is to recover λ'_i with a multiplicity $\kappa_{\max}(i)$, i.e., we certainly loose the information about the geometric multiplicity of λ'_i. This information is not in $f(z)$ and hence we can not recover it.

The Lanczos algorithm will beak down when all the information from $f(z)$ is obtained. In terms of Padé approximation, this means that we have an exact approximant. This happens when for some Kronecker index we have $\alpha_{0,n+1} = \text{rank}\, H(f) = r = \deg f =$ McMillan degree of the system. In the remainder of this section, we shall denote $\alpha_{0,n+1}$ as α. Note that

$$r = \deg f(z) = \sum_{i=1}^{I'} \kappa'(i) = \alpha.$$

Now, if $r < m$, the realization of the system was not minimal. For the solution of the system $Mx = b$, there is no problem at all since the residual will be zero, meaning that we have found an exact solution. For the eigenvalue problem however, we have reached the point where no more information can be extracted about the spectrum of

M. We have to consider two gradations in the non-minimality of the realization: (A) $\kappa'(i) = \kappa_{\max}(i)$ for all $i \in \{1, \ldots, I'\}$ and (B) $\kappa'(i) < \kappa_{\max}(i)$ for some $i \in \{1, \ldots, I'\}$.

In case (A), $m > \alpha = r = \min\{r_z, r_y\}$, meaning that the system is reachable or observable. If it were both, then it would be minimal and thus $m = r$. Suppose $r = r_y < r_z$, then the Lanczos algorithm will end because $\hat{y}_\alpha = \hat{y}_{0,n+1} = a_{0,n+1}(M)r_0 = 0$. Then $\mathcal{K}_{\alpha-1}(y_0, M) = \mathcal{K}_\alpha(y_0, M) = \cdots = \mathcal{K}_\infty(y_0, M)$. Note that $\alpha = \dim \mathcal{K}_\infty(y_0, M) = \operatorname{rank} Y_\infty = r_y$. Thus this subspace is M-invariant. The matrix $\hat{Y}_n$ is not square and thus not invertible, but when considered as an operator restricted to this invariant subspace, it is, and therefore we still have that the spectrum of the block tridiagonal matrix T_n in the relation $\hat{Z}_n^T M \hat{Y}_n = D_n T_n$ is the same as the spectrum of M, also restricted to this subspace $\mathcal{K}_{\alpha-1}(y_0, M)$. These eigenvalues are given by the zeros of $a_{0,n+1}(z)$. The other possibility where $r = r_z < r_y$ is completely symmetric.

In case (B), i.e., when $\kappa'(i) < \kappa_{\max}(i)$ for some i, then $r < \min\{r_y, r_z\}$ and some incurable breakdown will inevitably occur. This means that the algorithm breaks down with a Kronecker index $\alpha = r < \min\{r_y, r_z\}$. What happens here is that our starting vectors were not appropriate for the eigenvalue problem. This is because y_0 (resp. z_0) does not have components along all the right (resp. left) principal vectors associated with the maximal Jordan block for each eigenvalue. Therefore the algorithm is trapped in the smaller dimensional spaces and we can only recover the spectral information for M restricted to those spaces. We can actually construct these spaces, because it is possible to find y_0' and/or z_0' such that $f(z) = z_0'^T(zI - M)^{-1}y_0'$ is a realization for $f(z)$ with $r = \alpha = \min\{r_y', r_z'\}$, which brings us back to the previous situation. This implies that we still have that $a_{0,n+1}(z)$ is the characteristic polynomial of T_n and it turns out to be the minimal polynomial of M restricted to some M-invariant subspace $\mathcal{K}_{\alpha-1}(y_0', M)$ or $\mathcal{K}_{\alpha-1}(z_0', M)$ (depending on the case). The latter is a formulation of the Mismatch Theorem. How these vectors y_0' and z_0' can be constructed from the information one has, can e.g. be found in [19], see also [27].

The Euclidean algorithm, or all the variants we have mentioned will give a general framework to avoid a curable breakdown in the Lanczos algorithm, i.e., it says how to jump over the singular blocks in the Padé table of $f(z)$. The Mismatch Theorem says what happens when the situation is not curable.

A curable breakdown ideally corresponds to avoiding a division by zero. In practical situations however, one will never get an exact zero, and serious numerical trouble can be expected when such a number is nearly zero. The borders of the blocks are numerically blurred and it is then advisable to leap over near-singular blocks when we get a near-breakdown. One then computes in the sequence of polynomials a_{0n} only those that are numerically well defined, so that they are computed in a numerically stable way. This Look Ahead idea can be used in all possible related situations like Padé approximation or more general rational interpolation problems, the computation of orthogonal polynomials, (modified) moment problems, the construction of quadrature formulae, Hankel and Toeplitz solvers etc. [20, 12].

8. General Gram matrix and Toeplitz case. We have seen before that to get a short (three-term) recurrence relation for the (block) orthogonal polynomials, the Gram matrix had to be Hankel. In general, one might consider an arbitrary matrix to be the Gram matrix. Such a situation occurs when we consider Krylov subspaces

$$\mathcal{K}_n(y_0, M) \quad \text{and} \quad \mathcal{K}_n(z_0, N^T).$$

The moments are then given by

$$f_{ij} = z_i^T y_j = z_0 N^i M^j y_0, \quad i,j = 0,1,\ldots,n.$$

Define as before

$$Y_n = [y_0|y_1|\ldots|y_n] \quad \text{and} \quad Z_n = [z_0|z_1|\ldots|z_n]$$

then

$$H_n = Z_n^T Y_n = [f_{ij}]_{i,j=0,\ldots,n}$$

is an arbitrary (i.e., without any structure) matrix. A bi-orthogonalization will generate

$$\hat{Y}_n = Y_n A_n \quad \text{and} \quad \hat{Z}_n = Z_n B_n$$

with A_n and B_n upper triangular matrices and

$$\hat{Z}_n^T \hat{Y}_n = D_n$$

a (block) diagonal matrix. Moreover [8] (symmetric formulations exist with M replaced by N)

$$\hat{Z}_n^T M \hat{Y}_n = D_n T_n$$

with T_n a block upper Hessenberg matrix which is defined by

$$F(a_{0,n+1}) A_n = A_n T_n$$

where $a_{00}(z), a_{01}(z), \ldots$ are the first polynomials in the successive blocks of the right block orthogonal polynomials. The stacking vectors of these polynomials are the columns of the matrix A_n. As we might expect, T_n is not block tridiagonal, but block upper Hessenberg, which is caused by the lack of structure in the Gram matrix H_n. We also get some Padé-like rational approximants here. It can be shown [8] that if we associate formal series with the rows of H_∞:

$$f^{[i]}(z) = \sum_{k=0}^{\infty} \frac{f_{ik}}{z^{k+1}}$$

then the rational functions

$$f_n^{[i]}(z) = e_i^T B_n^{-T} D_n (zI - T_n)^{-1} e_0 = \frac{c_{0,n+1}^{[i]}}{a_{0,n+1}}$$

(e_j is is the zero vector with a 1 at position j) will approximate $f^{[i]}$ in the sense that for $i = 0,1,\ldots,\nu = \alpha_{0,n+1} - 1$

$$\deg\left(f^{[i]}(z) - f_n^{[i]}(z)\right) < -(\nu + 2)$$

where $\alpha_{0,n+1} = \deg a_{0,n+1}(z)$ is the $(n+1)$-st right Kronecker index for H_∞. Comparing the number of coefficients in the polynomials $c_{0,n+1}^{[i]}$ and $a_{0,n+1}$, with the number

of coefficients that are fitted, we see that the first number exceeds the latter number by 1, showing that we get here a vector Padé or multipoint Padé approximant for the formal series $f^{[i]}(z)$.

The short recurrence is lost to the general structure of the Gram matrix. However, another situation of a short recurrence of different type can be obtained for polynomials orthogonal on the unit circle of the complex plane. The moments then satisfy

$$(z^k, z^l) = \frac{1}{2\pi} \int_0^{2\pi} e^{ik\theta} e^{-il\theta} d\mu(\theta) = f_{k-l}$$

which implies that the Gram matrix is Toeplitz. This occurs in our previous situation when $NM = I$. The Padé approximation problem behind this situation simplifies because there are only two different series involved, viz.

$$f^+(z) = \sum_{k=0}^{\infty} f_{-k} z^k \quad \text{and} \quad f^-(z) = \sum_{k=1}^{\infty} f_k z^{-k}.$$

We get so called two-point Padé approximants for these series or equivalently a Laurent-Padé approximant for the formal Laurent series $f(z) = f^+(z) + f^-(z)$ [7]. The path that is followed in the Laurent-Padé table is a row instead of a diagonal. When there occur singular blocks, we get a rather complicated block structure in the upper Hessenberg matrix T_n. This is due to the fact that the right Euclidean index α_{n+1} which gives the size of such a block is given by $\alpha_{n+1} = 1 + \alpha^+_{n+1} + \alpha^-_{n+1}$ where α^+_{n+1} and α^-_{n+1} are right Iohvidov indices for that block and only the first $\alpha^-_{n+1} + 1$ polynomials in that block can be obtained by a simple multiplication with z, as was the case for a Hankel matrix. It can be shown however [8] that the relation

$$F(a_{0,n+1}) A_n = A_n T_n$$

with the complicated block Hessenberg matrix T_n can be brought in a much simpler form

$$F(a_{0,n+1}) A_n U_n = A_n T'_n$$

where U_n is a unit upper triangular (block) bidiagonal and T'_n is a block lower triangular bidiagonal

$$U_n = \begin{bmatrix} U_{00} & U_{01} & & \\ & U_{11} & \ddots & \\ & & \ddots & U_{n-1,n} \\ & & & U_{nn} \end{bmatrix}, \quad T'_n = \begin{bmatrix} T_{00} & & & \\ T_{10} & T_{11} & & \\ & \ddots & \ddots & \\ & & T_{n,n-1} & T_{nn} \end{bmatrix}$$

where all the blocks have a simple structure. Hence the eigenvalues of M which can be found as the zeros of $a_{0,n+1}(z)$ are also the solutions of a simple generalized eigenvalue problem since

$$a_{0,n+1}(z) = \det(zU_n - T'_n).$$

For the details we refer to [8]. It is somewhat more complicated, but it is still possible to design look-ahead algorithms for the Toeplitz case. This can include (extended) Euclidean algorithms, Lanczos algorithms etc. These algorithms are not direct generalizations of the Levinson algorithm though. See in this context also [20].

9. Conclusion. We have illustrated that with the simple idea of the Euclidean algorithm that already existed three centuries BC, we catch the essence of the recursive computations that are needed to compute rational (Padé) approximants for formal power series. It works, even for the situation where the Padé table is not normal. The same type of algorithm applied for power series at infinity, i.e., series in z^{-1}, solves the (minimal) partial realization problem in linear system theory. When this algorithm is linked with the Gram-Schmidt (bi-)orthogonalization of Krylov subspaces, generated by a matrix, it is possible to find here the basis of the Lanczos method for the iterative solution of systems with a matrix that is neither symmetric nor positive definite. It is also the basis for a procedure to reduce a matrix to a (block) tridiagonal form by similarity transformations, which is one step for computing its eigenvalues. These iterative methods were originally discovered by Lanczos in 1950, 1952. Until recently however, they were almost exclusively applied for the case of a positive definite symmetric matrix. Many of the algorithms that were recently designed in this context can be easily understood and classified using the theory of Padé approximants and the idea of the Euclidean algorithm to jump over singular blocks in the Padé table. We show that these algorithms compute Padé approximants which are located near a diagonal in the Padé table and require the computation of formal orthogonal polynomials which are orthogonal with respect to a Hankel matrix. To illustrate the generality of this idea we briefly surveyed some results for the computation of a row in the Padé table, where the polynomials involved are orthogonal with respect to a Toeplitz matrix.

REFERENCES

[1] R. ANDREE, *Selections from modern abstract algebra*, Holt, Rinehart and Winston, New York, 1958.

[2] C. BAHEUX, *New implementations for Lanczos method*, Numerical Algorithms, (1994). Presented at the 4th International Symposium on Orthogonal Polynomials and their Applications, Evian, October 1992.

[3] E. BERLEKAMP, *Algebraic coding theory*, McGraw-Hill, New York, 1968.

[4] C. BREZINSKI, *Padé-type approximants and general orthogonal polynomials*, vol. 50 of ISNM, Birkhäuser Verlag, Basel, 1980.

[5] ———, *History of continued fractions and Padé approximants*, vol. 12 of Springer series in Computational Mathematics, Springer-Verlag, Berlin, 1990.

[6] C. BREZINSKI, M. REDIVO ZAGLIA, AND H. SADOK, *A breakdown free Lanczos's type algorithm for solving linear systems*, Numerische Mathematik, 63 (1993), pp. 29–38.

[7] A. BULTHEEL, *Laurent series and their Padé approximations*, vol. OT-27 of Series in Operator Theory and its Applications, Birkhäuser Verlag, Basel-Boston, 1987.

[8] A. BULTHEEL AND M. VAN BAREL, *Formal orthogonal polynomials and rational approximation for arbitrary bilinear forms*, Technical Report TW163, Department of Computer Science, K.U. Leuven, December 1992.

[9] A. DRAUX, *Polynômes orthogonaux formels – applications*, vol. 974 of Lecture Notes Mathematics, Springer-Verlag, Berlin, 1983.

[10] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis Dundee 1975, G. Watson, ed., no. 506 in Lecture Notes in Mathematics, Springer Verlag, 1976, pp. 73–89.

[11] R. FREUND AND N. NACHTIGAL, *QMR: A quasi-minimal residual method for non-Hermitian matrices*, Numerische Mathematik, 60 (1991), pp. 315–339.

[12] R. FREUND AND H. ZHA, *A look-ahead strategy for the solution of general Hankel systems*, Numer. Math., (1991). (To appear).

[13] G. GOLUB AND D. O'LEARY, *Some history of the conjugate gradient and Lanczos methods*, SIAM Review, 31 (1989), pp. 50–102.

[14] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, John Hopkins University Press, Baltimore, London, 2nd ed., 1989.

[15] W. GRAGG, *The Padé table and its relation to certain algorithms of numerical analysis*, SIAM Rev., 14 (1972), pp. 1–62.

[16] W. GRAGG AND A. LINDQUIST, *On the partial realization problem*, Lin. Alg. Appl., 50 (1983), pp. 277–319.

[17] M. GUTKNECHT, *A completed theory of the unsymmetric Lanczos process and related algorithms. Part II*, SIAM J. Matrix Anal. Appl., (1990).

[18] ———, *The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, the QD algorithm, biconjugate gradient squared algorithms and fast Hankel solvers*, in Proceedings of the Copper Mountain Conference on Iterative Methods, 1990.

[19] ———, *A completed theory of the unsymmetric Lanczos process and related algorithms. Part I*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 594–639.

[20] ———, *Stable row recurrences for the Padé table end generically superfast look-ahead solvers for non-Hermitian Toeplitz systems*, Lin. Alg. Appl., (1992). (To appear).

[21] M. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436.

[22] P. JENNINGS AND J. MCKEOWN, *Matrix computation*, John Wiley & Sons, 2nd ed., 1992.

[23] T. KAILATH, *Linear Systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.

[24] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.*, J. Res. Natl. Bur. Stand., 45 (1950), pp. 225–280.

[25] ———, *Solution of systems of linear equations by minimized iterations*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 33–53.

[26] J. MASSEY, *Shift-register synthesis and BCH decoding*, IEEE Trans. Inf. Theory, IT-15 (1969), pp. 122–127.

[27] B. PARLETT, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matr. Anal. Appl., 13 (1992), pp. 567–593.

[28] B. PARLETT, D. TAYLOR, AND Z. LIU, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Mathematics of Comp., 44 (1985), pp. 105–124.

[29] Y. SAAD, *Numerical methods for large eigenvalue problems*, Algorithms and Architectures for Advanced Scientific Computation, Manchester University Press/ Halsted Press, 1992.

[30] Y. SAAD AND M. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[31] P. SONNEVELD, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 36–52.

[32] D. TAYLOR, *Analysis of the look ahead Lanczos algorithm*, tech. report, Center for Pure and Applied Mathematics, University of California, Berkeley, 1982.

[33] H. VAN DER VORST, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CGS for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.