TOWARDS AN ERROR ANALYSIS OF FAST TOEPLITZ FACTORIZATION

by A.BULTHEEL

Report TW 44, May 1979

Applied Mathematic and Programming Division
Katholieke Universiteit Leuven (Belgium)

Towards an error analysis of fast Toeplitz factorization

Abstract

Two algorithms for the triangular decomposition of Toeplitz matrices are shown to be numerically unstable in general. For the special case of symmetric positive definite Toeplitz matrices, the former algorithms reduce to the Levinson algorithm and its continued fraction analogue. They are shown to be stable if the Toeplitz matrix is bounded away from singularity. There are some indications that the continued fraction form is more trustworthy for the computation of the reflectioncoefficients.

1. Fast factorization of Toeplitz matrices

Let $\{T_j\}$ be a set of Toeplitz matrices, based upon the coefficients $\{t_j\}_{-\infty}^{+\infty}$:

$$T_N = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-N} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_N & \cdots & t_1 & t_0 \end{bmatrix} \qquad N = 0,1,\ldots \tag{1}$$

and $\{\overline{T}_j\}$ and $\{\underline{T}_j\}$ two adjacent families based upon $\{\overline{t}_k\}_{-\infty}^{+\infty} = \{t_{k+1}\}_{-\infty}^{+\infty}$ and $\{\underline{t}_k\}_{-\infty}^{+\infty} = \{t_{k-1}\}_{-\infty}^{+\infty}$ respectively.
We have the following nesting of these families:

$$T_{j+1} = \left[\begin{array}{c|c} T_j & v_j \\ \hline w_j^T & t_0 \end{array}\right] = \left[\begin{array}{c|c} t_0 & \hat{v}_j^T \\ \hline \hat{w}_j & T_j \end{array}\right] = \left[\begin{array}{c|c} \underline{w}_j & \underline{T}_j \\ \hline \underline{t}_{j+2} & \underline{w}_j^T \end{array}\right] = \left[\begin{array}{c|c} \hat{\overline{v}}_j & \overline{t}_{-j-2} \\ \hline \overline{T}_j & \overline{v}_j \end{array}\right] \tag{2}$$

and similarly for $\overline{T}_{j+1}$ and $\underline{T}_{j+1}$.

We have set: $v_j^T = [\, t_{-j-1} \cdots t_{-1} \,]$, $w_j^T = [\, t_{j+1} \cdots t_1 \,]$.
Vectors with upper and lower bar are similarly defined and for any vector a, $\hat{a}$ denotes an order reversion: $\hat{a}=Ja$, with

$$J = \begin{bmatrix} & & & 1 & 1 \\ & & \cdot & \cdot & \\ & 1 & \cdot & & \\ 1 & & & & \end{bmatrix}$$

Relations (2) are a consequence of the persymmetry of a Toeplitz matrix:
I.e. symmetry with respect to the antidiagonal: $T = J\, T^T\, J$.
On condition that $\det T_j \neq 0$, $j = 0,1,\ldots$, this property brings forth that the Toeplitz family $\{T_j\}$ has the following factorizations [1,5]

$$T_N = \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ \star & & \boxed{R_i} & \\ & & & \ddots \\ & & \star & & 1 \end{bmatrix} \begin{bmatrix} r_0 & & & \\ & \ddots & & 0 \\ & & r_i & \\ & 0 & & \ddots \\ & & & & r_N \end{bmatrix} \begin{bmatrix} 1 & & \star & \\ & \ddots & & \\ & & \boxed{\hat{\bar{P}}_i^T} & \\ & 0 & & \ddots \\ & & & & 1 \end{bmatrix} \equiv L_N\, D_N (J\bar{U}_N^T J)$$

$$= \begin{bmatrix} 1 & & & \\ & \ddots & \star & \\ & & \boxed{\bar{P}_i} & \\ & 0 & & \ddots \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{p}_N & & & \\ & \ddots & & 0 \\ & & \bar{p}_i & \\ & 0 & & \ddots \\ & & & & \bar{p}_0 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ \star & & \boxed{\hat{R}_i^T} & \\ & \ddots & & \ddots \\ & \star & & & 1 \end{bmatrix} \equiv \bar{U}_N\, \bar{D}_N (J L_N^T J) \qquad (3)$$

and

$$T_N^{-1} = \begin{bmatrix} 1 & & & \\ & \ddots & \star & \\ & & \boxed{Q_i} & \\ & 0 & & \ddots \\ & & & & 1 \end{bmatrix} \begin{bmatrix} r_0^{-1} & & & \\ & \ddots & & 0 \\ & & r_i^{-1} & \\ & 0 & & \ddots \\ & & & & r_N^{-1} \end{bmatrix} \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ \star & & \boxed{\hat{\bar{Q}}_i^T} & \\ & \ddots & & \ddots \\ & \star & & & 1 \end{bmatrix} \equiv U_N\, D_N^{-1} (J \bar{L}_N^T J)$$

$$= \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ \star & & \boxed{\bar{Q}_i} & \\ & \ddots & \star & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{p}_N^{-1} & & & \\ & \ddots & & 0 \\ & & \bar{p}_i^{-1} & \\ & 0 & & \ddots \\ & & & & \bar{p}_0^{-1} \end{bmatrix} \begin{bmatrix} 1 & & \star & \\ & \ddots & & \\ & & \boxed{\hat{\bar{Q}}_i^T} & \\ & 0 & & \ddots \\ & & & & 1 \end{bmatrix} \equiv \bar{L}_N\, \bar{D}_N^{-1} (J U_N^T J) \qquad (4)$$

where

$$D_N = J\,\overline{D}_N\,J$$

$$L_N^{-1} = J\,\overline{L}_N^{T}\,J \quad ; \quad \overline{U}_N^{-1} = J\,U_N^{T}\,J \tag{5}$$

$$U_N^{-1} = J\,\overline{U}_N^{T}\,J \quad ; \quad \overline{L}_N^{-1} = J\,L_N^{T}\,J$$

so that also:

$$T_N\,U_N = L_N\,D_N$$

$$\overline{T}_N\,\overline{L}_N = \overline{U}_N\,\overline{D}_N \tag{6}$$

All these factorizations are nested, just like the T_j are. I.e. if N is augmented by 1, then the factors are also extended by an appropriate row or column, leaving the rest unaffected.

These factorizations may be given a Padé approximation interpretation. Indeed, if t_j is the (k+j+1)-st coefficient of a formal power series, for some $k \geqslant 0$ and $j = 0, \pm1, \pm2, \ldots$, then Q_i contains the coefficients of the monically normalized denominator of the Padé approximant [k/i], while $P_i p_i$ and $R_i r_i$ contain corresponding numerator and residual coefficients. The quantities with an upper bar refer to the row k+1 of the table. For more details see [1].

So called fast algorithms for obtaining these factors are known [2-5]. A possible form is:

Algorithm A

$$r_0 = t_0 = \overline{p}_0$$

$$Q_0 = 1 = \overline{Q}_0$$

for i = 0, 1, 2, ...

$$p_i = \hat{v}_i^{T} Q_i \qquad \overline{r}_i = w_i^{T} \overline{Q}_i$$

$$Q_{i+1} = \begin{bmatrix} 0 \\ Q_i \end{bmatrix} - \begin{bmatrix} \overline{Q}_i \\ 0 \end{bmatrix} \frac{p_i}{\overline{p}_i} \qquad \overline{Q}_{i+1} = \begin{bmatrix} \overline{Q}_i \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ Q_i \end{bmatrix} \frac{\overline{r}_i}{r_i}$$

$$r_{i+1} = r_i - \overline{r}_i\,p_i\,\overline{p}_i^{-1} = \overline{p}_i - p_i\,\overline{r}_i\,r_i^{-1} = \overline{p}_{i+1}$$

The factors L_N and $\overline{U}_N$ can be found if we simultaneously compute also the diagonals $\tilde{D}_N$ and $\underset{\sim}{D}_N$ in the following factorizations:

$$\underline{T}_N \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & \ddots & \\ \star & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & \star & & \\ & \ddots & \ddots & \\ & & \ddots & \star \\ 0 & & & 1 \end{bmatrix} \begin{bmatrix} p_N & & & 0 \\ & \ddots & & \\ & & p_i & \\ 0 & & & p_0 \end{bmatrix} \equiv \underline{T}_N \underline{X}_N = \underline{U}_N \underset{\sim}{D}_N$$

$$\overline{T}_N \begin{bmatrix} 1 & \star & & \\ & \ddots & \ddots & \\ & & \ddots & \star \\ 0 & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & \ddots & \\ \star & & & 1 \end{bmatrix} \begin{bmatrix} \overline{r}_0 & & & 0 \\ & \ddots & & \\ & & \overline{r}_i & \\ 0 & & & \overline{r}_N \end{bmatrix} \equiv \overline{T}_N \overline{X}_N = \underline{L}_N \tilde{D}_N \qquad (7)$$

to give <u>algorithm B</u>

$$r_0 = t_0 = \overline{p}_0$$

$$\overline{r}_0 = t_1 \qquad \Big| \qquad p_0 = t_{-1}$$

$$A_0^T = [\, t_0 \ \dots \ t_N \,] \qquad \Big| \qquad B_0^T = [\, t_{-N-1} \ \dots \ t_{-1} \,]$$

$$\overline{A}_0^T = [\, t_1 \ \dots \ t_{N+1} \,] \qquad \Big| \qquad \overline{B}_0^T = [\, t_{-N} \ \dots \ t_0 \,]$$

<u>for</u> $i = 0, 1, 2, \dots$

$$A_{i+1} = \underline{E}(A_i - \overline{A}_i \, p_i \, \overline{p}_i^{-1}) \qquad \Big| \qquad B_{i+1} = \underline{E}(B_i - \overline{B}_i \, p_i \, \overline{p}_i^{-1})$$

$$\overline{A}_{i+1} = \overline{E}(\overline{A}_i - A_i \, \overline{r}_i \, r_i^{-1}) \qquad \Big| \qquad \overline{B}_{i+1} = \overline{E}(\overline{B}_i - B_i \, \overline{r}_i \, r_i^{-1})$$

$$r_{i+1} = e_0^T A_{i+1} = e_0^T \hat{\overline{B}}_{i+1} = \overline{p}_{i+1}$$

$$\overline{r}_{i+1} = e_0^T \overline{A}_{i+1} \qquad \Big| \qquad p_{i+1} = e_0^T \hat{B}_{i+1}$$

Here $e_0^T = [\, 1 \ 0 \ \dots \ 0 \,]$ is the zeroth unit vector and

$$\overline{E} = \left[\begin{array}{c|c} 0 & \\ \vdots & I \\ 0 & \end{array}\right], \qquad \underline{E} = \left[\begin{array}{c|c} I & 0 \\ & \vdots \\ & 0 \end{array}\right]$$

represent chopping operations on the first, resp. last element of a vector.

The factors of T_N are then found from the relations

$$A_i = R_i\, r_i \qquad\qquad B_i = P_i\, p_i$$

$$\overline{A}_i = \overline{R}_i\, \overline{r}_i \qquad\qquad \overline{B}_i = \overline{P}_i\, \overline{p}_i$$

Those algorithms are called fast because they require an order N^2 computations, instead of the factorizing algorithms for arbitrary matrices that use about N times more operations.

This speeding up of the algorithm is of course due to the exploitation of the Toeplitz structure of the matrices. This structure is incorporated in the algorithm by what we could call a "Toeplitz trick". So in algorithm A e.g. we compute Q_{i+1} as a "linear combination" of Q_i and $\overline{Q}_i$. Only one Gaussian elimination step then is needed to zero out the elements above the diagonal in the (i+1)st column of the factor L_N (see(6)) instead of the i steps when not using this trick.

2. Numerical Instability

Unfortunately we have to pay for this saving of operations by a loss of numerical stability. Thus although very fast, one can come to an answer that is useless. This instability is caused by three facts:

(1) The recursivity does not allow a pivoting strategy like e.g. Gaussian elimination does.

(2) Some unstable elimination steps are performed which could be circomvented.

(3) The Toeplitz trick can also introduce numerical instability.

These facts can be illustrated with an example for algorithm A.

Let $\{t_k\} = \{\ldots,1,1,t_0=\varepsilon,1,\varepsilon,\ldots\}$ with some small $\varepsilon > 0$, then:

$$T_2 = \begin{bmatrix} \varepsilon & 1 & \varepsilon \\ 1 & \varepsilon & 1 \\ 1 & 1 & \varepsilon \end{bmatrix} \quad ; \quad Q_1^T = \hat{Q}_1^T = [-\varepsilon^{-1} \;\; 1]$$

$$Q_2^T = [0 \quad -\varepsilon^{-1} \quad 1] - [1 \quad -\varepsilon^{-1} \quad 0] = [-1 \quad 0 \quad 1] \qquad (8)$$

$$T_2 \begin{bmatrix} 1 & -\varepsilon^{-1} & -1 \\ & 1 & 0 \\ & & 1 \end{bmatrix} = \begin{bmatrix} \varepsilon & & \\ 1 & \varepsilon-\varepsilon^{-1} & \\ 1 & 1-\varepsilon^{-1} & \varepsilon-1 \end{bmatrix}$$

(note that these T_j are not positive definite)

(1) The numbers ε^{-1} that occur in the factorization are not necessary. E.g.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} T_2 \begin{bmatrix} 1 & -\varepsilon & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & 1-\varepsilon^2 & 0 \\ 1 & 1-\varepsilon & \varepsilon-1 \end{bmatrix}$$

contains reasonable numbers. The reason is that $T_0 = [\varepsilon]$ is nearly singular, while T_1 and T_2 are not. A matrix like T_1 also gives rise to numerical problems when using the Cholesky decomposition [6,19] . Pivoting techniques allow to avoid this problem.

(2) Because the algorithms stick to the unit triangular decomposition, they may intr duce numerical instability. E.g.

$$\bar{Q}_{i+1}^T = [0 \quad Q_i^T]\,\bar{p}_i \; - \; [\bar{Q}_i^T \quad 0]\,p_i \qquad (9)$$

would also be a column of an upper triangular factor, but not with 1 on

the diagonal. If $\overline{p}_i << p_i$ then (9) is much more reasonable to evaluate numerically than the recursion in A is. Even orthogonal transformations like Givens rotations could be used. (see also [23]).

(3) The Toeplitz trick causes the central number in Q_2 to be computed as a difference of two numbers of the order of ε^{-1}. This is clearly unstable. A Gaussian elimination e.g. would give Q_2^T as the difference of $[0 \quad 0 \quad 1]$ and $[1 \quad 0 \quad 0]$, which would cause no trouble.

If ε is not bounded away from zero, then it is not difficult to find a more elaborated proof for the numerical instability in the sense of de Jong [7] of algorithms A and B for the set of all Toeplitz matrices. The inherent instability of the Toeplitz trick kills the hope that fast and numerically stable algorithms exist for general Toeplitz matrix factorization.

3. The Levinson algorithm and its continued fraction variant

It is however reasonable to ask if there do not exist practical situations where the recursive techniques are stable on a special set of Toeplitz matrices.

If we do not want to destroy the Toeplitz structure, then there is no pivoting possible. This suggests the consideration of the set S_N = {all symmetric positive definite matrices that are bounded away from singularity }. On this set, the Cholesky algorithm is known to be stable, even without pivoting. So we concentrate on the set T_N of all Toeplitz matrices in S_N.

Such problems are of practical interest in autoregressive digital filtering. Algorithm A then becomes the well known Levinson algorithm [8] which is a standard procedure there.
Remark that $L_N = J\,\overline{U}_N\,J$ and $U_N = J\,\overline{L}_N\,J$, so that we obtain:
Algorithm C

$Q_0 = 1 \quad ; \quad r_0 = t_0$

for $i = 0,1,2,\ldots$

$p_i = [t_1 \ \ldots \ t_{i+1}]\ Q_i \quad ; \quad \rho_i = p_i\, r_i^{-1}$

$Q_{i+1}^T = [0 \quad Q_i^T] \ - \ [\hat{Q}_i^T \quad 0]\ \rho_i$

$r_{i+1} = r_i - p_i^2\, r_i^{-1} = r_i(1-\rho_i^2)$

r_i is the i-the order filter prediction error, Q_i stores the coefficients of the i-th order filter and $\rho_i = p_i\, r_i^{-1}$ are the so called reflection coefficients or partial correlation (PARCOR) coefficients. If T_{i+1} is strictly positive definite, then $|\rho_j| < 1$, $j=0,1,\ldots,i$ and thus r_j decreases from t_0 to some non negative value, which need not be zero, even when N approaches infinity [9] .

The recursion in algorithm C is also the recursion for the Szegö orthogonal polynomials on the unit circle [10,11] so we could call the ρ_i also Szegö parameters. Algorithm C is also the recursion for the denominators in the zeroth row of the Chebyshev-Padé table [16] , for the Schur sequences [12,13] and is also related to two point Padé tables [14] and Darlington factorization in network theory [15] .

Some of the following relations are of further interest:

Set $\phi_i(z) = \sum_{j=0}^{i} Q_{ji}\, z^j$, $Q_{ji} = e_j^T Q_i$, $j=0,1,\ldots,i$, $i \geqslant 0$

then $\phi_i(0) = -\rho_{i-1}$ and

$$\begin{bmatrix} \phi_{i+1}(z) \\ \hat{\phi}_{i+1}(z) \end{bmatrix} = \begin{bmatrix} z & -\rho_i \\ -z\rho_i & 1 \end{bmatrix} \begin{bmatrix} \phi_i(z) \\ \hat{\phi}_i(z) \end{bmatrix} = \prod_{j=0}^{i} \begin{bmatrix} z & -\rho_j \\ -z\rho_j & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad (10)$$

$\{\hat{\phi}_j(z)\}$ is a solution of the difference equation

$$\rho_i\, f_{i+2}(z) - (\rho_i + \rho_{i+1}\, z)\, f_{i+1}(z) + \rho_i z(1-\rho_i^2)\, f_i(z) = 0 \qquad (11)$$

another independent solution is $\hat{\psi}_i(z)$ with $\{\psi_i\}$ the so called polynomials of the second kind, obeying

$$\begin{bmatrix} \psi_{i+1}(z) \\ -\hat{\psi}_{i+1}(z) \end{bmatrix} = \begin{bmatrix} z & \rho_i \\ z\rho_i & 1 \end{bmatrix} \begin{bmatrix} \psi_i(z) \\ -\hat{\psi}_i(z) \end{bmatrix} = \prod_{j=0}^{i} \begin{bmatrix} z & \rho_j \\ z\rho_j & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad (12)$$

The n-th convergent of the T-fraction [14]

$$t_0\Big(1 + \frac{2\rho_0 z}{\big| 1 - \rho_0 z} \Big| - \frac{\rho_1(1-\rho_0^2)z}{\big| \rho_1 + \rho_2 z} \Big| + \sum_{j=2}^{\infty} \frac{-\rho_{j-2}\rho_{j+1}\,(1-\rho_j^2)z}{\big| \rho_j + \rho_{j+1}\, z} \Big| \Big) \qquad (13)$$

is $\hat{\psi}_n(z)/\hat{\phi}_n(z)$ which is an order n+1 approximant for the power series

$t_0 + 2 \sum_{j=1}^{\infty} t_j\, z^j$, while

$$\frac{1}{2}\left(\frac{\hat{\psi}_n(z)}{\hat{\phi}_n(z)} + \frac{\psi_n(z)}{\phi_n(z)} \right) = \frac{z^n}{\hat{\phi}_n(z)\,\phi_n(z)} = \ldots + t_n z^{-n} + \ldots + t_1 z^{-1} + t_0 + \ldots + t_n z^n + \ldots$$

and thus fits the Laurent series $\sum_{-\infty}^{\infty} t_{|j|}\, z^j$ from $-n$ up to $+n$.

The algorithm B reduces to:

Algorithm D

$$r_0 = t_0 \qquad\qquad p_0 = t_1$$

$$A_0^T = [\, t_0 \ \ldots\ t_N \,] \qquad\qquad B_0^T = [\, t_1 \ \ldots\ t_{N+1} \,]$$

for $i = 0,1,2,\ldots$

$$\rho_i = p_i\, r_i^{-1}$$

$$A_{i+1} = \underline{E}\,(A_i - B_i\rho_i) \qquad\qquad B_{i+1} = \overline{E}\,(B_i - A_i\rho_i)$$

$$r_{i+1} = e_0^T A_{i+1} \qquad\qquad p_{i+1} = e_0^T B_{i+1}$$

Note that we wrote B instead of $\hat{B}$ for notational convenience.

This recursion is used in the so called ladder form techniques [17] for the realization of digital filters. In such a realization, only the reflection coefficients ρ_j are of importance. Thus rather than computing the denominators of the convergents of the continued fraction (13) , algorithm D only finds the coefficients of this fraction . Therefore, we could call it a continued fraction form of the Levinson algorithm.

Concerning the Toeplitz factorizations, we only have to consider $T_N = L_N D_N L_N^T$ and $T_N^{-1} = U_N D_N^{-1} U_N^T$. The other factors in (3) and (4) contain the same numbers.

The restriction that we concentrate on the set $\mathcal{T}_N$ is no constraint in practical situations. Indeed T_N will be singular if a diagonal element $r_i = 0$. Since $r_{j+1} = r_j(1-\rho_j^2)$, this quantity is monotonically non-increasing. So with increasing N, T_N will become closer to singularity, thus the condition of the problem becomes worse. Indeed we may show:

Theorem 1

The condition number of $T_N \in \mathcal{T}_N$ is increasing with N.

Proof

The condition number of T_N is: $\kappa_N = \sigma_{max}^{(N)} / \sigma_{min}^{(N)}$. Now

$$\sigma_{opt}^{(N)} = \underset{\|v_N\|=1}{opt} \; (v_N^T T_N v_N)$$

Increasing the dimension N, increases the space over which we optimize, due to the nesting property, so the optimum can only be made stronger with increasing N.

□

A more carefull analysis of the eigenvalues of the Toeplitz matrices [18] could eventually give accurate bounds on the L_2 condition number when N goes to infinity. However r_j has also the interpretation of the prediction error

$$r_{j+1} = \det T_{j+1} / \det T_j = t_0 \prod_{j=0}^{i} (1 - \rho_j^2) = Q_{j+1}^T T_{j+1} Q_{j+1} = \|Q_{j+1}\|_T^2$$

r_j will only become zero if the filter fits the system exactly, which will never occur in practical situations. The user will stop the algorithm at the moment that r_j becomes small enough. So we may characterize $\mathcal{T}_N$ as the set of Toeplitz matrices for which the corresponding ρ_i satisfy: $1-|\rho_i|^2 > r > 0$, i=0,1,...N-1. Then we have also $t_0 = r_0 \geq r_1 \geq \ldots \geq r_N > r^N > 0$. Therefore we sometimes denote this set by $\mathcal{T}_N(r)$.

4. The numerical stability on S_N

To prove the stability of the Cholesky algorithm on S_N, a backward error analysis in the sense of Wilkinson [19] may be used. This means that we look for a bound of ΔT_N, defined by

$$(T_N + \Delta T_N)^{-1} = fl(U_N)\,(fl(D_N))^{-1}\,(fl(U_N))^T \tag{14}$$

for algorithm C where U_N, D_N are as defined in (4) and $fl(A)$ stands for the numerically obtained approximation for A.
It will be notationally convenient to abreviate $fl(A)$ as $\bar{A}$ which is not to be confused with the upper bar in previous sections.
For algorithm D, ΔT_N is defined by

$$T_N + \Delta T_N = fl(L_N)\,(fl(D_N))^{-1}\,(fl(L_N))^T \tag{15}$$

and L_N is lower triangular with A_i in the i-th column beneath the diagonal ($i=0,1,\ldots,N$) and D_N as above. Note that this definition is different from the one in (3) but is more convenient to handle algorithm D.

ΔT_N will be called the backward error. From now on, we also suppose $t_0 = 1$ withouth loss of generality.

It is trivial that any recursive algorithm to find a triangular factorization of T_N (and by recursive we mean that the factorization of T_{j+1} is found after and with the aid of the factorization of T_j) will produce a ΔT_N that is nested in the following way:

$$\Delta T_{i+1} = \left[\begin{array}{c|c} \Delta T_i & \star \\ \hline \star & \star \end{array}\right] \tag{16}$$

Therefore T_N need not be Toeplitz and it is simply a consequence of the fact that T_i is contained in T_{i+1}.
That this is also true for a recursive algorithm, finding the factors U_N of T_N^{-1} needs some proof. The essence is that the factors U_N are also nested if T_N is a Toeplitz matrix, while this is not true in general.
Suppose

$$(T_{i+1} + \Delta T_{i+1})^{-1} = \bar{T}_{i+1}^{-1} = \bar{U}_{i+1}\,\bar{D}_{i+1}^{-1}\,\bar{U}_{i+1}^T = \frac{1}{\bar{r}_{i+1}} \left[\begin{array}{c|c} \bar{S}_i & \bar{u}_i \\ \hline \bar{u}_i^T & 1 \end{array}\right]$$

where

$$\bar{S}_i = \bar{r}_{i+1}\,\bar{U}_i\,\bar{D}_i\,\bar{U}_i + \bar{u}_i\,\bar{u}_i^T \quad , \quad \bar{u}_i = \underline{E}\,\bar{Q}_{i+1}$$

Set

$$\Delta T_{i+1} = \left[\begin{array}{c|c} \Delta T_i + \Delta & \Delta v_i \\ \hline \Delta v_i^T & \varepsilon \end{array}\right]$$

then

$$(T_{i+1} + \Delta T_{i+1}) \bar{T}_{i+1}^{-1} = I$$

implies:

(1) $(T_i + \Delta T_i + \Delta)(\bar{U}_i \bar{D}_i \bar{U}_i^T + \bar{u}_i \bar{u}_i^T/\bar{r}_{i+1}) + (\bar{v}_i + \Delta v_i) \bar{u}_i^T/\bar{r}_{i+1} = I$

or

$$(T_i + \Delta T_i + \Delta) \bar{u}_i \bar{u}_i^T + (\bar{v}_i + \Delta v_i) \bar{u}_i^T + \Delta \bar{U}_i \bar{D}_i \bar{U}_i^T \bar{r}_{i+1} = 0$$

(2) $(\bar{v}_i + \Delta v_i)^T (\bar{r}_{i+1} \bar{U}_i \bar{D}_i \bar{U}_i^T + \bar{u}_i \bar{u}_i^T) + (1 + \varepsilon)\bar{u}_i^T = 0$

(3) $(\bar{v}_i + \Delta v_i)^T \bar{u}_i + (1+\varepsilon) = \bar{r}_{i+1}$

or

$$1 + \varepsilon = \bar{r}_{i+1} - (\bar{v}_i + \Delta v_i)^T \bar{u}_i$$

(3) in (2) gives:

$$(\bar{v}_i + \Delta v_i)^T = - \bar{u}_i^T (T_i + \Delta T_i)$$

and this in (1):

$$(T_i + \Delta T_i + \Delta) \bar{u}_i \bar{u}_i^T - (T_i + \Delta T_i) \bar{u}_i \bar{u}_i^T + \Delta \bar{U}_i \bar{D}_i \bar{U}_i^T \bar{r}_{i+1} = 0$$

or

$$\Delta \bar{S}_i = 0$$

In general $\bar{S}_i$ will be of full rank and so $\Delta = 0$

Thus we have proved:

Theorem 2

Any recursive algorithm producing the triangular pactors of $\{T_i\}$ or $\{T_i^{-1}\}$ $i = 0,1,2,\ldots$ will cause a backward error on T_i that is nested as in (16) if $\{T_i\}$is a family of symmetric Toeplitz matrices.

However, this is about the only structure that ΔT_N allows, apart from its symmetry. It is clearly not of Toeplitz form. Since the Cholesky algorithm is applicable on any element in S_N , a bound for ΔT_N solves the question of its stability, however because algorithms C and D are only applicable on the subset T_N of S_N , and $T_N + \Delta T_N$ need not be in this subset, we can not conclude anything from ΔT_N about the numerical stability of these algorithms if we use the definitions of de Jong [7] . We then can not even speak about backward stability of algorithms C and D . Although the estimation of ΔT_N in S_N may be of some value to overcome this shortcomming, standard backward error analysis techniques are not applicable here and we will not persue this any further at the moment.

In this sense thus the Cholesky algorithm is not comparable with algorithms C and D, not even in its recursive implementations [20-22] . See also [24] for an error analysis.

5. Numerical stability of C on $\mathcal{T}_N$

Since the factor $\bar{U}_N$ (as it is computed by a numerical implementation of algorithm C) will in general not be a factor of an inverse of some Toeplitz matrix in $\mathcal{T}_N$, the question of numerical stability of C must be generalized now to the following problem:
Is there an upper triangular $\tilde{U}_N$ and a diagonal $\tilde{D}_N$ in a numerical neighbourhood of $\bar{U}_N$ and $\bar{D}_N$ such that

$$\tilde{T}_N = \tilde{U}_N^{-1} \tilde{D}_N \tilde{U}_N^{-T} \in \mathcal{T}_N \qquad (17)$$

is in a numerical neighbourhood of T_N?

First remark that there is a 1-1 relation between the set $\mathcal{B}_N$ of Szegö parameters, bounded away from 1 : $\mathcal{B}_N = \{\rho \mid \rho = \{\rho_i\}_0^{N-1}, |\rho_i| < 1-r < 1, i = 0,1,\ldots,N-1\}$, the set $\mathcal{T}_N$ and the set of its factors $\{U_N\}$ and $\{D_N\}$. More precisely, we define by algorithm E the following factors

$\tilde{U}_N = U_N(\tilde{\rho})$, $\tilde{D}_N = D_N(\tilde{\rho})$ and $\tilde{T}_N = T_N(\tilde{\rho})$ as follows:

$\tilde{r}_0 = 1$; $\tilde{Q}_0 = 1$; $\tilde{Q}_{i+1}^T = [\,0 \quad \tilde{Q}_i^T]^T - [\tilde{Q}_i^T \quad 0]^T \tilde{\rho}_i$;

$\tilde{r}_{i+1} = \tilde{r}_i (1 - \tilde{\rho}_i^2)$, $i = 0,1,\ldots,N-1$

$$\tilde{U}_N = \begin{bmatrix} 1 & \star & & \\ & \ddots & \begin{bmatrix} \tilde{Q}_i \end{bmatrix} & \star \\ & 0 & & \ddots \\ & & & & 1 \end{bmatrix} \quad ; \quad \tilde{D}_N = \mathrm{diag}(\tilde{r}_0, \tilde{r}_1, \ldots, \tilde{r}_N)$$

$\tilde{T}_N$ as in (17).

Conversely $\tilde{\rho}$ is defined completely by $\tilde{T}_N \in \mathcal{T}_N$ or what is the same by its elements $\tilde{t} = \{\tilde{t}_0 = 1, \tilde{t}_1, \ldots, \tilde{t}_N\}$ therefore we sometimes denote this as $\tilde{\rho} = \rho(\tilde{t})$.

So the problem of numerical stability of algorithm C can be reformulated as
Is there some $\tilde{\rho} \in \mathcal{B}_N$ for which $\tilde{U}_N$ and $\tilde{D}_N$ and $\tilde{T}_N$ are in a numerical neighbourhood of $\bar{U}_N$, $\bar{D}_N$ and T_N respectively ?
We will show that the problem may be solved in two steps:
(1) Is $\tilde{U}_N$ and $\tilde{D}_N$ in a numerical neighbourhood of $\bar{U}_N$ and $\bar{D}_N$ when $\tilde{\rho} = \bar{\rho}$?
This is the problem of forward stability of the algorithm E.
(2) Is the computation of $\rho = \{\rho_i\}_0^{N-1}$ from $t = \{t_i\}_0^N$ backward stable?
That is: do there exist reasonable bounds for the magnitude of Δt when $\bar{\rho} = \rho(t + \Delta t)$?
Notice that because the set S_N is much larger than $\mathcal{T}_N$, the error $T_N(t+\Delta t) - T_N(t)$ will probably be much larger than ΔT_N as defined in section 4.

5.1 The forward stability of algorithm E

$\bar{D}_N$ is defined as diag $(\bar{r}_0,\ldots,\bar{r}_N)$ with

$\bar{r}_0 = 1$; $\bar{r}_{i+1} = fl(\bar{r}_i \quad fl(1 - fl(\bar{\rho}_i^2)))$, $i = 0,1,\ldots N-1$

and $\tilde{\bar{D}}_N = D_N(\bar{\rho})$ by the evaluation scheme E.

$$\bar{U}_N = \begin{bmatrix} 1 & \star & \star \\ & \ddots & \\ 0 & & 1 \end{bmatrix} \text{ and } \bar{Q}_0 = 1 \; ; \; \bar{Q}_{i+1} = fl\left(\begin{bmatrix} 0 \\ \bar{Q}_i \end{bmatrix} - fl\left(\begin{bmatrix} \hat{\bar{Q}}_i \\ 0 \end{bmatrix} \bar{\rho}_i \right) \right),$$

$i = 0,1,\ldots,N-1.$

Theorem 3

Algorithm E is forward stable on $\mathcal{B}_N$.

Proof:

We prove the induction step:

Suppose $\bar{Q}_i = \tilde{Q}_i + y_i$ and $\|y_i\| \leq \eta_i \|\tilde{Q}_i\|$ and $\|.\|$ denotes the usual norm in l_2 : $\|y_i\|^2 = y_i^T y_i$. Remark that $1 < \|Q_i\| < 2^i$.

Now:

$fl(\bar{Q}_i \bar{\rho}_i) = fl((\tilde{Q}_i + y_i)\bar{\rho}_i) = (I + E_1)(\tilde{Q}_i + y_i)\bar{\rho}_i = \tilde{Q}_i \bar{\rho}_i + z_i$

$z_i = (y_i + E_1 \tilde{Q}_i + E_1 y_i)\bar{\rho}_i$

$E_j = \text{diag}(\varepsilon_{0j},\varepsilon_{1j},\ldots,\varepsilon_{ij})$ here and below will have $|\varepsilon_{kj}| < \varepsilon$

with ε the machine precision and $\varepsilon_{0j} = 0$, $j = 1,2$

$$\begin{aligned} \bar{Q}_{i+1} &= fl([0 \quad \bar{Q}_i^T]^T - fl([\hat{\bar{Q}}_i^T \quad 0]^T \bar{\rho}_i)) \\ &= (I + E_2)([0 \quad \bar{Q}_i^T]^T - [\hat{\bar{Q}}_i^T \quad 0]^T \bar{\rho}_i - [\hat{z}_i^T \quad 0]^T) \\ &= (I + E_2)([0 \quad \tilde{Q}_i^T]^T - [\hat{\bar{Q}}_i^T \quad 0]^T \bar{\rho}_i) + (I + E_2)([0 \quad y_i^T]^T - [\hat{z}_i^T \quad 0]^T) \\ &= \tilde{Q}_{i+1} + E_2 \tilde{Q}_{i+1} + (I + E_2)([0 \quad y_i^T]^T - [\hat{z}_i^T \quad 0]^T) \end{aligned}$$

and thus

$$y_{i+1} = E_2 \tilde{Q}_{i+1} + (I + E_2)([0 \quad y_i^T]^T - [\hat{z}_i^T \quad 0]^T)$$

so that

$$\eta_{i+1}^2 = \|y_{i+1}\|^2 / \|\tilde{Q}_{i+1}\|^2 < (i+1)(\varepsilon^2 + (1+\varepsilon)^2(\eta_i^2 \|\tilde{Q}_i\|^2 + \|z_i\|^2)/\|\tilde{Q}_{i+1}\|^2)$$

$$\eta_{i+1}^2 < (i+1)(\varepsilon^2 + (1+\varepsilon)^2(\eta_i^2 4^i + \|z_i\|^2))$$

using

$$\|z_i\|^2 < (i+1)((1+\varepsilon)^2 \|y_i\|^2 + \varepsilon^2 \|\tilde{Q}_i\|^2) < (i+1)((1+\varepsilon)^2 \eta_i^2 + \varepsilon^2) 4^i$$

this yields:

$$\eta_{i+1}^2 < (i+1)(\varepsilon^2 + (1+\varepsilon)^2 4^i((i+1)\varepsilon + (1 + (i+1)(1+\varepsilon)^2)\eta_i^2) \qquad (18)$$

which remains bounded if $i \leq N < \infty$ and this concludes the induction step. The forward stability of the evaluation of $\tilde{D}_N$ is no problem, and thus the theorem is proved.

□

It seems reasonable to conclude from (18) that η_i grows at a rate of about $i!\ 2^{i.i/2}$, but this is a drastical overestimation because this can only happen if $\|\tilde{Q}_i\|$ (growing at a rate that is at most $\Pi_0^{i-1}(1 + |\bar{\rho}_j|)$) will grow like about $(2-r)^i$, which only happens if $|\bar{\rho}_j| \approx 1$ and thus $\tilde{r}_i$ is very steeply decreasing so that the algorithm will soon stop with small N. A practical situation seems to be that E is fairly stable in forward sense and a blowing up of the accumulated rounding error will be largely due to the conditioning of the problem.

5.2 Backward stability of C for the computation of $\rho(t)$.

Let us indicate the relation between t and ρ, as computed by algorithm C by $\rho_i = c(t_0,\ldots,t_{i+1})$. A numerical computation however will give a perturbed result $\bar{\rho}_i = fl(c)(t_0,\ldots,t_{i+1}) = \bar{c}(t_0,\ldots,t_{i+1})$.
Suppose $\bar{\rho}_j$, $j < i$ are given as an exact result of perturbed data: $\bar{\rho}_j = c(\bar{t}_0,\bar{t}_1,\ldots,\bar{t}_{j+1})$, then we will define $\bar{t}_{i+1}$ through $\bar{\rho}_i = c(\bar{t}_0,\ldots\bar{t}_{i+1})$
If $\bar{t} = \{\bar{t}_i\}_0^N$ will be close to $t = \{t_i\}_0^N$, then C is backward stable.
Let us try now to find bounds for $\varepsilon_i = \bar{t}_i - t_i$, $i = 0,1,\ldots,N$.

We have the expression

$$\bar{\rho}_i = fl\left(\frac{fl(t_{i+1} + fl([t_1\ldots t_i]\ \bar{u}_i))}{fl(\bar{r}_{i-1}\ fl(1-fl(\bar{\rho}_{i-1}^2)))} \right) \qquad (19)$$

$$\bar{u}_i = \underline{E}\ \bar{Q}_i = fl([0 \quad \bar{u}_{i-1}^T]^T - fl([1 \quad \hat{\bar{u}}_{i-1}^T]^T \bar{\rho}_{i-1}))$$

(a) The denominator equals:

$$\bar{r}_{i-1}(1 - \bar{\rho}_{i-1}^2(1 + \varepsilon^{(1)}))(1 + \varepsilon^{(2)}) = \bar{r}_{i-1}(1 - \bar{\rho}_{i-1}^2) + \eta^{(1)}$$

with

$$\eta^{(1)} = -\bar{r}_{i-1}\ \bar{\rho}_{i-1}^2\ \varepsilon^{(1)} + r_{i-1}(1 - \bar{\rho}_{i-1}(1 + \varepsilon^{(1)}))\varepsilon^{(2)} = \bar{r}_{i-1}\ \varepsilon^{(3)}$$

and

$$|\varepsilon^{(1)}| < \varepsilon, |\varepsilon^{(2)}| < 2\varepsilon \text{ and } |\varepsilon^{(3)}| < 3\varepsilon$$

(b) The numerator equals:

$$n = (t_{i+1} + [t_1 \ldots t_i]\ (I + E^{(1)})\bar{u}_i)\ (1 + \varepsilon^{(4)})$$

$$E^{(1)} = \text{diag}\ (i\varepsilon^{(1,1)}, i\varepsilon^{(2,1)}, (i-1)\varepsilon^{(3,1)}, \ldots, 2\varepsilon^{(i,1)})$$

and $|\varepsilon^{(j,1)}| < \varepsilon$, $|\varepsilon^{(4)}| < \varepsilon$, so:

$$n = (t_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]\ \bar{u}_i - [\varepsilon_1 \ldots \varepsilon_i]\ (I + E^{(1)})\bar{u}_i + [t_1 \ldots t_i]\ E^{(1)}\bar{u}_i).$$

$$(1 + \varepsilon^{(4)}) = (t_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]\bar{u}_i + \eta^{(2)})(1 + \varepsilon^{(4)})$$

$$\eta^{(2)} = -[\varepsilon_1 \ldots \varepsilon_i](I + E^{(1)})\bar{u}_i + [t_1 \ldots t_i]E^{(1)}\ \bar{u}_i$$

(c) The error in the computation of $\bar{u}_i$ has the following effect:

Set $\bar{u}_j = u_j(\bar{t}_0, ., \bar{t}_j) + y_j$, $u_j^T(\bar{t}_0, \ldots, \bar{t}_j) = [0 \quad u_{j-1}^T(\bar{t})] - \bar{\rho}_{j-1}[1 \quad \hat{u}_{j-1}^T(\bar{t})]$

$$\bar{u}_i^T = ([0 \quad \bar{u}_{i-1}^T] - \bar{\rho}_{i-1}[1 \quad \hat{\bar{u}}_{i-1}^T](I + E^{(2)}))(I + E^{(3)})$$

$$E^{(k)} = \text{diag}\ (\varepsilon^{(j,k)}) \ , |\ \varepsilon^{(j,k)}| < \varepsilon \ , \ k = 2,3$$

So

$$\bar{u}_i = u_i(\bar{t}) + y_i$$

with

$$y_i^T = u_i^T(\bar{t})\ E^{(3)} + (I + E^{(3)})([0 \quad y_{i-1}^T]^T - z_i^T \bar{\rho}_{i-1})$$

$$z_i^T = [0 \quad \hat{y}_{i-1}^T]^T\ (I + E^{(2)}) + [1 \quad \hat{\bar{u}}_{i-1}^T]^T\ E^{(2)}$$

(d) So the numerator becomes:

$$n = (t_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]\ u_i(\bar{t}) + \eta^{(3)} + \eta^{(2)})(1 + \varepsilon^{(4)})$$

with

$$\eta^{(3)} = [\bar{t}_1 \ldots \bar{t}_i]y_i$$

(e) Finally

$$\bar{\rho}_i = \frac{(t_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]u_i(\bar{t}) + \eta^{(2)} + \eta^{(3)})\ (1 + \varepsilon^{(6)})}{\bar{r}_{i-1}\ (1 - \bar{\rho}_{i-1}^2) + \eta^{(1)}} = \frac{\bar{p}_i}{\bar{r}_i}$$

$$|\varepsilon^{(6)}| \lesssim 3\varepsilon$$

or

$$\bar{\rho}_i = \frac{t_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]\ \bar{u}_i + \eta^{(2)} + \eta^{(3)} + \eta^{(4)} + \eta^{(5)}}{\bar{r}_{i-1}\ (1 - \bar{\rho}_{i-1}^2)}$$

$$\eta^{(4)} = (t_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]\ \bar{u}_i + \eta^{(2)})\varepsilon^{(6)}$$

$$\eta^{(5)} = -\ \bar{\rho}_i\ \eta^{(1)}$$

The latter equality follows from

$$\frac{b + \eta^I}{a} = \frac{b}{a + \eta} \qquad \text{then} \qquad \eta^I = -\frac{b}{a + \eta}\ \eta$$

Thus

$$\varepsilon_{i+1} = \eta^{(2)} + \eta^{(3)} + \eta^{(4)} + \eta^{(5)}$$

where we need bounds for every term in the RHS.

$$|\eta^{(5)}| \leq (1-r)|\eta^{(1)}| < 4(1-r)|\bar{r}_{i-1}|\varepsilon < 4\varepsilon$$

$$|\eta^{(4)}| \leq \left| \frac{\bar{\rho}_i \bar{r}_i}{1 + \varepsilon^{(6)}} \varepsilon^{(6)} \right| \simeq |\bar{\rho}_i||\bar{r}_i| |\varepsilon^{(6)}(1 - \varepsilon^{(6)} + (\varepsilon^{(6)})^2 - \ldots)|$$

$$\simeq |\bar{\rho}_i||\bar{r}_i||\varepsilon^{(6)}| < 4\varepsilon$$

y_i can be bounded as we did in section 5.1.
We also suppose that $|\bar{t}_i| < 1 - r'$ with $0 < r' \leq r$, thus

$$T_N(\bar{t}) \in \mathcal{T}_N(r') \supset \mathcal{T}_N(r)$$

which is the same as requiring that $\bar{\rho} \in \mathcal{B}_N(r') \supset \mathcal{B}_N(r)$.
Thus $|\eta^{(3)}|$ can be bounded.
Similarly $|\eta^{(2)}|$ can be bounded by bounding the coefficients in $\bar{u}_i$ by binomial coefficients C_j^i e.g..
Thus we have proved the backward stability of C for the computation of the $\rho(t)$.
To put things together once more, we resume this section as follows: There exist boundedly perturbed data $\bar{t} = t + \Delta t$ (or what is the same a Toeplitz matrix being a boundedly perturbed T_N : $T_N(\bar{t}) \in \mathcal{T}_N$) such that with exact arithmetic, algorithm C would give $\bar{\rho} = \rho(\bar{t})$, which in turn gives $\tilde{U}_N = U_N(\bar{\rho})$ and $\tilde{D}_N = D_N(\bar{\rho})$ and those factors are boundedly perturbed versions of the factors $\bar{U}_N$ and $\bar{D}_N$ that are obtained by a numerical application of algorithm C. Thus we have proved:

Theorem 4

The Levinson algorithm C is numerically stable on the set $\mathcal{T}_N$

□

6. Numerical stability of D on $\mathcal{T}_N$

Let us define for a given $\tilde{\rho} \in \mathcal{B}_N$: $\tilde{D}_N = D_N(\tilde{\rho})$ as before in algorithm E and $\tilde{L}_N = L_N(\tilde{\rho})$ through the following algorithm F:

$$\tilde{A}_0^T = [t_0 \dots t_N] \quad ; \quad \tilde{B}_0^T = [t_1 \dots t_{N+1}]$$

$$\tilde{A}_{i+1} = \underline{E}(\tilde{A}_i - \tilde{B}_i \tilde{\rho}_i) \quad ; \quad \tilde{B}_{i+1} = \overline{E}(\tilde{B}_i - \tilde{A}_i \tilde{\rho}_i)$$

with

$$\tilde{L}_N = \begin{bmatrix} \begin{bmatrix} \tilde{A}_0 \end{bmatrix} & & 0 \\ & \ddots & \\ \star & \begin{bmatrix} \tilde{A}_i \end{bmatrix} & \ddots \\ & \star & \end{bmatrix}$$

then

$$\tilde{T}_N = T_N(\tilde{\rho}) = \tilde{L}_N \tilde{D}_N^{-1} \tilde{L}_N^T \quad \in \quad \mathcal{T}_N$$

The problem of numerical stability of algorithm D will again be solved in two steps:

(1) Is $\tilde{L}_N$ and $\tilde{D}_N$ in a numerical neighbourhood of $\bar{L}_N$ and $\bar{D}_N$ as they are computed by a numerical implementation of algorithm D when $\tilde{\rho} = \bar{\rho}$?

(2) Is the computation of the Szegö parameters ρ from the Toeplitz parameters t by algorithm D, numerically stable in backward sense?

Before we start the error analysis, let us first state the following

Lemma 2

If T is a Toeplitz matrix, symmetric and positive definite and let $L\,L^T$ (L lower triangular) be its Cholesky factorization, then

$|L_{ji}| \leq L_{ii} \quad \forall\, j \geq i \;,\; i = 0,1,\dots$

For the alternative factorization $U\,U^T$ (U upper triangular) we have

$|U_{ij}| \leq U_{jj} \;,\; \forall\, i \leq j \;,\; j = 0,1,\dots$

Proof

Since any submatrix of a symmetric positive definite T matrix of the form

$$\begin{bmatrix} T_{ii} & T_{ij} \\ T_{ji} & T_{jj} \end{bmatrix}$$

is positive definite, we have that $|T_{ij}|^2 \leq T_{ii}\, T_{jj} = T_{jj}^2$

Thus for the first column of the Cholesky factor L, which is $[T_{00} \quad T_{10} \ldots T_{N0}]^T \, T_{00}^{-1/2}$ we have that the conjecture is true. An induction proof then will give the result for the other columns.

The property carries over from L to U because the persymmetry of T allows us to write

$$J\, T^T\, J = T = (J\, L\, J)\,(J\, L^T\, J) = U\, U^T$$

□

The previous lemma gives us bounds on the numbers in $\tilde{A}_i$. Indeed all components of $\tilde{A}_i$ are bounded by $r_i < 1$.
The next lemma is needed to obtain bounds on $\tilde{B}_i$ components

Lemma 3

If $c(z)$ is analytic in the unit circle $D = \{z \mid |z| \leq 1\}$ and $|c(z)| < 1$ for $z \in D$, then the coefficients of its Taylor series are bounded by 1.

Proof

The k-th coefficient is given by Cauchy's formula

$$c_k = \frac{1}{2\pi j} \int_{-\pi}^{\pi} e^{-jk\theta}\, c(e^{j\theta})\, d\theta$$

Bounding this integral gives directly the result.

□

If we now let N go to infinity and set
$a_i(z) = [1 \; z \; z^2 \ldots]\, A_i$ and $b_i(z) = [1 \; z \; z^2 \ldots]\, B_i$
with A_i and B_i as generated in algorithm $D(N = \infty)$, then it is known that $c_i(z) = b_i(z)/a_i(z)$ is a sequence of Schur functions [12,13,15]. This means that $c_i(z)$ meets the requirements of lemma 3 and thus

if $c_i(z) = \sum_{j=0}^{\infty} c_{ji}\, z^j$, then $|c_{ji}| \leq 1$

From $b_i(z) = c_i(z)\, a_i(z)$ we then find

$$|b_{ji}| \leq \left|\sum_{k=0}^{j} c_{ki}\; a_{j-k,i}\right| \leq (j+1) r_i$$

Since a finite N does not alter the first (N-i) coefficients in A_i and B_i, we thus have proved the following:

Lemma 4

The coefficients in $\tilde{A}_i$ are bounded in modulus by $\tilde{r}_i$, the j-th coefficient of $\tilde{B}_i$ is bounded in modulus by $(j+1)\, \tilde{r}_i$

□

Remark that the bounds for A_i or B_i numbers are much stronger than the bounds we were able to give for Q_i in previous section.

Unfortunately the forward stability of algorithm F may be much more critical than the forward stability of E because now the A_i numbers are guaranteed to be of decreasing magnitude, but again this is mostly an effect of the conditioning.

Algorithm D will however have the advantage over the Levinson algorithm C that for D, the forward error of F will not interfer with the backward error for the computation of the reflectioncoefficients from $\{t_i\}_0^N$. This is unlike the previous section where we needed bounds for $\|y_i\|$, computed in section 5.1, to find bounds for the backward analysis in section 5.2.

6.1 The forward stability of algorithm F

Theorem 5

Algorithm F is forward stable on $\mathcal{B}_N$.

Proof

The forward stability of the evaluation of $\tilde{D}_N$ is trivial as before in algorithm E.

For the evaluation of $\tilde{L}_N$, we have to prove that when $\bar{A}_i = \tilde{A}_i + y_i$, then $\eta_i = \|y_i\|/\|\tilde{A}_i\|$ remains bounded.

We set also $\bar{B}_i = \tilde{B}_i + z_i$, then we will prove this boundedness by induction.

(a) $$fl(\bar{B}_i\,\bar{\rho}_i) = (I + E_1)(\tilde{B}_i\,\bar{\rho}_i + z_i\,\bar{\rho}_i) = \tilde{B}_i\,\bar{\rho}_i + v_i$$

$$E_k = \mathrm{diag}(\varepsilon_{jk}) \ , \ |\varepsilon_{jk}| < \varepsilon \ , \ j = 0,1,\ldots,N-i \ , \ k = 1,2,3,4.$$

$$v_i = E_1\,\tilde{B}_i\,\bar{\rho}_i + (I + E_1)\,z_i\,\bar{\rho}_i$$

$$\begin{aligned}\bar{A}_{i+1} &= \underline{E}\ fl(\bar{A}_i - fl(\bar{B}_i\,\bar{\rho}_i)) \\ &= \underline{E}(I + E_2)(\bar{A}_i - \tilde{B}_i\,\bar{\rho}_i - v_i) \\ &= \underline{E}(\tilde{A}_i - \tilde{B}_i\,\bar{\rho}_i) + y_{i+1} \\ &= \tilde{A}_{i+1} + y_{i+1}\end{aligned}$$

$$y_{i+1} = E_2\,\underline{E}^T\,\tilde{A}_{i+1} + \underline{E}(I + E_2)(y_i - v_i)$$

(b)

$$fl(\bar{A}_i\,\bar{\rho}_i) = (I + E_3)(\tilde{A}_i\,\bar{\rho}_i + y_i\,\bar{\rho}_i) = \tilde{A}_i\,\bar{\rho}_i + w_i$$

$$w_i = E_3\,\tilde{A}_i\,\bar{\rho}_i + (I + E_3)\,y_i\,\bar{\rho}_i$$

$$\bar{B}_{i+1} = \bar{E}\,fl(\bar{B}_i - fl(\bar{\rho}_i\,\bar{A}_i))$$

$$= \bar{E}(I + E_4)(\bar{B}_i - \tilde{A}_i\,\bar{\rho}_i - w_i) = \tilde{B}_{i+1} + z_{i+1}$$

$$z_{i+1} = E_4\,\bar{E}^T\,\tilde{B}_{i+1} + \bar{E}(I + E_4)(z_i - w_i)$$

(c) let us now use lemma 4 to estimate some norms:

$$\eta_{i+1}^2 = \|y_{i+1}\|^2/\|\tilde{A}_{i+1}\|^2 \leq (N-i)(\varepsilon^2+(1+\varepsilon)^2(\|\underline{E}v_i\|^2+\|\underline{E}y_i\|^2)/\|\tilde{A}_{i+1}\|^2)$$

$$\|\underline{E}v_i\|^2/\|\tilde{A}_{i+1}\|^2 \leq (N-i)(\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 + (1+\varepsilon)^2\|\underline{E}z_i\|^2/\|\tilde{A}_{i+1}\|^2)$$

$$\|\underline{E}y_i\|^2/\|\tilde{A}_{i+1}\|^2 = \|\underline{E}y_i\|^2/\|\tilde{A}_i\|^2.\|\tilde{A}_i\|^2/\|\tilde{A}_{i+1}\|^2 \leq \eta_i^2(N-i+1)/(1-\bar{\rho}_i^2)^2$$

$$\|\underline{E}z_i\|^2/\|\tilde{A}_{i+1}\|^2 \leq (N-i)(\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2+(1+\varepsilon)^2(\|\underline{E}y_{i-1}\|^2+\|\underline{E}w_{i-1}\|^2)/\|\tilde{A}_{i+1}\|^2)$$

$$\|\underline{E}w_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2 \leq (N-i+1)(\varepsilon^2\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2+(1+\varepsilon)^2\|y_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2)$$

$$\|\underline{E}z_i\|^2/\|\tilde{A}_{i+1}\|^2 \leq (N-i)\{\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 + (1+\varepsilon)^2[\,\|y_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2+(N-i+1).$$
$$(\varepsilon^2\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2+(1+\varepsilon)^2\|y_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2)]\}$$

$$= (N-i)\{\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2+(1+\varepsilon)^2[\{1+(N-i+1)(1+\varepsilon)^2\}\|y_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2$$
$$+(N-i+1)\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2]\}$$

$$\leq (N-i)\{\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 + (1+\varepsilon)^2[\{1+(N-i+1)(1+\varepsilon)^2\}\eta_{i-1}^2 +$$
$$(N-i+1)]\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2\}$$

$$\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2 \leq (N-i+2)(N-i+1)/\{(1-\bar{\rho}_{i-1}^2)(1-\bar{\rho}_i^2)\}^2$$

$$\eta_{i+1}^2 \leq (N-i).$$
$$\{\varepsilon^2 + (1+\varepsilon)^2.$$
$$\{\,(N-i).$$
$$\{\,\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 + (1+\varepsilon)^2(N-i).$$
$$\{\varepsilon^2\|\underline{E}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 + (1+\varepsilon)^2[\{1+(N-i+1)(1+\varepsilon)^2\}\eta_{i-1}^2 + (N-i+1)].$$
$$\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2$$
$$\}$$
$$\}$$
$$\}$$
$$\}$$

$$\eta_{i+1}^2 \leq (N-i).$$
$$\{\varepsilon^2+(1+\varepsilon)^2.$$
$$\{(N-i).$$
$$\{[(1+\varepsilon)^2(N-i)+1]\varepsilon^2\|\underline{\tilde{E}}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 + (1+\varepsilon)^2(N-i).$$
$$\{[1+(N-i+1)(1+\varepsilon)^2]\eta_{i-1}^2\}+(N-i+1)\|\tilde{A}_{i-1}\|^2/\|\tilde{A}_{i+1}\|^2$$
$$\}$$
$$\}$$
$$\}$$

$$\|\underline{\tilde{E}}\tilde{B}_i\|^2/\|\tilde{A}_{i+1}\|^2 \leq \tilde{r}_i^2 \sum_{j=1}^{N-i} j^2/\tilde{r}_{i+1}^2 = (N-i)(N-i+1)(2N-2i+1)/[6(1-\bar{\rho}_i^{-2})^2]$$

$$\eta_{i+1}^2 \leq (N-i).$$
$$\{\varepsilon^2+(1+\varepsilon)^2.$$
$$\{(N-i)^2.$$
$$\{\varepsilon^2[1+(N-i)(1+\varepsilon)^2](N-i+1)(2N-2i+1)/[6(1-\bar{\rho}_i^{-2})^2]+(1+\varepsilon)^2.$$
$$\{[1+(N-i+1)(1+\varepsilon)^2]\eta_{i-1}^2 +(N-i+1)\frac{(N-i+1)(N-i+2)}{(1-\bar{\rho}_{i-1}^{-2})^2(1-\bar{\rho}_i^{-2})^2}\}$$
$$\}$$
$$\}$$
$$\}$$

The latter can be bounded because $N < \infty$ and $1 - \bar{\rho}_j^{-2} > r$, $\forall j$, as long as $\bar{\rho} \in \mathcal{B}_N$ and η_{i-1} is bounded by induction hypothesis

□

6.2. Backward stability of D for computing $\rho(t)$

The technique is like in section 5.2.

For notational convenience, let us drop the upper bar :

$$\rho_i = fl(B_{0,i}/A_{0,i}) = \frac{\varepsilon^Q + fl(B_{1,i-1} - fl(A_{1,i-1}\rho_{i-1}))}{fl(A_{0,i-1} - fl(B_{0,i-1}\rho_{i-1}))}$$

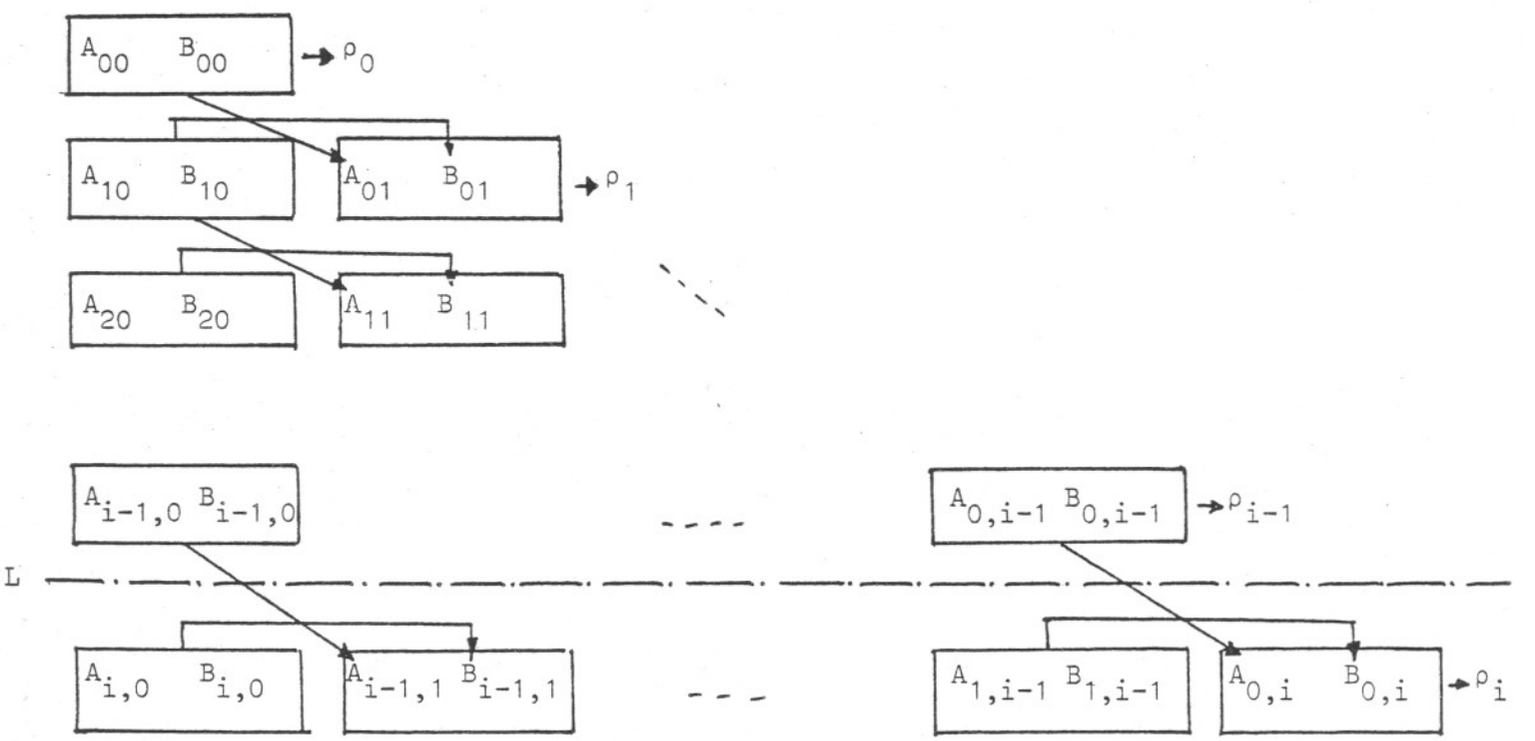
$$= \frac{\varepsilon^Q + \varepsilon_1^{SB} - \varepsilon_1^{PB} + B_{1,i-1} - A_{1,i-1}\rho_{i-1}}{\varepsilon_1^{SA} - \varepsilon_1^{PA} + A_{0,i-1} - B_{0,i-1}\rho_{i-1}} = \frac{\varepsilon^Q + \varepsilon_1^{SB} - \varepsilon_1^{PB} + \varepsilon^I + B_{1,i-1} - A_{1,i-1}\rho_{i-1}}{A_{0,i-1} - B_{0,i-1}\rho_{i-1}} \quad (20)$$

with $|\varepsilon^Q| \leq |B_{0,i}|\varepsilon$; $|\varepsilon_1^{SB}| \leq |B_{0,i}|\varepsilon$; $|\varepsilon_1^{PB}| \leq |A_{1,i-1}\rho_{i-1}|\varepsilon$

$|\varepsilon_1^{SA}| \leq |A_{0,i}|\varepsilon$; $|\varepsilon_1^{PA}| \leq |B_{0,i-1}\rho_{i-1}|\varepsilon$

$\varepsilon^I = -\rho_i(\varepsilon_1^{SA} - \varepsilon_1^{PA})$

To see how we have to go back through the computations, let us look at the following scheme (remember $A_{j,0} = t_j$; $B_{j,0} = t_{j+1}$)



Once we hit a number that is above the line L, then we do not have to go any further, since the rounding error which is in such a number has already been taken into account when determining $\varepsilon_j = \bar{t}_j - t_j$ for $j \leq i$.
So it is seen that we only have to analyse the numerator of (20) somewhat further.

$$n = \varepsilon^Q + \varepsilon_1 + fl(B_{2,i+2} - fl(A_{2,i-2}\rho_{i-2})) - \rho_{i-1}\, fl(A_{1,i-2} - fl(\rho_{i-2}B_{1,i-2}))$$

$$= \varepsilon^Q + \varepsilon_1 + \varepsilon_2^{SB} - \varepsilon_2^{PB} + (B_{2,i-2} - \rho_{i-2}\, A_{2,i-2}) - \rho_{i-1}(A_{1,i-2} - \rho_{i-2}B_{1,i-2} + \varepsilon_2^{SA} - \varepsilon_2^{PA})$$

with $\varepsilon_2^{SB} \leq |B_{1,i-1}|\varepsilon$; $|\varepsilon_2^{PB}| \leq |\rho_{i-2}\, A_{2,i-2}|\varepsilon$;

$\varepsilon_2^{SA} \leq |A_{1,i-1}|\varepsilon$; $|\varepsilon_2^{PA}| \leq |\rho_{i-2}\, B_{1,i-2}|\varepsilon$

$$\varepsilon_1 = \varepsilon_1^{SB} - \varepsilon_1^{PB} - \rho_i(\varepsilon_1^{SA} - \varepsilon_1^{PA}) = \varepsilon_1^{SB} - \varepsilon_1^{PB} + \varepsilon^I$$

$$n = \varepsilon^Q + \varepsilon_1 + \varepsilon_2 - \rho_{i-1}(A_{1,i-2} - \rho_{i-2}\, B_{1,i-2}) + z_2$$

$$\varepsilon_2 = \varepsilon_2^{SB} - \varepsilon_2^{PB} - \rho_{i-1}(\varepsilon_2^{SA} - \varepsilon_2^{PA}) \quad ; \quad z_2 = B_{2,i-2} - \rho_{i-2}\, A_{2,i-2}$$

z_2 is the only factor to be analysed further :

$$z_2 = B_{3,i-3} - \rho_{i-3}\, A_{3,i-3} + \varepsilon_3^{SB} - \varepsilon_3^{PB} - \rho_{i-2}(\varepsilon_3^{SA} - \varepsilon_3^{PA} + A_{2,i-3} - \rho_{i-3}\, B_{2,i-3})$$

$$= \varepsilon_3 + (B_{3,i-3} - \rho_{i-3}\, A_{3,i-3}) - \rho_{i-2}(A_{2,i-3} - \rho_{i-3}\, B_{2,i-3})$$

$$\varepsilon_3 = \varepsilon_3^{SB} - \varepsilon_3^{PB} - \rho_{i-2}(\varepsilon_3^{SA} - \varepsilon_3^{PA})$$

$$|\varepsilon_3^{SB}| \leq |B_{2,i-2}|\varepsilon \; ; |\varepsilon_3^{PB}| \leq |\rho_{i-3}\, A_{3,i-3}|\varepsilon \; ;$$

$$|\varepsilon_3^{SA}| \leq |A_{2,i-2}|\varepsilon \; ; |\varepsilon_3^{PA}| \leq |\rho_{i-3}\, B_{2,i-3}|\varepsilon$$

and we continue the analysis on $z_3 = B_{3,i-3} - \rho_{i-3}\, A_{3,i-3}$.
At the end, this results in

$$n = \varepsilon^T + T + (B_{i,0} - \rho_0\, A_{i,0})$$

where

$$T = - \sum_{j=1}^{i-1} \rho_{i-j} \left(A_{j,i-j-1} - \rho_{i-j-1} B_{j,i-j-1}\right)$$

$$B_{i,0} = t_{i+1} \quad \text{and} \quad A_{i,0} = t_i = \bar{t}_i - \Delta t_i$$

$$\varepsilon^T = \varepsilon^Q + \sum_{j=1}^{i} \varepsilon_j$$

Thus we find that

$$\rho_i = \frac{\varepsilon^T + \rho_0 \Delta t_i + t_{i+1} + T - \rho_0 \bar{t}_i}{A_{0,i-1} - B_{0,i-1} \rho_{i-1}}$$

and thus

$$\Delta t_{i+1} = \varepsilon^T + \rho_0 \Delta t_i$$

This may be bounded as follows:

For $j \geqslant 1$:

$$|\varepsilon_j| \leqslant (|B_{j-1,i-j+1}| + |A_{j,i-j}| + |A_{j-1,i-j+1}| + |B_{j-1,i-j}|)\varepsilon$$
$$\lesssim 2(j\, r_{i-j+1} + r_{i-j})\varepsilon$$

$$|\varepsilon^Q| \leqslant r_i\, \varepsilon$$

Thus

$$|\varepsilon^T| \leqslant r_i \varepsilon + 2\varepsilon \sum_{j=1}^{i} (j\, r_{i-j+1} + r_{i-j}) \leqslant (1 + 3i + i^2)\varepsilon$$

and

$$|\Delta t_{i+1}| \leqslant |\Delta t_i| + (1 + 3i + i^2)\varepsilon$$

which can be bounded, and thus we have

Theorem 6

Algorithm D, implementing a ladder form technique for the computation of the reflection coefficients is numerically stable on T_N.

□

7. Conclusion

We have shown that certain recursive techniques to compute factorizations of Toeplitz matrices or their inverses are unstable on the whole set of Toeplitz matrices. This includes recursive Padé approximation on a row or a sawtooth of the Padé table. Similar results may be obtained for Hankel matrices [23].

However, when restricting the input set to symmetric positive definite Toeplitz matrices, bounded away from singularity, then the former algorithms reduce to the well known Levinson algorithm or its continued fraction or ladder form analogue, and these are proved to be numerically stable on that set.

Stability of the factorization algorithms corresponding to these was shown by a mixed error analysis: Forward stability for the computation of the factors from the reflection coefficients and backward stability for the computation of the reflection coefficients from the Toeplitz data.

In so called ladder form realizations of digital filters, only the reflection coefficients are of interest. The backward stability analysis as above seems to indicate that the Levinson algorithm is critical on extreme examples, though satisfactory in practical situations, while its continued fraction form seems to be much more stable because the numbers involved in the computation are nicely bounded.

References

1.A.Bultheel: Recursive algorithms for the Padé table: two approaches. Proceedings Conference on Padé approximation and its applications, Antwerpen, april 4-6,1979. To appear Lecture notes in mathematics, Springer Verlag, Berlin , 1979.

2.W.F.Trench:An algorithm for the inversion of finite Toeplitz matrices. SIAM J.Appl.Math 12(1964),515-522.

3.S.Zohar: Toeplitz matrix inversion. The algorithm of W.F.Trench. J.Assoc.Comput.Mach.,16(1969),592-601.

4.J.Rissanen: Algorithm for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring positive matrix polynomials. Math. of Comp.,27(1973),147-154.

5.A.Bultheel: Fast factorization algorithms and Padé approximation. Submitted for publ. Numer Math.

6.J.R.Bunch-B.N.Parlett: Direct methods for solving symmetric indefinite systems of linear equations. SIAM J. Numer.Anal.,8(1971),639-655.

7.L.S.de Jong: Towards a formal definition of numerical stability. Numer. Math. 28(1977),211-219.

8.N.Levinson: The Wiener RMS (root mean square) error criterion in filter design and prediction. J.Mathematical Phys.,25(1947),261-278.

9.J.Makhoul: Linear prediction: a tutorial review. Proceedings of the IEEE,63(1975),561-580.

10.G.Szegö: Orthogonal polynomials. AMS Colloquium publ.XXIII,AMS Providence,Rhode Island,1939.

11.L.Y.Geronimus: Orthogonal polynomials. Consultant bureau, New York, 1961.

12.J.Schur: Uber Potenzreihen die im Innern des Einheitskreises beschränkt sind. Z.Reine Angew.Math.,147(1917),205-232 and 148(1918),122-145.

13.P.Delsarte,Y.Genin,Y.Kamp: Schur parametrization of positive block-Toeplitz systems. SIAM J. on Appl. Math.,36(1979),34-46.

14.W.J.Thron: Two-point Padé tables,T-fractions and sequences of Schur. in: E.B.Saff-R.S.Varga (eds.): Padé and rational approximation. Acad.Press,New York,1977,163-172.

15.P.Dewilde,A.Vieira,T.Kailath: On a generalized Szegö-Levinson realization algorithm for optimal linear predictors based on a network synthesis approach. IEEE Transactions on Circuits and Systems. CAS-25(1978),663-675.

16.W.B.Gragg: Laurent-,Fourier- and Chebyshev Padé tables. in: E.B.Saff-R.S.Varga (eds.): Padé and rational approximation. Academic Press,New York,1977,61-72.

17.M.Morf, A.Vieira,D.T.Lee: Ladder forms for identification and speech processing. Proc.Confer. on Decision and Control, New York Louisiana, dec. 7-9,1977.

18.I.Hirschman Jr.,D.E.Hughes: Extreme eigenvalues of Toeplitz operators. Lecture notes in mathematics 618, Springer Verlag, Berlin, 1977.

19.J.H.Wilkinson: The algebraic eigenvalue problem. Oxford University press, 1965.

20.G.J.Bierman: Factorization methods for discrete sequential estimation. Academic Press, New York, 1977.

21.R.Fletcher-M.J.D.Powell: On the modification of LDL^t factorizations. Math.of Comput.,28(1974),1067-1087.

22.A.Bultheel: Fast algorithms for triangular factorization of matrices. Presented at the workshop on fast algorithms, Louvain la Neuve, march 10,1978; unpublished.

23.L.S.de Jong: Numerical aspects of recursive realization algorithms. SIAM J. Control and Optimization, 16(1978),646-659.

24.G.W.Stewart: The effect of rounding error on an algorithm for downdating a Cholesky factorization. Jour. Inst.Math.Appl.,23(1979),203-213.

25.K.W.Brodlie,A.R.Gourlay,J.Greenstadt: Rank-one and rank-two corrections to positive definite matrices expressed in product form. Journ. Inst.Math.Appl.,11(1973),73-82.