A BLOCK TOEPLITZ LOOK-AHEAD SCHUR ALGORITHM

KYLE GALLIVAN
Coordinated Science Laboratory
University of Illinois
Urbana, IL USA
gallivan@csrd.uiuc.edu

SRIKANTH THIRUMALAI
Coordinated Science Laboratory
University of Illinois
Urbana, IL USA
srikanth@csrd.uiuc.edu

PAUL VAN DOOREN
Cesame
Université Catholique de Louvain
Louvain-la-Neuve
Belgium
vandooren@anma.ucl.ac.be

ABSTRACT. This paper gives a look-ahead Schur algorithm for finding the symmetric factorization of a Hermitian block Toeplitz matrix. The method is based on matrix operations and does not require any relations with orthogonal polynomials. The simplicity of the matrix based approach ought to shed new light on other issues such as parallelism and numerical stability.

KEYWORDS. Block Toeplitz matrix, Schur algorithm, numerical methods, look-ahead.

Introduction

The Schur algorithm yields a method to compute the symmetric decomposition

$$T = U^* D U, \quad U \text{ upper-triangular} \tag{1}$$

of an $(n+1) \times (n+1)$ Hermitian Toeplitz matrix

$$T = \begin{bmatrix} t_0 & t_1 & \cdots & t_n \\ \bar{t}_1 & t_0 & \ddots & t_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ \bar{t}_n & \bar{t}_{n-1} & \dots & t_0 \end{bmatrix} \tag{2}$$

in $O(n^2)$ operations [8]. This algorithm actually derives this decomposition for all leading principal submatrices as well via a simple vector recurrence, which explains the low complexity of the method. Another well known algorithm for the same problem is the Levinson algorithm [10]. Yet, the Schur algorithm has gained a lot of popularity over the Levinson algorithm for various reasons : (i) it is known to be better suited for fine grain parallelism [9], (ii) it constructs the factor U in (1) directly, rather than its inverse as in the Levinson algorithm [8], (iii) it exploits better matrix properties such as bandedness and low rank [5] and (iv) it has been shown to have better numerical properties for positive definite T [1].

Both algorithms, though, are known to be potentially unstable when T is indefinite. This is the case when the leading principal minors of T are (nearly) singular, since both algorithms implicitly use these submatrices in their recurrence. Remedies for this were proposed for the Schur algorithm [11] and for the Levinson algorithm [2] and were essentially based on a look-ahead technique, whereby one "jumps" over the singular submatrices. Although this requires a slight increase in complexity, this is in general quite an effective technique. These techniques are linked to the theory of orthogonal polynomials and can become quite involved in the case of look-ahead [6], [3]. In this paper we present a matrix based derivation of such a look-ahead method and give the algorithm directly for block Toeplitz matrices. This extension is quite easy because of the use of matrix manipulations rather than orthogonal polynomials.

1 Schur complements and displacement rank

Let T be a general Hermitian, block Toeplitz matrix of dimension $N \times N$ and block size $m \times m$, i.e.

$$T = \begin{bmatrix} T_0 & T_1 & \cdots & T_r \\ T_1^* & T_0 & \ddots & T_{r-1} \\ \vdots & \ddots & \ddots & \vdots \\ T_r^* & T_{r-1}^* & \dots & T_0 \end{bmatrix}, \qquad T_0 = T_0^*, \qquad N = m \times (r+1). \tag{3}$$

The purpose here is to find a factorization as in (1) where D is diagonal or block diagonal and U is upper triangular or upper block triangular. Schur type algorithms are based on the concept of *displacement*, which is defined as follows. Choose Z to be the block right shift matrix of the same dimension :

$$Z = \begin{bmatrix} 0_m & I_m & & \\ & \ddots & \ddots & \\ & & \ddots & I_m \\ & & & 0_m \end{bmatrix} . \quad (4)$$

The *displacement rank* of the matrix T is then defined as :

$$\alpha = rank \; (\; T - Z^*TZ \;) \leq 2m \; , \quad (5)$$

and the *displacement* of the matrix T can therefore be factored as :

$$T - Z^*TZ = G^*\Sigma G \quad (6)$$

where the $\alpha \times \alpha$ matrix Σ equals

$$\Sigma = \left[\begin{array}{c|c} I_p & \\ \hline & -I_q \end{array} \right] , \qquad p, q \leq m. \quad (7)$$

This factorization can be automatically written for block Toeplitz matrices. For arbitrary matrices satisfying (6) with $\alpha \ll N$, the factorization can be obtained from the Bunch Kaufman decomposition or also from the eigen decomposition of $T - Z^*TZ$. Matrices with such a low displacement rank are said to be quasi block Toeplitz. The complexity of this preliminary decomposition is normally O(αN^2).

It is well known that factorizations of the type (1) are working on Schur complements of the original matrix at each stage of their recurrence. We now derive updating formulas for the Schur complement of a matrix T with low displacement rank, and show that it also has low displacement rank. This part is related to the work of [7] as was pointed out to us, but is not contained in it. Partition T and Z conformally as :

$$T = \left[\begin{array}{c|c} T_{11} & T_{12} \\ \hline T_{21} & T_{22} \end{array} \right] , \qquad Z = \left[\begin{array}{c|c} Z_{11} & Z_{12} \\ \hline 0 & Z_{22} \end{array} \right] , \quad (8)$$

where T_{11} and Z_{11} are of dimension $mk \times mk$ (a multiple of the block size) and T_{11} is assumed to be invertible (this is always possible by choosing k large enough). Define

$$X = T_{11}^{-1} T_{12} \; , \qquad X^* = T_{12}^* T_{11}^{-1} \; , \qquad U = \left[\begin{array}{c|c} I & -X \\ \hline & I \end{array} \right] , \quad (9)$$

then it follows that

$$U^*TU = \left[\begin{array}{c|c} T_{11} & \\ \hline & T_{sc} \end{array} \right] , \qquad T_{sc} = T_{22} - T_{12}^* T_{11}^{-1} T_{12} \; , \quad (10)$$

where T_{sc} is the Schur complement of T with respect to T_{11}. Applying $U^*(\; . \;)U$ to (6) yields :

$$U^*TU \; - \; (U^*Z^*U^{-*}) \; U^*TU \; (U^{-1}ZU) \; = \; U^*G^*\Sigma GU \; . \quad (11)$$

Notice that

$$U^{-1}ZU = \left[\begin{array}{c|c} Z_{11} & \hat{Z}_{12} \\ \hline & Z_{22} \end{array} \right] , \qquad \hat{Z}_{12} = \left[\; I \mid X \; \right] Z \left[\begin{array}{c} -X \\ \hline I \end{array} \right] . \quad (12)$$

Using (10) and (12) we can reduce (11) to :

$$\left[\begin{array}{c|c} T_{11} & \\ \hline & T_{sc} \end{array}\right] - \left[\begin{array}{c|c} Z_{11}^* & \\ \hline \hat{Z}_{12}^* & Z_{22} \end{array}\right] \left[\begin{array}{c|c} T_{11} & \\ \hline & T_{sc} \end{array}\right] \left[\begin{array}{c|c} Z_{11} & \hat{Z}_{12} \\ \hline & Z_{22} \end{array}\right] = U^* G^* \Sigma G U \ . \tag{13}$$

Equating the $(1,2)$ and $(2,2)$ positions in the above equation we have

$$\begin{aligned} M &= Z_{11}^* T_{11} \hat{Z}_{12} + \left[\begin{array}{c|c} I & 0 \end{array}\right] G^* \Sigma G \left[\begin{array}{c} -X \\ \hline I \end{array}\right] = 0 \ , \\ \Delta T_{sc} &= T_{sc} - Z_{22}^* T_{sc} Z_{22} = \hat{Z}_{12}^* T_{11} \hat{Z}_{12} + \left[\begin{array}{c|c} -X^* & I \end{array}\right] G^* \Sigma G \left[\begin{array}{c} -X \\ \hline I \end{array}\right] \ . \end{aligned} \tag{14}$$

Substituting for $\hat{Z}_{12}$ from (12) we can further simplify M and ΔT_{sc} to :

$$M = \left[\begin{array}{c|c} I & 0 \end{array}\right] \left\{ Z^* \left[\begin{array}{c} I \\ \hline X^* \end{array}\right] T_{11} \left[\begin{array}{c|c} I & X \end{array}\right] Z + G^* \Sigma G \right\} \left[\begin{array}{c} -X \\ \hline I \end{array}\right] = 0 \ , \tag{15}$$

$$\Delta T_{sc} = \left[\begin{array}{c|c} -X^* & I \end{array}\right] \left\{ Z^* \left[\begin{array}{c} I \\ \hline X^* \end{array}\right] T_{11} \left[\begin{array}{c|c} I & X \end{array}\right] Z + G^* \Sigma G \right\} \left[\begin{array}{c} -X \\ \hline I \end{array}\right] \ . \tag{16}$$

Substituting for X in the matrix in the middle of the above equations we get

$$\begin{aligned} W &= Z^* \left[\begin{array}{c} T_{11}^* \\ \hline T_{12}^* \end{array}\right] T_{11}^{-1} \left[\begin{array}{c|c} T_{11} & T_{12} \end{array}\right] + G^* \Sigma G \\ &= \left[\begin{array}{c|c} Z^* \left[\begin{array}{c} T_{11}^* \\ \hline T_{12}^* \end{array}\right] & G^* \end{array}\right] \left[\begin{array}{c|c} T_{11}^{-1} & 0 \\ \hline 0 & \Sigma \end{array}\right] \left[\begin{array}{c} \left[\begin{array}{c|c} T_{11} & T_{12} \end{array}\right] Z \\ \hline G \end{array}\right] \ . \end{aligned} \tag{17}$$

This expression can now be further simplified to prove that the rank of ΔT_{sc} is at most α. In order to prove this we first need the following lemma.

Lemma 1 *Let*

$$W = \begin{bmatrix} F_{11}^* & F_{21}^* \\ F_{12}^* & F_{22}^* \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{18}$$

where Σ_1 and $W_{11} = F_{11}^* \Sigma_1 F_{11} + F_{21}^* \Sigma_2 F_{21}$ are invertible. Then there always exists a transformation H such that

$$H^* \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \end{bmatrix} H = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \tag{19}$$

$$H \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} \hat{F}_{11} & \hat{F}_{12} \\ 0 & \hat{F}_{22} \end{bmatrix} \tag{20}$$

Proof. Let $H = RQ$, where R is block upper triangular and Q is unitary. We choose Q such that

$$Q\begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tag{21}$$

where B is upper triangular. Moreover, since W_{11} is assumed invertible, $\begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix}$ is full rank and hence B is invertible as well. Let R be partitioned conformally with W as

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} , \tag{22}$$

then H automatically satisfies (20) and $R_{11}B = \hat{F}_{11}$. Also, H will satisfy (19) if and only if

$$\begin{bmatrix} R_{11}^* & 0 \\ R_{12}^* & R_{22}^* \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = Q \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} Q^* , \tag{23}$$

where the right hand side is now known. A decomposition of the type $R^*\tilde{\Sigma}R$ is known to exist iff the (1,1) block of the right hand side is invertible. Because of (21) this equals $B^{-*}(F_{11}^*\Sigma_1 F_{11} + F_{21}^*\Sigma_2 F_{21})B^{-1}$ which is invertible. $\square$

In order to simplify (17) we now want to apply this lemma to construct a transformation H such that

$$H^* \left[\begin{array}{c|c} \tilde{T}_{11}^{-1} & \\ \hline & \tilde{\Sigma} \end{array}\right] H = \left[\begin{array}{c|c} T_{11}^{-1} & \\ \hline & \Sigma \end{array}\right] , \tag{24}$$

$$H \left[\begin{array}{c} \begin{bmatrix} T_{11} & T_{12} \end{bmatrix} Z \\ \hline G \end{array}\right] = \left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \\ \hline 0 & \hat{G}_2 \end{array}\right] , \tag{25}$$

where $\tilde{T}_{11}$ and $\hat{T}_{11}$ are matrices of size $mk \times mk$, G has dimensions $\alpha \times N$ and $\hat{G}_2$ has dimensions $\alpha \times (N - mk)$. In order to apply the above lemma we only need to show that W_{11} is invertible since T_{11} is invertible by assumption. From (13) it follows that

$$T_{11} = Z_{11}^* T_{11} Z_{11} + G_1^* \Sigma G_1 , \qquad \text{where} \qquad G_1 = G \left[\begin{array}{c} I \\ \hline 0 \end{array}\right] . \tag{26}$$

From (17), W_{11} equals

$$W_{11} = \left[\begin{array}{c|c} I & 0 \end{array}\right] \left\{ Z^* \left[\begin{array}{c} T_{11}^* \\ \hline T_{12}^* \end{array}\right] T_{11}^{-1} \left[\begin{array}{c|c} T_{11} & T_{12} \end{array}\right] + G^*\Sigma G \right\} \left[\begin{array}{c} I \\ \hline 0 \end{array}\right] \tag{27}$$

and since

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix} , \qquad G = \begin{bmatrix} G_1 & G_2 \end{bmatrix} , \tag{28}$$

we have

$$W_{11} = Z_{11}^* T_{11} T_{11}^{-1} T_{11} Z_{11} + G_1^* \Sigma G_1 = T_{11} , \tag{29}$$

which thus shows that W_{11} is invertible as well.

Applying (24) and (25) to (17) we obtain

$$W = \left[\begin{array}{c|c} \hat{T}_{11}^* & 0 \\ \hline \hat{T}_{12}^* & \hat{G}_2^* \end{array}\right] \left[\begin{array}{c|c} \tilde{T}_{11}^{-1} & 0 \\ \hline 0 & \tilde{\Sigma} \end{array}\right] \left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \\ \hline 0 & \hat{G}_2 \end{array}\right]. \tag{30}$$

Inserting this in (15) and (16) yields

$$\begin{aligned} M &= \left[\begin{array}{c|c} I & 0 \end{array}\right] W \left[\begin{array}{c} -X \\ \hline I \end{array}\right] = \hat{T}_{11}^* \tilde{T}_{11}^{-1} \left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \end{array}\right] \left[\begin{array}{c} -X \\ \hline I \end{array}\right] = 0 \\ \Delta T_{sc} &= \hat{G}_2^* \tilde{\Sigma} \hat{G}_2 + \left[\begin{array}{c|c} -X^* & I \end{array}\right] \left[\begin{array}{c} \hat{T}_{11}^* \\ \hline \hat{T}_{12}^* \end{array}\right] \tilde{T}_{11}^{-1} \left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \end{array}\right] \left[\begin{array}{c} -X \\ \hline I \end{array}\right]. \end{aligned} \tag{31}$$

Since $M = 0$ and $\hat{T}_{11}$ and $\tilde{T}_{11}$ are invertible, we have $\left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \end{array}\right] \left[\begin{array}{c} -X \\ \hline I \end{array}\right] = 0$, which yields,

$$\Delta T_{sc} = \hat{G}_2^* \tilde{\Sigma} \hat{G}_2. \tag{32}$$

This establishes a new displacement identity where $\tilde{\Sigma}$ and $\hat{G}_2$ are obtained from (24-25).

2 Block look-ahead algorithm

The above construction thus suggests the following algorithm for block Toeplitz matrices.

Algorithm 1 ***Block-Toeplitz***

Construct generator $\Sigma_{(0)}$, $G_{(0)}$

Use Bunch Kaufman on T_0 *to obtain* $T_0 = U_0^* \Sigma_0 U_0$ *and define*

$$\Sigma_{(0)} \doteq \begin{bmatrix} \Sigma_0 & 0 \\ 0 & -\Sigma_0 \end{bmatrix}, \quad G_{(0)} \doteq \begin{bmatrix} U_0 & \Sigma_0 U_0^{-*} T_1 & \dots & \Sigma_0 U_0^{-*} T_r \\ 0 & \Sigma_0 U_0^{-*} T_1 & \dots & \Sigma_0 U_0^{-*} T_r \end{bmatrix}, \quad i = 1$$

Apply block Schur algorithm

while $G_{i-1} \neq$ ***void***

Construct leading rows $[T_{11}|T_{12}]$ *from* $\Sigma_{(i-1)}$, $G_{(i-1)}$ *until* T_{11} *is well conditioned*

Compute $U_i = T_{11}^{-1}\,[T_{11}|T_{12}] = [I \mid X]$

Append T_{11} *to* D *and* U_i *to* U

Apply a transformation H *satisfying*

$$H^* \begin{bmatrix} \tilde{T}_{11}^{-1} & 0 \\ 0 & \Sigma_{(i)} \end{bmatrix} H = \begin{bmatrix} T_{11}^{-1} & 0 \\ 0 & \Sigma_{(i-1)} \end{bmatrix}$$

such that

$$H \left[\begin{array}{c} \left[\begin{array}{cc} T_{11} & T_{12} \end{array}\right] Z \\ \hline G_{(i-1)} \end{array}\right] = \left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \\ \hline 0 & G_{(i)} \end{array}\right]$$

Increment i

end while

This algorithm is of course only conceptual. It does not describe how to construct the transformation H nor how to track the condition number of T_{11}. For the latter we refer to techniques as those described in [2, 6, 3]. For the construction of the Σ-unitary transformations H we can use skew Householder transformations of block versions of them (see [4]). In [4] issues of efficient parallel implementation of such transformations are also adressed. We point out that when T_{11} is well conditioned then the transformation H and its construction should give no numerical problems. It should be pointed out that the first m columns of $\left[\begin{array}{cc} T_{11} & T_{12} \end{array}\right] Z$ are zero which can be exploited in the factorization

$$H \left[\begin{array}{c} \left[\begin{array}{cc} T_{11} & T_{12} \end{array}\right] Z \\ \hline G_{(i-1)} \end{array}\right] = \left[\begin{array}{c|c} \hat{T}_{11} & \hat{T}_{12} \\ \hline 0 & G_{(i)} \end{array}\right]$$

This is especially the case when there is no look-ahead needed (i.e. when $k = 1$). The above matrix has then $3m$ rows of which only $2m$ have to be processed. One checks that this economical version is precisely the usual block Toeplitz algorithm without look-ahead. The complexity of this method is $O(m^2N^2)$ in the best case (i.e. without need for look-ahead). With look-ahead of moderate size this will increase slightly as a function of k.

Finally, notice also that the results presented in this paper can be extended to the non Hermitian case, provided two generator are kept and updated instead of one. For simplicity, we did not develop this here.

Acknowledgments

We are indebted to V. Olshevsky for pointing out reference [7] to us. This research was supported by the National Science Foundation under Grants CCR 9209349 and CCR9120008 and by ARPA under Grant 60NANB2D1272

References

[1] A. Bojanczyk, R. Brent, F. de Hoog and D. Sweet. On the Stability of the Bareiss and Related Toeplitz Factorization Algorithms *SIAM J. Matrix Anal. Appl.* **15**, pp , 1994.

[2] T. Chan and P.C. Hansen. A look-ahead Levinson algorithm for indefinite Toeplitz systems. *SIAM J. Matrix Anal. Appl.* **13**, pp 490-506, 1992.

[3] R. Freund and H. Zha. Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems. *Linear Algebra and its Applications* **188-189**, pp 255-303, 1993.

[4] K. Gallivan, S. Thirumalai and P. Van Dooren. On solving block Toeplitz systems using a block Schur algorithm. in *Intern. Conf. Parallel Processing*, Proc. ICPP-94, St. Charles IL, pp III-274–III-281, 1994.

[5] K. Gallivan, S. Thirumalai and P. Van Dooren. Regularization Teoplitz least squares problems. Int. Rept. CSL, Univ. Illinois at Urbana-Champaign, 1994.

[6] M. Gutknecht. Stable row recurrences for the Padé table and generically superfast look-ahead solvers for non-Hermitian Toeplitz systems. *Linear Algebra and its Applications* **188-189**, pp 351-421, 1993.

[7] T. Kailath and A. Sayed. Fast algorithms for generalized displacement structrures. in *Recent Advances in Mathematical Theory of Systems* (H. Kimura, S. Kodoma, Eds.), Proc. MTNS-91, pp 27-32, 1992.

[8] T. Kailath, A. Vieira and M. Morf. Inverses of Toeplitz operators, innovations and orthogonal polynomials. *SIAM Review* **20**, pp 106-119, 1978.

[9] S.-Y. Kung and Y. Hu. A highly concurrent algorithm and pipelined architecture for solving Toplitz systems. *IEEE Trans. ASSP* **31**, pp 66-76, 1983.

[10] N. Levinson. The Wiener RMS (root-mean-square) error criterion in filter design and prediction. *J. Math. Phys.* **25**, pp 261-278, 1947.

[11] D. R. Sweet. The use of pivoting to improve the numerical performance of Toeplitz solvers. in *Advanced Algorithms and Architectures for Signal Processing* (J.M. Speiser, Ed.), Proc. SPIE 696, pp 8-18, 1986.