- $\boldsymbol{X}_1 = \gamma \boldsymbol{I}_n$: The 2–norm matrix approximation for computing $\hat{\boldsymbol{X}}_2$ such that $\|\boldsymbol{X}_2 - \hat{\boldsymbol{X}}_2\|_2 < \gamma$ [13, 8, 6]. The presented Schur–type method uses a preparatory QRD of $\boldsymbol{X}_2$ and applies Algorithm 2 to $[\gamma \boldsymbol{I}_n \;\; \boldsymbol{B}^T]^T$ using only n hyperbolic rotations, while in [13, 8, 6] the required number of hyperbolic rotations is not well defined such that it is difficult to elaborate the numerical behavior of these methods.
  Using $\boldsymbol{X}_1 = \varepsilon \boldsymbol{I}$ the Schur algorithm can be applied to condition estimation or ε–rank decision to determine if an iterative refinement step is worthwhile for the least squares problem $\|\boldsymbol{X}_2 \boldsymbol{y} - \boldsymbol{b}\|_2 \to \min$.

Remark 4 Extension to more than two matrices:

- The triangularization of a $m \times n$ Toeplitz matrix $\boldsymbol{X}$ can be expressed as a generalized Schur algorithm [10, 11]. From $\boldsymbol{X}$ one obtains upper triangular matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{B}_1$, and $\boldsymbol{B}_2$ such that
$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} \begin{matrix} + \\ + \\ - \\ - \end{matrix} . \qquad (18)$$
  The straightforward extension of the above generalized Schur–type method applied to $\boldsymbol{P}$ yields the upper triangular Cholesky factor $\boldsymbol{R}$ of $\boldsymbol{X}^T\boldsymbol{X}$ in $O(n^2)$ operations.

5. CONCLUSIONS

Based on a subspace criteria Schur–type methods requiring a minimal number of hyperbolic rotations were derived. By using a minimal number of hyperbolic rotations it is possible to relate the hyperbolic rotations to the implicit downdates of standard matrix decompositions. Therefore, the numerical properties (e.g. breakdown) of the Schur–type methods can be deduced from the properties of the known matrix decomposition algorithms. Various algorithms which emerged independently in linear algebra and signal processing are shown to be essentially variants of the presented Schur–type methods based on subspace criteria (see remarks 3 and 4).

REFERENCES

[1] J. Chun, T. Kailath, H. Lev-Ari. Fast parallel algorithms for QR and triangular factorization. *SIAM J. Sci. Stat. Comput.*, 8:899–913, 1987.

[2] J.M. Delosme, I.C.F. Ipsen. Parallel solution of symmetric positive definite systems with hyperbolic rotations. *Linear Alg. and Its Appl.*, 77:75–111, 1986.

[3] K. Diepold, R. Pauli. A recursive algorithm for lossless embedding of non–passive systems. In P. Dewilde, M.A. Kaashoek, and M. Verhaegen, editors, *Challenges of a Generalized System Theory.*, pages 209–222. North–Holland, 1993.

[4] K.V. Fernando, B.N. Parlett. Accurate singular values and differential qd algorithm. *Numer. Math.*, 67:191–229, 1994.

[5] G.H. Golub, C.F. van Loan. *Matrix Computations.* The John Hopkins University Press, second edition, 1989.

[6] J. Götze, M. Haardt, J.A. Nossek. Subspace estimation using unitary Schur–type methods. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit (USA), 1153–1156, 1995.

[7] J. Götze, R. Pauli, U. Schwiegelshohn. Hyperbolic rotation of subspaces for signal processing. *AEÜ*, 44:92–96, 1990.

[8] J. Götze, A.J. van der Veen. On–line subspace estimation using a Schur–type method. *IEEE Trans. on Signal Processing.* 44:1585–1589, 1996

[9] G. Heinig, K. Rost. *Algebraic methods for Toeplitz–like matrices and operators.* Operator Theory: Advances and Applications, OT13. Birkhäuser Verlag, 1985.

[10] H. Park, L. Eldén. Stability analysis and fast algorithms for triangularization of Toeplitz matrices. *Numerische Mathematik*, to appear.

[11] H. Park, L. Eldén. Fast solutions for least squares problems with Toeplitz structure. submitted to a journal,February, 1996.

[12] I. Schur. Über Potenzreihen, die im Inneren des Einheitskreises beschränkt sind. I. *J. für reine und angewandte Mathematik*, 147:205–232, 1917.

[13] A.J. van der Veen. A Schur method for low–rank matrix approximation. *SIAM J. Matrix Anal. Appl.*, 17:139–160, 1996.

[14] U. von Matt. The orthogonal qd-algorithm. Technical Report CS–TR–3211, Dept. of Computer Science, University of Maryland, College Park, 1994.

[15] T. Kailath, A.H. Sayed. Displacement Structure: Theory and Applications. *SIAM Review*, 37:297-386, 1995.

[16] C.J. Zarovski. Schur algorithms for hermitian Toeplitz, and Hankel matrices with singular leading principal submatrices. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 39:2464–2480, 1991.

Appendix

Algorithm 1 (Schur algorithm (algebraic))
Initialization: $a_k^0 = a_k$, $b_k^0 = b_k$

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

$$\rho_i \leftarrow \frac{b_0^i}{a_0^i}; \quad \boldsymbol{H}_i \leftarrow \frac{1}{\sqrt{1-\rho_i^2}} \begin{bmatrix} 1 & -\rho_i \\ -\rho_i & 1 \end{bmatrix}$$

$$\begin{bmatrix} a_0^{i+1} & a_1^{i+1} & \cdots \\ 0 & b_1^{i+1} & \cdots \end{bmatrix} \leftarrow \boldsymbol{H}_i \begin{bmatrix} a_0^i & a_1^i & \cdots \\ b_0^i & b_1^i & \cdots \end{bmatrix}$$

$$\begin{bmatrix} a_0^{i+1} & a_1^{i+1} & \cdots \\ b_0^{i+1} & b_1^{i+1} & \cdots \end{bmatrix} \leftarrow \begin{bmatrix} a_0^{i+1} & a_1^{i+1} & \cdots \\ b_1^{i+1} & b_2^{i+1} & \cdots \end{bmatrix}$$

endfor

Algorithm 2 (Generalized Schur algorithm)
Given $\boldsymbol{P} = \left[\boldsymbol{A}^T \; \boldsymbol{B}^T\right]^T$ *as computed from* $\boldsymbol{S}$ *using (9) this algorithm computes* $[\boldsymbol{\Sigma}_1 \; \boldsymbol{\Sigma}_2]^T \boldsymbol{R}$*. For the positive definite case* $|\rho_i| < 1$ *holds for all* i *and one obtains* $\boldsymbol{\Sigma}_1 = \boldsymbol{I}_n$ *and* $\boldsymbol{\Sigma}_2 = \boldsymbol{O}_n$.

for $i = 1 : n$
  $\rho_i = p_{i+n,i}/p_{ii}$
  if $|\rho_i| < 1$, $\rho \leftarrow \rho_i$; $k = n$; **endif** %*(pos. def.)*
  if $|\rho_i| > 1$, $\rho \leftarrow 1/\rho_i$; $k = 0$; **endif**
  $\boldsymbol{H} \leftarrow \frac{1}{\sqrt{1-\rho^2}} \begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$
  $\boldsymbol{P} \leftarrow \boldsymbol{\Theta}_{i,i+n}(\boldsymbol{H}) \cdot \boldsymbol{P}$
  for $j = i+1 : n$
    $\tau \leftarrow p_{k+i,j}/p_{k+j,j}$
    $\boldsymbol{G} \leftarrow \frac{1}{\sqrt{1+\tau^2}} \begin{bmatrix} 1 & -\tau \\ \tau & 1 \end{bmatrix}$
    $\boldsymbol{P} \leftarrow \boldsymbol{\Theta}_{k+i,k+j}(\boldsymbol{G}) \cdot \boldsymbol{P}$
  endfor
endfor

i.e.,

$$\mathbf{\Theta}\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}\begin{matrix}+\\-\end{matrix} = \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}\begin{matrix}+\\-\end{matrix} = \begin{bmatrix}\boldsymbol{I}_n\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{R}, \qquad (11)$$

where

$$\mathbf{\Theta} = \prod_{k=n}^{n}\mathbf{\Theta}_{k,s_n}(\boldsymbol{H}_{nk})\cdots\prod_{k=2}^{n}\mathbf{\Theta}_{k,s_2}(\boldsymbol{H}_{2k})\prod_{k=1}^{n}\mathbf{\Theta}_{k,s_1}(\boldsymbol{H}_{1k}), \qquad (12)$$

$\boldsymbol{R}$ is upper triangular and $\boldsymbol{S} = \boldsymbol{R}^T\boldsymbol{R}$. According to

$$\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{R}^T\boldsymbol{R} - \boldsymbol{O}^T\boldsymbol{O},$$

we have indicated by the signatures in (10) and (11) that the rows of $\boldsymbol{A}$ and $\boldsymbol{B}$ (resp. $\boldsymbol{R}$ and $\boldsymbol{O}$) are in a positive subspace and a negative subspace, respectively. Since all transformations combine the rows of $\boldsymbol{A}$ with the rows of $\boldsymbol{B}$, i.e., the rows in the positive subspace with the rows in the negative subspace, all rotations are of the hyperbolic type. The difference in computing $\mathbf{\Theta}$ in (6) and (12) is that in (12), since $\boldsymbol{A}$ and $\boldsymbol{B}$ are not Toeplitz, the $n(n+1)/2$ hyperbolic rotations that are computed for annihilating the $n(n+1)/2$ elements of $\boldsymbol{B}$ are all different. This corresponds to the algorithm presented in [2].

It is possible, however, to minimize the number of hyperbolic rotations, by annihilating the matrix elements in different order and planes. For that purpose the matrix elements of $\boldsymbol{B}$ are annihilated row–by–row, where only the diagonal elements b_{ii} of $\boldsymbol{B}$ are annihilated by elementary $\boldsymbol{J}$–orthogonal transformations $\mathbf{\Theta}_{i,i+n}(\boldsymbol{H}_i)$. The remaining elements b_{ij} ($j = i+1 : n$) of the i–th row can be annihilated by elementary orthogonal transformations $\mathbf{\Theta}_{i+n,j+n}(\boldsymbol{G})$. Here the i–th row of $\boldsymbol{B}$ is combined with the j–th row of $\boldsymbol{B}$. Since both rows are in the negative subspace, orthogonal transformations can be used. This is summarized in Algorithm 2,where $|\rho_i| < 1$ for all i for positive definite cases (see appendix).

If the minimal number of hyperbolic rotations is used, the n hyperbolic rotations can be related to the n implicit downdates that appear in computing the diagonal elements of the resulting matrix $\boldsymbol{R}$ in the standard algorithm for computing the Cholesky decomposition [5]. Therefore, as there is no mathematical breakdown of the Cholesky decomposition for symmetric positive definite matrices, the same holds for the generalized Schur algorithm, which uses the minimum number of hyperbolic rotations. The possibility of a breakdown for general matrices is reduced to the breakdown of the n hyperbolic rotations.

4.2. Symmetric Indefinite Matrices

The main difference between the symmetric positive definite case and the symmetric indefinite case is that the generation of zeros is no longer restricted to the negative subspace, i.e., to matrix elements in the lower n rows ($\boldsymbol{B}$) of $\boldsymbol{P}$. If the $\boldsymbol{J}$–orthogonal transformation $\mathbf{\Theta}_{i,i+n}(\boldsymbol{H}_i)$ for annihilating b_{ii} does not exist ($|\rho_i| > 1$), $\boldsymbol{H}_i$ is computed so that a_{ii} is annihilated (i.e., the zero is generated in the positive subspace by using $\rho = 1/\rho_i$ instead of $\rho = \rho_i$). Then the remaining elements a_{ij} ($j = i+1 : n$) of the i–th row of $\boldsymbol{A}$ are annihilated using orthogonal transformations $\mathbf{\Theta}_{i,j}(\boldsymbol{G})$ acting in the positive subspace only. Therefore, the only difference from the positive definite case is that $|\rho_i| > 1$ can occur and that in this case a zero row is generated in the positive subspace instead of the negative subspace. This possibility is already included in Algorithm 2 (see appendix).

In general, we obtain the following factorization

$$\mathbf{\Theta}\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}\begin{matrix}+\\-\end{matrix} = \begin{bmatrix}\mathbf{\Sigma}_1\\ \mathbf{\Sigma}_2\end{bmatrix}\boldsymbol{R}, \qquad (13)$$

where $\mathbf{\Sigma}_1 = \text{diag}[\sigma_{1i}]$, $\mathbf{\Sigma}_2 = \text{diag}[\sigma_{2i}]$ are $n \times n$ diagonal matrices with zeros and ones as their diagonal elements. The total number of ones on the diagonals of $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ is exactly n, and $\sigma_{1i} \neq \sigma_{2i}$ holds for all i. Therefore, we obtain

$$\boldsymbol{S} = \boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}, \qquad (14)$$

where

$$\boldsymbol{D} = \mathbf{\Sigma}_1^T\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2^T\mathbf{\Sigma}_2,$$

i.e., the $\boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}$–decomposition of $\boldsymbol{S}$ with signature matrix $\boldsymbol{D}$.

Again the n hyperbolic rotations correspond to the n implicit downdates in the standard algorithm for computing the $\boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}$–decomposition. The standard algorithm fails when the $\boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}$–decomposition of $\boldsymbol{S}$ does not exist. In the Schur–type algorithm this leads to a Schur parameter $\rho = 1$ (hyperbolic rotation does not exist). Therefore, using a Schur–type method (13) does not remove this problem if the upper triangular structure of $\boldsymbol{R}$ must be preserved. However, the Schur–type methods can be extended to the singular case [9, 16]. By applying orthogonal transformations from the right hand side, the singular cases ($\rho = 1$) can be omitted and one obtains

$$\mathbf{\Theta}\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{Q}^T = \begin{bmatrix}\mathbf{\Sigma}_1\\ \mathbf{\Sigma}_2\end{bmatrix}\boldsymbol{R}, \qquad (15)$$

such that

$$\boldsymbol{S} = \boldsymbol{M}^T\boldsymbol{D}\boldsymbol{M},$$

where $\boldsymbol{D}$ is still the diagonal signature matrix but $\boldsymbol{M} = \boldsymbol{R}\boldsymbol{Q}$ is no longer upper triangular (it can be computed to be as close as possible to the upper triangular structure). Strategies for dealing with these singular cases in Schur–type methods have been presented in [3]. The discussions in this section show that the Schur–type methods also allow extensions of known matrix decompositions to more general matrices.

4.3. Arbitrary Matrices

Let an $m_1 \times n$ matrix $\boldsymbol{X}_1$ ($m_1 \geq n$) and an $m_2 \times n$ matrix $\boldsymbol{X}_2$ ($m_2 \geq n$) have their QR decompositions

$$\boldsymbol{X}_1 = \boldsymbol{Q}_1\begin{bmatrix}\boldsymbol{A}\\ \mathbf{0}\end{bmatrix} \qquad \boldsymbol{X}_2 = \boldsymbol{Q}_2\begin{bmatrix}\boldsymbol{B}\\ \mathbf{0}\end{bmatrix}$$

respectively, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are upper triangular $n \times n$ matrices. Then with

$$\boldsymbol{P} = \begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}\begin{matrix}+\\-\end{matrix}, \qquad (16)$$

the generalized Schur algorithm 2 computes (13), i.e. the decomposition

$$\boldsymbol{X}_1^T\boldsymbol{X}_1 - \boldsymbol{X}_2^T\boldsymbol{X}_2 = \boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}, \qquad (17)$$

as in (14).

Remark 3 For two special cases with $\boldsymbol{X}_1 = \gamma\boldsymbol{I}_n$ and $\boldsymbol{X}_2 = \gamma\boldsymbol{I}_n$, respectively, there are some important applications in linear algebra and signal processing:

- $\boldsymbol{X}_2 = \gamma\boldsymbol{I}_n$: The implicit Cholesky decomposition of Fernando and Parlett [4] and von Matt [14] is actually the presented Schur–type method, where γ is the shift for the singular value computation. The presented Schur–type method for arbitrary matrices also allows shifts larger than the smallest singular value without causing a breakdown of the algorithm.

Definition 2 *The $2n \times 2n$ matrix $\boldsymbol{\Theta}_{k,l}(\boldsymbol{F})$ which is obtained by embedding of the 2×2 matrix $\boldsymbol{F} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$ in the $2n \times 2n$ identity matrix $\boldsymbol{I}_{2n}$ such that*

$$\boldsymbol{\Theta}_{k,l}(\boldsymbol{F}) = \begin{array}{c} \begin{array}{ccccccc} & & k & & l & & \end{array} \\ \left[\begin{array}{ccccccc} 1 & & 0 & & 0 & & 0 \\ & \ddots & \vdots & & \vdots & & \\ 0 & \cdots & k_{11} & \cdots & k_{12} & \cdots & 0 \\ & & \vdots & \ddots & \vdots & & \\ 0 & \cdots & k_{21} & \cdots & k_{22} & \cdots & 0 \\ & & \vdots & & \vdots & \ddots & \\ 0 & & 0 & & 0 & & 1 \end{array}\right] \begin{array}{c} \\ \\ k \\ \\ l \\ \\ \\ \end{array} \end{array}$$

is an elementary $\boldsymbol{J}$–orthogonal transformation if $\boldsymbol{F} = \boldsymbol{H}$, and an elementary orthogonal transformation if $\boldsymbol{F} = \boldsymbol{G}$, respectively.

3. THE SCHUR ALGORITHM

Schur stated that in order to put the criterion for the boundedness of a given power series in a more elegant form, we should not start with a power series but with the quotient of two power series [12]. Defining $c(z)$ as a fraction, i.e. $c(z) = b(z)/a(z)$ with $b(z) = c(z)$ and $a(z) = 1$, also leads to a linearization of the bilinear form (1).

As there is an isomorphism between power series and upper triangular Toeplitz matrices, it is possible to represent the power series

$$a(z) = a_0 + a_1 z + a_2 z^2 + \ldots$$

and

$$b(z) = b_0 + b_1 z + b_2 z^2 + \ldots$$

by upper triangular Toeplitz matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, where

$$\boldsymbol{A} = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & & & \\ & a_0 & a_1 & a_2 & \cdots & & \\ & & a_0 & a_1 & a_2 & \cdots & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & & \end{bmatrix} \tag{4}$$

and $\boldsymbol{B}$ is defined similarly.

Assuming that the polynomials are of degree $n-1$, define a $2n \times n$ matrix $\boldsymbol{P}$

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix}. \tag{5}$$

The elementary $\boldsymbol{J}$–orthogonal transformations which annihilate the elements of matrix $\boldsymbol{B}$ diagonalwise are all defined by the same hyperbolic rotation for each diagonal. For example, $\boldsymbol{H}_1$ defines all the transformations which annihilate the b_0's on the diagonal of $\boldsymbol{B}$. Then the necessary Toeplitz structure is preserved for the next subproblem. With $s_i = k + n - (i-1)$, define

$$\boldsymbol{\Theta} = \underbrace{\prod_{k=n}^{n} \boldsymbol{\Theta}_{k,s_n}(\boldsymbol{H}_n)}_{\boldsymbol{\Theta}_n} \cdots \underbrace{\prod_{k=2}^{n} \boldsymbol{\Theta}_{k,s_2}(\boldsymbol{H}_2)}_{\boldsymbol{\Theta}_2} \underbrace{\prod_{k=1}^{n} \boldsymbol{\Theta}_{k,s_1}(\boldsymbol{H}_1)}_{\boldsymbol{\Theta}_1}. \tag{6}$$

Then $\boldsymbol{\Theta}$ is a $2n \times 2n$ $\boldsymbol{J}$–orthogonal transformation composed of $n(n+1)/2$ elementary $\boldsymbol{J}$–orthogonal transformations $\boldsymbol{\Theta}_{k,s_i}(\boldsymbol{H}_i)$ such that

$$\boldsymbol{\Theta} \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{R}$ is an upper triangular matrix which corresponds to the Cholesky factor of $\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{B}^T\boldsymbol{B}$. This is a formulation of the Schur algorithm with a lot of redundancy since the Toeplitz structure of $\boldsymbol{A}, \boldsymbol{B}$ is not taken into consideration. The redundancy can be removed by simply skipping the repeated computations in (7) which is apparently an $O(n^3)$ method. Then the decomposition (7) can be computed in $O(n^2)$, i.e. it reduces to the original Schur algorithm (Appendix: Algorithm 1). However, the formulation of the algorithm as in (6), (7) serves as the basis for the generalizations of the Schur algorithm to more general matrices.

4. GENERALIZATIONS OF SCHUR'S ALGORITHM

In this section we show the generalization of the Schur algorithm progressively from special matrices to more general matrices.

As shown in [2] every symmetric matrix $\boldsymbol{S}$ can be expressed as

$$\boldsymbol{S} = \boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{B}^T\boldsymbol{B}, \tag{8}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are upper triangular. Since the matrix $\boldsymbol{S}$ can be represented as

$$\boldsymbol{S} = \boldsymbol{U}^T + \boldsymbol{D}_S + \boldsymbol{U}$$

where $\boldsymbol{U}$ is the strictly upper triangular part of $\boldsymbol{S}$ and $\boldsymbol{D}_S$ is the diagonal part of $\boldsymbol{S}$, we can define $\boldsymbol{A}$ and $\boldsymbol{B}$ as

$$\boldsymbol{A} = \frac{1}{2}(\boldsymbol{D}_S + 2\boldsymbol{U} + \boldsymbol{I}) \qquad \boldsymbol{B} = \frac{1}{2}(\boldsymbol{D}_S + 2\boldsymbol{U} - \boldsymbol{I}). \tag{9}$$

According to the signs of $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{B}^T\boldsymbol{B}$ in (8) we can construct a $2n \times n$ matrix

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \begin{matrix} + \\ - \end{matrix},$$

where the rows of $\boldsymbol{A}$ form a *positive subspace* and the rows of $\boldsymbol{B}$ form a *negative subspace*. The desired decomposition is then obtained by annihilating the matrix elements by hyperbolic or circular rotations. The type of each rotation for annihilating a specific matrix element is determined as follows:

> *If two rows in the same subspace* (i.e., both in the negative or both in the positive subspace) are combined in annihilation, a *circular rotation* is used.
> *If two rows in subspaces with different signature* are combined in annihilation, a *hyperbolic rotation* is used.

By applying this rule for choosing the type of rotation and using an appropriate ordering scheme for the annihilation of the matrix elements, it is possible to minimize the number of required hyperbolic rotations in Schur–type methods. For example, the hyperbolic Cholesky decomposition of [2] can be computed with n hyperbolic rotations and $n(n-1)/2$ circular rotations (in [2] $n(n+1)/2$ hyperbolic rotations are used).

4.1. Symmetric Positive Definite Matrices

The generalization of the Schur algorithm to symmetric positive definite matrices without Toeplitz structure is straightforward. It is similar to the procedure which is described in the previous section except that unlike in Toeplitz case, there will be no redundant computations in general.

The hyperbolic Cholesky decomposition of $\boldsymbol{S}$ is obtained by applying a sequence of elementary $\boldsymbol{J}$–orthogonal transformations to the $2n \times n$ matrix

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \begin{matrix} + \\ - \end{matrix}, \tag{10}$$

SCHUR–TYPE METHODS BASED ON SUBSPACE CONSIDERATIONS

Jürgen Götze[1 †] *and Haesun Park*[2 ‡]

[1]Network Theory & Circuit Design, Munich University of Technology, Arcisstr. 21, D–80290 Munich, Germany
[2]Computer Science Department, University of Minnesota, Minneapolis, Minnesota 55455, U.S.A.

ABSTRACT

Generalizations of the Schur algorithm are presented and their relation and application to several algorithms in signal processing and linear algebra is elaborated. Based on an algebraic formulation, Schur's algorithm (for symmetric positive definite Toeplitz matrices) is generalized to more general matrices such as symmetric positive definite matrices, symmetric matrices, and general rectangular matrices. The resulting Schur–type methods are related to matrix decompositions such as Cholesky decomposition, $\mathbf{R}^T\boldsymbol{DR}$–decomposition, and implicit Cholesky decomposition. When the number of hyperbolic rotations is minimized (which simultaneously maximizes the number of circular rotations) based on a subspace criteria, the relationship between the Schur algorithm and these decompositions as well as the suitability of the Schur algorithm for various signal processing applications (particularly signal/noise subspace estimation) becomes evident.

1. INTRODUCTION

In his 1917 paper 'Über Potenzreihen, die im Inneren des Einheitskreises beschränkt sind' [12] Issai Schur presented a *continued fraction algorithm* for determining if an analytic function $c(z)$ given by its power series expansion

$$c(z) = c_0 + c_1 z + c_2 z^2 + \dots$$

convergent for $|z| < 1$ satisfies the condition $|c(z)| < 1$. The algorithm gives *an intrinsically parametric representation for the coefficients of the power series* by a sequence of complex numbers ρ_i $(i = 0, 1, 2, \dots)$, which are called *Schur parameters*. With $c^0(z) = c(z)$,

$$c^{i+1}(z) \equiv \frac{1}{z} \cdot \frac{c^i(z) - c^i(0)}{1 - \bar{c}^i(0)c^i(z)}\,, \quad i = 0, 1, 2, \dots \tag{1}$$

yields a finite or infinite sequence of functions, where the overbar denotes complex conjugation. By using the Schwarz Lemma it can be shown [12] that $|c(z)| < 1$ holds for $|z| < 1$ if and only if the Schur parameters fullfill

$$|\rho_i| \equiv |c^i(0)| < 1\,, \quad i = 0, 1, 2, \dots\,. \tag{2}$$

This parametrization of function $c(z)$ has an algebraic analogue in the parametrization of a symmetric positive definite Toeplitz matrix. If θ_i are the rotation angles of the hyperbolic rotations required for computing the *hyperbolic Cholesky factorization* [2] of the respective Toeplitz matrix, the Schur parameters are given by $\rho_i = \tanh(\theta_i)$. In recent years various generalizations and applications of Schur's algorithm were presented mainly based on the concept of displacement rank (see [15] and the references therein).

In this paper we present a derivation of Schur–type methods based on a subspace criteria. For that purpose we describe the related factorization and parametrization problems for an $n \times n$ matrix using a $2n \times n$ matrix, where n rows are in a positive subspace and n rows are in a negative subspace [7] (for rectangular matrices we assume a preparatory QR decomposition). The presented derivation and the resulting algorithm does not require any assumptions on the displacement rank of the matrices. It works for arbitrary matrices with a minimal number of hyperbolic rotations. The subspace criteria also reveals the connection to subspace estimation in signal processing [8]. Furthermore, the numerical behavior of the Schur algorithm with a minimal number of hyperbolic rotations can directly be related to some well-known matrix decompositions. Therefore, many algorithms in linear algebra and signal processing can be incorporated into the presented framework of Schur–type methods: the shift of the singular values in the qd–algorithm, low–rank matrix approximation and subspace estimation, the ε–rank decision for the solution of a least squares problem, and the fast triangularization of $m \times n$ Toeplitz matrices $(m \geq n)$.

2. $\boldsymbol{J}$-ORTHOGONALITY

Definition 1 *A $2n \times 2n$ matrix $\boldsymbol{\Theta}$ is said to be $\boldsymbol{J}$–orthogonal with*

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_n \end{bmatrix}$$

if it satisfies

$$\boldsymbol{\Theta}^T \boldsymbol{J} \boldsymbol{\Theta} = \boldsymbol{J}\,. \tag{3}$$

$\boldsymbol{\Theta}$ is orthogonal if it satisfies (3) with $\boldsymbol{J} = \boldsymbol{I}_{2n}$.

Remark 1 The Givens rotation

$$\boldsymbol{G} = \frac{1}{\sqrt{1+\tau^2}} \begin{bmatrix} 1 & -\tau \\ \tau & 1 \end{bmatrix}$$

is a 2×2 orthogonal matrix $(\tau = \tan\theta)$.

Remark 2 The hyperbolic rotation

$$\boldsymbol{H} = \frac{1}{\sqrt{1-\rho^2}} \begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$$

is a 2×2 $\boldsymbol{J}$–orthogonal matrix $(\rho = \tanh\theta)$.

†The work of this author was supported in part by the Alexander von Humboldt Foundation.

‡The work of this author was supported in part by the National Science Foundation grants CCR-9209726 and CCR-9509085. Part of this work was done while visiting the Institute of Network Theory and Circuit Design, Munich University of Technology. This visit was supported by Rhode & Schwarz.