11

IN *COMMUNICATIONS, COMPUTATION, CONTROL, AND SIGNAL PROCESSING,* A. PAULRAJ, V. ROYCHOWDHURY, AND C. D. SCHAPER, EDS., CH. 11, PP. 223–232, KLUWER, MA, 1997.

IMPROVING THE ACCURACY OF THE GENERALIZED SCHUR ALGORITHM

S. Chandrasekaran and Ali H. Sayed

Dept. of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560

Dedicated with admiration and gratitude to Prof. Thomas Kailath on the occasion of his 60th birthday.

ABSTRACT

We show how to stabilize the generalized Schur algorithm for the Cholesky factorization of positive-definite structured matrices R that satisfy $R - FRF^T = GJG^T$, where J is a signature matrix, F is a stable lower-triangular matrix, and G is a generator matrix. We use a perturbation analysis to indicate the best accuracy that can be expected from any finite precision algorithm that uses the generator matrix as the input data. We then show that the modified Schur algorithm proposed in this work essentially achieves this bound for a large class of structured matrices.

1 INTRODUCTION

One of the most frequent structures, at least in signal processing applications, is the Toeplitz structure with constant entries along the diagonals of the matrix [1]. A classical algorithm for the Cholesky factorization of the *inverses* of such

matrices is the so-called Levinson-Durbin algorithm [1, 2], an error analysis of which has been provided by Cybenko [3]. He showed that, in the case of positive reflection coefficients, the residual error produced by the Levinson-Durbin procedure is comparable to the error produced by the Cholesky factorization [2, p.191].

A related analysis has been carried out by Sweet [4] for the Bareiss algorithm, which is closely related to an algorithm of Schur [1]. These are fast procedures for the Cholesky factorization of the Toeplitz matrix itself rather than its inverse. Sweet concluded that the Bareiss algorithm is asymptotically stable.

In recent work, Bojanczyk et al [5] further extended and strengthened the conclusions of Sweet [4] by employing elementary downdating techniques [6] that are also characteristic of array formulations of the Schur algorithm [7]. They considered the larger class of quasi-Toeplitz matrices, which includes the Toeplitz matrix as a special case, and provided an error analysis that establishes that the Schur algorithm for this class of matrices is asymptotically stable.

The interesting formulation of Bojanczyk et al. [5] has motivated us to take a closer look at the numerical stability of a generalized Schur algorithm [7] that applies to a wider class of structured matrices. This class is briefly introduced in the next section, where a lower-triangular matrix F is shown to be pivotal in characterizing the structure of the matrix. For example, in the Toeplitz or quasi-Toeplitz cases mentioned above, the matrix F is equal to the shift matrix Z (i.e., a Jordan block with zero eigenvalue and ones on the first subdiagonal). Multiplying a column vector u by Z simply corresponds to shifting down the entries of u by one position. In general, however, the matrix F can be any lower triangular matrix (e.g., diagonal, bidiagonal, strictly lower-triangular, etc.). This creates several complications that we address in order to guarantee a numerically reliable algorithm.

In the discussion that follows we use $\|\cdot\|$ to denote the $2-$norm of its argument. We further assume, without loss of generality, that F is represented exactly in the computer. Also, the $\hat{\cdot}$ notation denotes computed quantities, while the $\bar{\cdot}$ notation denotes intermediate exact quantities. We assume $n\epsilon < 1$, where ϵ denotes the machine precision and n is the matrix size. We also use subscripted δ's to denote quantities smaller than machine precision in magnitude, and subscripted c's to denote low order polynomials in n. We further assume that in our floating point model, additions, subtractions, multiplications, divisions, and square-roots are done to full relative accuracy, i.e.,

$$fl(x \circ y) = (x \circ y)(1 + \delta),$$

where $\circ$ denotes $+, -, \times, \div$ and $|\delta| \leq \epsilon$. Likewise for the square-root operation.

2 DISPLACEMENT STRUCTURE

The notion of displacement structure has proven to be a very powerful and convenient tool in several problems in systems and mathematics. Since its introduction by Kailath et al. in the late seventies [8], it has provided a systematic way to elegantly describe and exploit different forms of matrix structures that often arise in applications. For reasons of brevity, however, we limit ourselves here to a brief review of the case of positive-definite symmetric structured matrices. For more details on the topic, the interested reader may consult [7]; a recent survey on the origins and theory of the displacement structure concept.

Consider an $n \times n$ symmetric positive-definite matrix R and an $n \times n$ lower-triangular real-valued matrix F. The displacement of R with respect to F is denoted by ∇_F and defined as

$$\nabla_F = R - FRF^T. \tag{11.1}$$

The matrix R is said to have low displacement rank with respect to F if the rank of ∇_F is considerably lower than n. In this case, R is said to have displacement structure with respect to F [7]. Let $r \ll n$ denote the rank of ∇_F. It follows that we can factor ∇_F as $\nabla_F = GJG^T$, where G is an $n \times r$ matrix and J is a signature matrix of the form $J = (I_p \oplus -I_q),\ p+q = r$. The factorization $\nabla_F = GJG^T$ is highly nonunique since G can be replaced by $G\Theta$ for any $J-$unitary matrix Θ ($\Theta J \Theta^T = J$). A matrix R is therefore structured with respect to the displacement operation defined by (11.1) if it satisfies a displacement equation of the form

$$R - FRF^T = GJG^T\ , \tag{11.2}$$

with a low rank matrix G. Equation (11.2) uniquely defines R iff the diagonal entries of F satisfy $1 - f_i f_j \neq 0$ for all i, j. This condition is assumed throughout the paper, although it can be relaxed in some instances [7].

The pair (G, J) is said to be a generator pair for R since, along with F, it completely identifies R. Note however that, while R has n^2 entries, the matrix G has nr entries and r is usually much smaller than n. Therefore, algorithms that operate on the entries of G, with the purpose of obtaining a triangular factorization for R, will generally be an order of magnitude faster than algorithms that operate on the entries of R itself. The generalized Schur algorithm

is one such fast $O(rn^2)$ procedure, which receives as input data the matrices (F, G, J) and provides as output data the Cholesky factor of R [7].

3 GENERALIZED SCHUR ALGORITHM

We focus here, for simplicity, on matrices R with displacement rank 2, say

$$R - FRF^T = \begin{bmatrix} u_1 & v_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_1 & v_1 \end{bmatrix}^T, \tag{11.3}$$

where u_1 and v_1 denote the $n \times 1$ column vectors of G. Extensions of the discussion to the $r > 2$ case are possible and will be discussed elsewhere. The diagonal entries of F are also assumed to be strictly inside the open unit disc ($|f_i| < 1$). In this case, the matrix F is said to be stable. Also, since G is highly nonunique, it can always be chosen in proper form,

$$G = \begin{bmatrix} x & 0 \\ x & x \\ \vdots & \vdots \\ x & x \end{bmatrix}. \tag{11.4}$$

That is, the top entry of v_1, v_{11}, can always be chosen to be zero.

GENERALIZED SCHUR ALGORITHM. *Start with (u_1, v_1) and repeat for $i \geq 1$:*

1. *Compute $\Phi_i = (F - f_i I)(I - f_i F)^{-1}$ and form $\begin{bmatrix} \Phi_i u_i & v_i \end{bmatrix}$. The top i entries of $\Phi_i u_i$ and v_i are zero.*

2. *Apply a hyperbolic rotation Θ_i in order to annihilate the $(i+1)$ entry of v_i. Denote the resulting column vectors by (u_{i+1}, v_{i+1}):*

$$G_{i+1} = \begin{bmatrix} u_{i+1} & v_{i+1} \end{bmatrix} = \begin{bmatrix} \Phi_i u_i & v_i \end{bmatrix} \Theta_i. \tag{11.5}$$

   G_{i+1} will also be in proper form, with its top i rows equal to zero.

3. *The $i-th$ column of the Cholesky factor L is given by*

$$l_i = \sqrt{1 - |f_i|^2}(I - f_i F)^{-1} u_i. \tag{11.6}$$

   The top $(i-1)$ entries of l_i are zero.

After n steps, the algorithm provides the Cholesky decomposition $R = \sum_{i=1}^{n} l_i l_i^T$. Moreover, let R_i denote the Schur complement of R with respect to its leading $i \times i$ submatrix (R_i is $(n-i) \times (n-i)$) and define the $n \times n$ embedding

$$\tilde{R}_i = \begin{bmatrix} 0 & 0 \\ 0 & R_i \end{bmatrix}.$$

It can be shown that $\tilde{R}_i - F\tilde{R}_i F^T = G_i J G_i^T$ [7]. In other words, G_i is a generator matrix for the i^{th} Schur complement, which is therefore structured.

4 LIMITS TO NUMERICAL ACCURACY

Given a symmetric positive-definite matrix R (not necessarily structured), if its Cholesky factor is evaluated by any standard backward stable method that operates on the entries of R, e.g., Gaussian elimination, the corresponding error bound is given by $\|R - \hat{L}\hat{L}^T\| \leq c_1 \epsilon \|R\|$ [2]. A fundamental question that needs to be answered then is the following: given (F, G, J), but not R, how accurately can we expect to be able to compute the Cholesky factorization of R *irrespective* of the algorithm used (*slow or fast*)? To address this issue we note that just representing (F, G) in finite precision already induces round-off errors. This in turn imposes limits on how accurate an algorithm that employs (F, G) can be. We demonstrate this point by the following example.

Let F be a stable diagonal matrix with distinct entries $\{f_i\}$ and assume f_1 is the largest in magnitude. Let the entries of the column vectors u_1 and v_1 be given by $u_{i1} = (0.5)^{i-1}$, $v_{i1} = \gamma f_i u_{i1}$, where $0 < \gamma < 1$. The matrix R that solves (11.3) for the given (F, u_1, v_1) is symmetric positive-definite. Now define the perturbed vectors $\hat{u}_1$ and $\hat{v}_1$ with $\hat{u}_{11} = u_{11}(1+\delta)$, $\hat{u}_{i1} = u_{i1}$, $i \geq 2$, and $\hat{v}_1 = v_1$. That is, we only make a relative perturbation in the first entry of u_1 and keep all other entries of u_1 and v_1 unchanged. Here, δ is a small number (e.g., for round-off errors, $|\delta|$ is smaller than machine precision). Let $\hat{R}$ be the solution of the displacement equation with the perturbed generator, $\hat{R} - F\hat{R}F^T = \hat{G}J\hat{G}^T$, $\hat{G} = \begin{bmatrix} \hat{u}_1 & \hat{v}_1 \end{bmatrix}$, and introduce the error matrix $E = R - \hat{R}$. Then it can be shown that a *lower bound* on the norm of E is

$$\|E\| \geq |E_{11}| \geq \frac{3}{2}|\delta| \, \|(I - F \otimes F)^{-1}\| \, \|u_1\|^2.$$

By suitably choosing γ, the norm of R can be much smaller than the above bound. Indeed,

$$\|R\| \quad \leq \quad n \max_i \left[\frac{1 - \gamma^2 f_i^2}{1 - f_i^2} \, u_{ii}^2 \right],$$

which shows that as $\gamma \to 1$, the norm of R can be bounded by $n\|u_1\|^2$. In summary, we have shown that, at the same time, $\|R - \hat{R}\|$ can be larger than $|\delta|\ \|(I - F \otimes F)^{-1}\|\ \|u_1\|^2$ and $\|R\|$ can be much smaller than $\|(I - F \otimes F)^{-1}\|\ \|u_1\|^2$. Hence, in general, we can not expect the error norm, $\|R - \hat{L}\hat{L}^T\|$, for any algorithm (slow or fast) that uses (F, G, J) as input data (but not R), to be as small as $c_1|\delta|\ \|R\|$, for some constant c_1.

Therefore, we conclude that *irrespective* of the algorithm we use (*slow or fast*), if the input data is (F, G, J), then we can not expect a better bound than

$$\|R - \hat{L}\hat{L}^T\| \leq c_2\epsilon\ \|(I - F \otimes F)^{-1}\|\ \|u_1\|^2. \tag{11.7}$$

5 HYPERBOLIC ROTATIONS

We now discuss how to improve the numerical accuracy of the generalized Schur algorithm. We start by discussing reliable implementations for the hyperbolic rotations.

For this purpose, we denote the prearray $\left[\begin{array}{cc} \Phi_i u_i & v_i \end{array}\right]$ by $\bar{G}_{i+1}$ and rewrite (11.5) in the compact form $G_{i+1} = \bar{G}_{i+1}\Theta_i$. This expression shows that, in infinite precision, the generator matrices G_{i+1} and $\bar{G}_{i+1}$ must satisfy the fundamental requirement $G_{i+1}JG_{i+1}^T = \bar{G}_{i+1}J\bar{G}_{i+1}^T$. Obviously, this condition cannot be guaranteed in finite precision. Indeed, a naive implementation of the hyperbolic transformation Θ_i can lead to large errors. In finite precision, if we apply Θ_i directly to $\bar{G}_{i+1}$ we obtain a computed $\hat{G}_{i+1}$ such that $\|\hat{G}_{i+1}J\hat{G}_{i+1}^T - \bar{G}_{i+1}J\bar{G}_{i+1}^T\| \leq c_3\epsilon\ \|\bar{G}_{i+1}\|^2\ \|\Theta_i\|^2$. This is not desirable since $\|\Theta_i\|$ can be large. One possible way to ameliorate the problem is to employ a mixed-downdating procedure [5, 6].

An alternative so-called OD procedure that we propose is based on using the SVD of a hyperbolic rotation Θ:

$$\begin{aligned} \Theta &= \frac{1}{\sqrt{1-\rho^2}} \left[\begin{array}{cc} 1 & -\rho \\ -\rho & 1 \end{array}\right], \quad \rho = \frac{\beta}{\alpha}, \\ &= \frac{1}{\sqrt{2}} \left[\begin{array}{cc} 1 & 1 \\ -1 & 1 \end{array}\right] \left[\begin{array}{cc} \sqrt{\frac{\alpha+\beta}{\alpha-\beta}} & 0 \\ 0 & \sqrt{\frac{\alpha-\beta}{\alpha+\beta}} \end{array}\right] \left[\begin{array}{cc} 1 & -1 \\ 1 & 1 \end{array}\right] \frac{1}{\sqrt{2}} = QDQ^T. \end{aligned}$$

This guarantees $\|\hat{G}_{i+1}J\hat{G}_{i+1}^T - \bar{G}_{i+1}J\bar{G}_{i+1}^T\| \leq c_4\epsilon\ \left(\|\bar{G}_{i+1}\|^2\ +\ \|\hat{G}_{i+1}\|^2\right)$.

Another proposal (with better numerical properties) is to employ what we refer to as the H procedure. Let $\begin{bmatrix} x_1 & y_1 \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \Theta$ and assume w.l.o.g $|x| > |y|$. The advantage of the H procedure is that the computed quantities $\hat{x}_1$ and $\hat{y}_1$ satisfy $\begin{bmatrix} \hat{x}_1 + e_1 & \hat{y}_1 + e_2 \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \Theta$, with $|e_1| \leq c_5 \epsilon |\hat{x}_1|$, $|e_2| \leq c_6 \epsilon (|\hat{x}_1| + |\hat{y}_1|)$. The term x_1 is computed as

$$x_1 = \frac{|\alpha| x}{\sqrt{(\alpha - \beta)(\alpha + \beta)}} \left[1 - \frac{\beta}{\alpha} \frac{y}{x} \right] ,$$

where $\xi = 1 - \frac{\beta}{\alpha} \frac{y}{x}$ is evaluated to full relative accuracy as follows: If $\frac{\beta}{\alpha} \frac{y}{x} < 1/2$ then $\xi \leftarrow 1 - \frac{\beta}{\alpha} \frac{y}{x}$, else compute $d_1 = \frac{|\alpha| - |\beta|}{|\alpha|}$, $d_2 = \frac{|x| - |y|}{|x|}$, and $\xi \leftarrow d_1 + d_2 - d_1 d_2$. To compute y_1 we use the expression

$$y_1 = x_1 - \sqrt{\frac{\alpha + \beta}{\alpha - \beta}} \, (x - y).$$

From now on we shall denote by $\hat{u}_{i+1}$ and $\hat{v}_{i+1}$ the computed generator columns at step i, i.e., $\hat{G}_{i+1} = \begin{bmatrix} \hat{u}_{i+1} & \hat{v}_{i+1} \end{bmatrix}$, starting with $\bar{G}_i = \begin{bmatrix} \Phi_i \hat{u}_i & \hat{v}_i \end{bmatrix}$. Each step of the algorithm also requires multiplying the Blaschke matrix Φ_i by u_i.

6 THE BLASCHKE MATRIX

When F is stable and diagonal, Φ_i itself will be diagonal with $(\Phi_i)_{jj} = \frac{f_j - f_i}{1 - f_i f_j}$. In this case, the numerator of $(\Phi_i)_{jj}$ can be computed to full relative accuracy since $fl(f_j - f_i) = (f_j - f_i)(1 + \delta_1)$, while the denominator $x_{ij} = (1 - f_i f_j)$ can be evaluated as follows. If $f_i f_j < 1/2$ then $x_{ij} \leftarrow 1 - f_i f_j$, else compute $d_j = 1 - |f_j|$, $d_i = 1 - |f_i|$, and $x_{ij} \leftarrow d_i + d_j - d_i d_j$. It can be shown that this guarantees $\hat{x}_{ij} = x_{ij} \, (1 + 33\delta_{11})$. Therefore, $\Phi_i \hat{u}_i$ can be computed to component-wise relative accuracy. As for the l_i's, define $\bar{l}_i = \sqrt{1 - f_i^2} \, (I - f_i F)^{-1} \hat{u}_i$. If we use the expression

$$(\bar{l}_i)_j = \frac{\sqrt{(1 - f_i)(1 + f_i)}}{1 - f_i f_j} (\hat{u}_i)_j,$$

to compute $\bar{l}_i$, with the technique explained above for the denominator $(1 - f_i f_j)$, then we can show that $(\hat{l}_i)_j = (\bar{l}_i)_j (1 + c_7 \delta_{13})$.

When F is strictly lower-triangular, the expression for Φ_i collapses to $\Phi_i = F$. In this case we use the standard matrix-vector multiplication and obtain $\| fl(F\hat{u}_i) - F\hat{u}_i \| \leq c_8 \epsilon \|F\| \, \|\hat{u}_i\|$. Also, since $f_i = 0$, $\bar{l}_i = \hat{u}_i = \hat{l}_i$.

In summary, it follows that for diagonal and for strictly lower-triangular F we can write $\|fl(\Phi_i\hat{u}_i) - \Phi_i\hat{u}_i\| \leq c_9\epsilon\|\Phi_i\|\ \|\hat{u}_i\|$.

Finally, we remark that condition (11.11) on the matrix R guarantees the positive-definiteness of the successive Schur complements and, therefore, that $|\Phi_i\hat{u}_i|_{i+1} > |\hat{v}_i|_{i+1}$. This assures that the reflection coefficients will be smaller than one in magnitude, a condition that can be enforced in finite-precision as follows: If $|fl(\Phi_i\hat{u}_i)|_{i+1} < |\hat{v}_{i+1,i}|$ then $fl(\Phi_i\hat{u}_i)_{i+1} \leftarrow |\hat{v}_{i+1,i}|(1+3\epsilon)\text{sign}(fl(\Phi_i\hat{u}_i)_{i+1})$.

7 ERROR ANALYSIS

It follows from the procedures suggested above, for implementing the hyperbolic transformations and for evaluating $\Phi_i u_i$, that

$$\hat{u}_{i+1}\hat{u}_{i+1}^T - \hat{v}_{i+1}\hat{v}_{i+1}^T = \Phi_i\hat{u}_i\hat{u}_i^T\Phi_i^T - \hat{v}_i\hat{v}_i^T - M_{i+1}, \tag{11.8}$$

where the error matrices M_{i+1} satisfy

$$\|M_{i+1}\| \leq c_{10}\epsilon(\|\hat{u}_{i+1}\|^2 + \|\Phi_i\|^2\|\hat{u}_i\|^2 + \|\hat{v}_{i+1}\|^2 + \|\hat{v}_i\|^2). \tag{11.9}$$

If we define $\bar{R} = \sum_{i=1}^n \bar{l}_i\bar{l}_i^T$, $\hat{R} = \sum_{i=1}^n \hat{l}_i\hat{l}_i^T$, and $E = R - \hat{R}$, then the following error bound can be established:

$$\|E\| \leq c_{11}\epsilon\|(I - F\otimes F)^{-1}\|\ (2 + \|F\|^2)\left[\|\bar{R}\| + \sum_{j=1}^n \|\hat{u}_j\|^2\right]. \tag{11.10}$$

The derivation of this bound assumes that the algorithm does not break down. That is, at every iteration the reflection coefficients of the hyperbolic rotations are assumed to be strictly less than one in magnitude. This can be guaranteed by requiring the minimum eigenvalue of R to meet the lower bound

$$\lambda_{\min}(R) \ > \ c_{11}\epsilon\|(I - F\otimes F)^{-1}\|\ (2 + \|F\|^2)\left[\|\bar{R}\| + \sum_{j=1}^n \|\hat{u}_j\|^2\right]. \tag{11.11}$$

This ensures the positive-definiteness of the computed Schur complements.

The term $\|(I-F\otimes F)^{-1}\|$ in the error bound is expected from the perturbation analysis of Sec. 4. However, the presence of the norms of the successive generators makes the error bound larger than the bound suggested by the perturbation

analysis, which only depends on the norm of the first generator matrix. The natural question then is how big can the norm of the generators be?

When F is strictly lower triangular, the error bound can be written in the alternate form

$$\|E\| \leq c_{12}\epsilon(2 + \|F\|^2) \left(\sum_{i=1}^{n} \|F^i\|^2\right) \left[\|\bar{R}\| + \left(\sum_{i=1}^{n} \|\hat{u}_i\|^2\right)\right],$$

and the u_i can be bounded by $\|u_i\|^2 \leq \|R\|$. This shows that, when F is contractive ($\|F\| \leq 1$), the error bound is as good as can be expected from the perturbation analysis of Sec. 4. If F is strictly lower triangular but non-contractive then the error norm can possibly depend on $\|(I - F \otimes F)^{-1}\|$.

When F is a stable diagonal matrix, the growth of the generators $\{\hat{u}_i\}$ can be further controlled [9]. Also, the bound in (11.10) may suggest that the norm of the error can become very large when the magnitude of the diagonal entries of F become close to one. But this is not necessarily the case since we can show that

$$\|E\| \leq c_{13}\epsilon \, \frac{\sum_{i=1}^{n} \|\bar{R}_i\|}{1 - \max_{j,i} \left(\rho_{j,i}^2\right)},$$

which is independent of the $\{f_i\}$. Here, $\rho_{j,i} = \hat{v}_{j,i}/\hat{u}_{j,i}$. In other words, if the coefficients $\rho_{j,i}$ are sufficiently smaller than one, then the algorithm will be backward stable irrespective of how close the $\{|f_i|\}$ are to one.

8 CONCLUDING REMARKS

For brevity, we have omitted several relevant points from our discussion. We briefly comment on these issues here.

To begin with, our analysis suggests that for $\|F\|$ sufficiently close to one, the error norm can become large. However, if our original motivation is to solve $Rx = b$, then the error can be improved by using iterative refinement [12].

Also, the analysis and results of this paper can be extended to positive-definite structured matrices with displacement ranks larger than 2, as well as to other forms of displacement structure, say the Hankel-like case $FR + RF^T = GJG^T$. We are also pursuing the extension of our results to general nonsymmetric structured matrices. Moreover, the results of this work suggest improvements to certain fast algorithms in adaptive filtering and state-space estimation in view of the connections of these algorithms to the Schur algorithm [7, 13, 14].

Finally, we remark that for indefinite or nonsymmetric matrices the LDU factorization is numerically difficult to compute without pivoting. Since pivoting can destroy the structure of a matrix, it is not always possible to incorporate it into a fast algorithm directly. Sometimes one can transform the matrix so that it is structured with respect to a diagonal F and then use partial pivoting during the Schur algorithm [10, 11]. Another way to approach this problem is to compute the QR factorization of the structured matrix rapidly. Several methods have been proposed earlier in the literature but none of them are numerically stable. In recent work, and motivated by the results of this paper, we have proposed a fast and backward stable Schur-type algorithm for solving linear systems of equations involving nonsymmetric Toeplitz (also quasi-Toeplitz and Toeplitz-like) matrices [15]. It is based on a modified fast QR factorization of the coefficient matrix T in $Tx = b$. The algorithm actually computes the factorization $T = \Delta(\Delta^{-1}Q)R$, where Δ is lower-triangular, $(\Delta^{-1}Q)$ is orthogonal and R is upper triangular. The solution x is then computed efficiently by using $x = R^{-1}(Q^T\Delta^{-T})\Delta^{-1}b$, which can be done in $O(n^2)$ operations, where n is the matrix dimension.

Acknowledgment

This work was partially supported by NSF under award no. MIP-9409319.

REFERENCES

[1] Kailath, T., "A theorem of I. Schur and its impact on modern signal processing,", *Operator Theory: Advances and Applications*, vol. 18, pp. 9–30, Birkhäuser, Boston, 1986.

[2] Golub, G. H. and Van Loan, C. F., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, second ed., 1989.

[3] Cybenko, G., "The numerical stability of the Levinson-Durbin algorithm for Toeplitz system of equations," *SIAM J. Sci. Statis. Comput.,* vol. 1, pp. 303–319, 1980.

[4] Sweet, D. R., *Numerical Methods for Toeplitz Matrices*, PhD thesis, University of Adelaide, Adelaide, Australia, 1982.

[5] Bojanczyk, A. W., Brent, R. P., de Hoog, F. R., and Sweet, D. R., "On the stability of the Bareiss and related Toeplitz factorization algorithms," *SIAM J. Matrix Anal. Appl.,* vol. 16, pp. 40–57, 1995.

[6] Bojanczyk, A. W., Brent, R. P., van Dooren, P., and de Hoog, F. R., "A note on downdating the Cholesky factorization," *SIAM J. Sci. Stat. Comput.,* vol. 8, pp. 210–221, 1987.

[7] Kailath, T. and Sayed, A. H., "Displacement structure: Theory and applications," *SIAM Review,* vol. 37, no. 3, Sep. 1995.

[8] Kailath, T., Kung, S. Y., and Morf, M., "Displacement ranks of a matrix," *Bulletin of the American Mathematical Society*, vol. 1, no. 5, pp. 769-773, Sep. 1979.

[9] Gu, M., "Stable and efficient algorithms for structured systems of linear equations," *in preparation.*

[10] Heinig, G., "Inversion of generalized Cauchy matrices and other classes of structured matrices," in *Linear Algebra for Signal Processing*, IMA volumes in Mathematics and its Applications, vol. 69, pp. 63–81, 1995.

[11] Gohberg, I., Kailath, T., and Olshevsky, V., "Fast Gaussian elimination with partial pivoting for matrices with displacement structure," to appear in *Math. of Computation.*

[12] Chandrasekaran, S. and Sayed, A. H., "Stabilizing the fast generalized Schur algorithm," *submitted for publication.*

[13] Sayed, A. H., Kailath, T., and Lev-Ari, H., "Generalized Chandrasekhar recursions from the generalized Schur algorithm," *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2265–2269, Nov. 1994.

[14] Sayed, A. H. and Kailath, T., "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18-60, Jul. 1994.

[15] Chandrasekaran, S. and Sayed, A. H., "A fast and stable solver for nonsymmetric structured systems," *in preparation.*