Journal of Technical Science and Technologies; ISSN 2298-0032; Volume 3, Issue 1, 2014

# Application of a Displacement Structure for Acceleration of Novel Matrix Spectral Factorization Algorithm

**Victoria BARAMIDZE***
**Lasha EPHREMIDZE****
**Cihan MERT*****
**Nika SALIA******

## Abstract

In this paper, we describe an algorithm for solving a certain system of algebraic linear equations using the underlying displacement structure of the coefficients matrix of the system. Fast solution of this system is a key component for acceleration of recently developed novel matrix spectral factorization algorithm. The results of numerical simulations, which compare optimized software implementation of structured system's solution to the standard one built in MATLAB, are presented as well.

**Keywords:** Displacement structure, Spectral factorization, Linear system of equations

## Introduction

Spectral factorization is the process by which a positive matrix-valued function S, on the unit circle in the complex plane, is expressed in the form $S(t)=S^+(t)(S^+(t))^*, |t|=1$. Here, $S^+(z), |z|<1$, is a certain analytic matrix-valued function and $\left(S^+(t)\right)^*$ is its Hermitian conjugate. Such factorization plays a prominent role in a wide range of fields in Communications (Fischer, 2002), System Theory (Cutain & Zwart, 1995), Control Engineering (Anderson & Moore 1979), and so on. No formula exists for an explicit computation of the spectral factor, and, consequently, several different methods have been developed for an approximate calculation of S^+. Recently, a new method of matrix spectral factorization has been proposed in (Janashia, Lagvilava, & Ephremidze, 2011). To describe this method in a few words, it carries out step-by-step approximate spectral factorization of leading principal minors of $S$. In this process the decisive role is played by unitary matrix functions of certain form, the so called wavelet matrices (Ephremidze, Lagvilava, 2013), which eliminate many technical difficulties related to the computation of $S^+$. The explicit construction of such matrices requires solution of a certain $N\times N$ linear system of algebraic equations,

$$\Delta X = B, \tag{1}$$

which, as it is widely known, requires $O\left(N^3\right)$ operations, even though in the considered case *Δ* is always positive definite. While the matrix function $S$ to be factorized has dimensions $r\times r$, and consequently its leading principal minors have order $m\leq r$, the positive integer $N$ is responsible for the accuracy of their approximately computed spectral factors. In fact, it has been proved (Ephremidze, Janashia, & Lagvilava, 2011), that the obtained algorithm produces an approximation which converges to the exact solution as $N\to\infty$. Consequently, in actual calculation process by the above mentioned spectral factorization algorithm, N is likely to be significantly larger than m, and $O\left(N^3\right)$ operations for solving (1) would be a heavy computational burden for the whole process. However, the special form of the matrix *Δ*, whose entries are determined by m×N elements instead of $N^2$ (see next section), promise the existence of fast ways of solving (1), which had been intensively researched for a long time. In the end, it has been shown that *Δ* possesses the so called "displacement structure" (Kailath, Sayed, &Hassibi, 2000, Appendix E) by means of which the computational burden of solving the system (1) reduces from $O\left(N^3\right)$ to $O\left(mN^2\right)$.

There are several forms of displacement structures and we selected a suitable one. Consider an $N\times N$ Hermitian matrix *Δ* and $N\times N$ upper-triangular shift matrix $Z$ with ones on

*Assoc. Prof. Dr., Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia
E-mail: vbaramidze@ibsu.edu.ge
**Assoc. Prof. Dr., Razmadze Mathematical Institute of Javakhishvili State University, Tbilisi, Georgia
E-mail: lasha@rmi.ge
***Assist. Prof. Dr., Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia
E-mail: cmert@ibsu.edu.ge
**** Department of Mathematics and its Applications of Central European University. Budapest, Hungary
E-mail: Salia_Nika@student.ceu.hu

the first superdiagonal and zeroes elsewhere (i.e. Jordan block with eigenvalue 0). The displacement of *Δ* with respect to $Z$ is denoted by $R_Z\Delta$ and is defined as the difference

$$R_Z\Delta = \Delta - Z\Delta Z^*$$

The matrix *Δ* is said to have displacement structure (or low displacement rank) with respect to $Z$, if the rank of $R_Z\Delta$ is considerably lower than (and independent of) N. It was shown in (Janashia, Lagvilava, Ephremidze, 2011, Appendix), that the coefficient matrix *Δ* of the system (1) has low dimensional rank m, where $m \le n$ is the order of the leading principal minor which has to be factorized at the given moment. Consequently, there exists a method of solution of (1) with reduced number of operations, as it is described in (Kailath, Sayed, & Hassibi, 2000, Appendix E).

In the present paper, we describe the fast algorithm for solving the system (1) incorporating the underlying displacement structure of the coefficient matrix *Δ*. We estimate the number of operations required at each step of the algorithm. Furthermore, the algorithm was encoded in MATLAB and, by means of numerical simulations, compared to the standard algorithm for solving linear system of equations built in MATLAB. This comparison demonstrates the significant advantage in computational time of the fast algorithm especially when $N \gg m$. Consequently, its application as a component of the above mentioned matrix spectral factorization algorithm will accelerate the whole computational procedure.

## Formulation of the Problem

According to the algorithm published in (Janashia, Lagvilava & Epremidze,2011), a positive matrix-valued function

$$S(t) = \begin{pmatrix} s_{11}(t) & s_{12}(t) & \cdots & s_{1r}(t) \\ s_{21}(t) & s_{22}(t) & \cdots & s_{2r}(t) \\ \vdots & \vdots & \vdots & \vdots \\ s_{r1}(t) & s_{r2}(t) & \cdots & s_{rr}(t) \end{pmatrix} = \sum_{k=-n}^{n} c_k t^k$$

is first factorized as

$$S(t) = M(t)M^*(t) \qquad (2)$$

where $M(t)$ is a lower triangular matrix with minimum phase entries on the diagonal, and $M^*(t)$ is the Hermitian conjugate. Afterwards, the spectral factorization of leading principle minors of $S(t)$ is performed step by step by multiplication of $M(t)$ on the right by unitary matrix functions $U_m(t)$, $m = 2, 3, \ldots, r$, of special form. Eventually we have

$$S^+(t) = M(t)U_2(t)\cdots U_r(t)$$

To make a new row of $M(t)$ causal in each following step, we first use a non-causal factorization

$$S_m(t) = M_m(t)M_m^*(t)$$

where $S_m(t)$ is the $m \times m$ leading principle minor of $S(t)$ and has a block matrix for

$$M_m(t) = \left(\begin{array}{cccc|c} & & & & 0 \\ & S_{m-1}^+(t) & & & 0 \\ & & & & \vdots \\ & & & & 0 \\ \hline \xi_1(t) & \xi_2(t) & \cdots & \xi_{m-1}(t) & f_m^+(t) \end{array}\right)$$

$\xi_k \in L_2(T)$, $k = 1, 2, \cdots, m-1$, $f_m^+(t) = \sqrt{\det S_m(t)} \big/ \sqrt{\det S_{m-1}(t)}$ $(\sqrt{S}$ denotes the scalar spectral factor of $S$, and stands for the Lebesgue space of square integrable functions on the unit circle . Since $S_{m-1}^+(t)$ has the same polynomial order n as $S_{m-1}(t)$, we can use the factorization (2) to find the functions $\xi_k$ explicitly using Cramer's rule. Namely

$$\xi_k(t) = \left(\det S_{m-1}^k(t) \big/ \det S_{m-1}^+(t)\right)^* \; k = 1, 2, \cdots, m-1,$$

where $S_{m-1}^k(t)$ is the matrix $S_{m-1}^+(t)$ with its kth column replaced by the first *m-1* entries of the *m*th column of $S(t)$. These functions have Fourier expansions

$$\xi_k(t) = \sum_{j=-\infty}^{\infty} \gamma_{kj} t^{-j} \qquad (3)$$

Theoretical background of the algorithm suggests that the larger we take $N$, the better the approximation we achieve as the final result. Conveniently enough, since $\xi_k(t)$ is represented as a ratio of two polynomials, where the denominator is free from zeros inside the unit circle, one can calculate as many coefficients of this function, as one wishes. Next we construct a unitary matrix polynomial $U_m(t) = U_m^{[N]}(t)$, which makes the product $M_m(t)U_m(t)$ causal. The construction of $\mathrm{U}_m(t)$ requires solution of the system of linear equations (1), where

$$\Delta = \sum_{k=1}^{m-1} \Theta_k \Theta^*_k + I_{N+1} \qquad (4)$$

is a positive definite matrix with all eigenvalues greater than or equal to 1. $I_{N+1}$ is the *( N + 1)* × *( N + 1)* unit matrix and $\Theta_k$ is a Toeplitz matrix

$$\Theta_k = \begin{pmatrix} \eta_{k0} & \eta_{k1} & \cdots & \eta_{kN} \\ \eta_{k1} & \eta_{k2} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \eta_{kN} & 0 & \cdots & 0 \end{pmatrix}$$

where $\sum_{j=1}^{N} \eta_{kj} t^{-j} = \left[\left(\sum_{j=0}^{N} \gamma_{kj} t^{-j}\right)\left(\sum_{j=0}^{N} b_j t^j\right)\right]^-$ ,

$$\sum_{j=0}^{\infty} b_j t^j = \frac{1}{f_m^+(t)} = \frac{\sqrt{\det S_{m-1}(t)}}{\sqrt{\det S_m(t)}}$$

$$\left( [\cdot]^- \text{ is the projector, i.e.} \left[ \sum_{j=-N}^{N} c_j t^j \right]^- = \sum_{j=-N}^{0} c_j t^j \right)$$

It is proved in Appendix of (Janashia, Lagvilava, & Ephremidze, 2011) that

$$\mathrm{R}_Z \Delta = \mathrm{A} \cdot \mathrm{A}^* \tag{5}$$

where $\mathrm{A} = [\Lambda_1, \Lambda_2, \cdots, \Lambda_{m-1}, \varepsilon]$ is the *(N+1)×m* matrix with each column $A_k$ denoting the first column of $\Theta_k$ , *k=1,2,…,m-1*, and $\varepsilon = (0,0,\cdots,0,1)^T$. As it was explained in Introduction, the equation (5) implies that $\Delta$ has displacement rank *m*. Consequently, as (Kailath, Said, & Hassibi, 2001) suggest UDL (upper-diagonal-lower) factorization of  can be performed in $O(mN^2)$ operations, which allows us to solve the system (1) in a more efficient way. In the next section we describe theoretical background for such factorization.

## Theoretical Background of Efficient UDL Factorization Algorithm

Following the idea presented in (Kailath, Said and Hassibi, 2000), we provide an efficient $O(mN^2)$ procedure for the computation of the triangular factors of a Hermitian positive-definite matrix  satisfying

$$\Delta - Z\Delta Z^* = AA^*, \ A \in C^{N\times r} \tag{6}$$

Let

$$\Delta = L^* D^{-1} L \tag{7}$$

define the triangular decomposition of $\Delta$, where $D = diag\{d_1, d_2, ..., d_N\}$, and the lower triangular factor *L* is normalized in such a way that the values $\{d_i\}$ appear on its main diagonal.

Note that since multiplication by Zon the left moves the rows up by 1 and the multiplication by $Z$ on the right shifts the columns to the left, $\Delta - Z\Delta Z^*$ will have only 0-s in the last row and column. Consequently, the last row (column) of $\Delta$ coincides with the last row (column) of $AA*$ by virtue of the equation (6), and it can be easily found by

$$\Delta_{[N]} = A_{[N]} \cdot A^*,$$

where $\Delta_{[N]}$ and $A_{[N]}$ denote the last rows of $\Delta$ and $A$, respectively. Furthermore, the last row $l_N$ of the lower triangular matrix $L$ in (7) coincides with, and $\Delta_{[N]}$ is its last entry,

$$l_N := \Delta_{[N]} = A_{[N]}A^*, \quad d_N := A_{[N]}A^*_{[N]}$$

Thus, following the standard UDL factorization algorithm, if we subtract from  the rank-one matrix , then we obtain a new matrix $l_N^* d_N^{-1} l_N$ whose last row and column are zero

$$\Delta - l_N^* d_N^{-1} l_N = \begin{bmatrix} \Delta_{N-1} & 0 \\ 0 & 0 \end{bmatrix} =: \tilde{\Delta}_{N-1}$$

where the matrix $\Delta_{N-1}$ is the Schur complement of  with respect to its *(N,N)* entry $d_N$. Our claim now is that $\Delta_{N-1}$ is also structured and satisfies

$$\Delta_{N-1} - Z_{N-1}\Delta_{N-1}Z_{N-1}^* = A_{N-1}A_{N-1}^* \tag{8}$$

with *(N-1)×(N-1)* upper triangular shift matrix $Z_{N-1}$, where the generator matrix $A_{(N-1)}$ can be directly obtained from the previous generator $A$ (see (6). Namely,

$$A_{N-1} = \overline{A}a^*ad^{-1} - \underline{A}a^*ad^{-1} + \underline{A} \tag{9}$$

where, $a{:=}A_N$ , $d{=}d_N$, and $\underline{M}$ (or $\overline{M}$) denotes the matrix $M$ without the last (or the first) row. Indeed, we have

$$\begin{aligned} \tilde{\Delta}_{N-1} - Z\tilde{\Delta}_{N-1}Z^* &= -l_N^* d^{-1} l_N - Z\left(-l_N^* d^{-1} l_N\right) Z^* = \\ &= \Delta - Z\Delta Z^* - l_N^* d^{-1} l_N + Z l_N^* d^{-1} l_N Z^* = \\ &= AA^* - l_N^* d^{-1} l_N + Z l_N^* d^{-1} l_N Z^* = \\ &= AA^* - Aa^* d^{-1} aA^* + ZAa^* d^{-1} aA^* Z^* \end{aligned}$$

Since we know that the last row and column of $\tilde{\Delta}_{N-1} - Z\tilde{\Delta}_{N-1}Z^*$ contains only 0-s, and we get $\Delta_{N-1} - Z_{N-1}\Delta_{N-1}Z_{N-1}^*$ if we eliminate them, we can get rid of the last rows and columns in the above sum to obtain

$$\tilde{\Delta}_{N-1} - Z\tilde{\Delta}_{N-1}Z^* = \underline{A}\underline{A}^* - \underline{A}a^* d^{-1} a\underline{A}^* + \overline{A}a^* d^{-1} a\overline{A}^*$$

On the other hand

$$\begin{aligned} A_{N-1}A_{N-1}^* &= (\overline{A}a^*ad^{-1} - \underline{A}a^*ad^{-1} + \underline{A})\,\mathrm{x} \\ &\left(a^*ad^{-1}\overline{A}^* - a^*ad^{-1}\underline{A}^* + \underline{A}^*\right) = \\ &= \overline{A}a^*ad^{-1}\overline{A}^* - \overline{A}a^*ad^{-1}\underline{A}^* + \overline{A}a^*ad^{-1}\underline{A}^* - \\ &- \underline{A}a^*ad^{-1}\overline{A}^* + \underline{A}a^*ad^{-1}\underline{A}^* - \underline{A}a^*ad^{-1}\underline{A}^* + \\ &+ \underline{A}a^*ad^{-1}\overline{A}^* - \underline{A}a^*ad^{-1}\underline{A}^* + \underline{A}\underline{A}^* = \\ &= \overline{A}a^*ad^{-1}\overline{A}^* - \underline{A}a^*ad^{-1}\underline{A}^* + \underline{A}\underline{A}^* \end{aligned}$$

which proves (8), where $A_{(N-1)}$ is defined by (9). We can further apply the formula (9) recurrently for *N-1,N-2,...,2.*

## Description of the Algorithm

In the present section we describe in details how from $N\times r$ matrix A in (6) one can construct the upper triangular matrix $U{=}L^*$ in (7) and the diagonal entries of D. The nonzero part of the consecutive columns of U is denoted by $u_k$ .

Step 1. Let $A_N{=}A$ and take $N{+}1$ as a starting value of an integer *k, k=N+1* .

Step 2. Take $k = k\text{-}1$.

Compute $u_k = A_k \cdot a_k^*$ , where $a_k$ denotes the last row of $A_k$ (this procedure requires  multiplications), and let  be the last entry of , so that $d_k^{-1}$ is the entry of D.

Step 3. Let $\overline{u}_k$ and $\underline{u}_k$ be the column  without the first and the last entries, correspondingly, and let $v_k = \overline{u}_k - \underline{u}_k$ (it will have *k-1* entries).

Step 4. Compute $B_k = v_k a_k d_k^{-1}$ (it requires again $k \cdot m$ multiplications).

Step 5. Let $A_{k-1} = B_k + \underline{A}_k$ where $\underline{A}_k$ is the matrix $A_k$ without the last row.

Step 6. If $k > 1$, then move to Step 2 .

Overall, the algorithm requires $2\sum_{k=1}^{N} k \cdot m = 2mN^2$ multiplications.

## Estimation of Diagonal Entries.

As it was mentioned in Section 3, the diagonal entries $D = diag\{d_1, d_2, ..., d_N\}$ in factorization (7) are the same for matrices *L**, *D* and *L*. Since $\| a_{k-1} \| \geq \| a_k \|$ is positive definite, we know that all $d_k$ are strictly positive. However, some preliminary estimation of these entries which guarantees that they are remote from 0 would be nice in real computations. In this section we prove that the following nice ordering of these entries happens to hold

$$d_1 \geq d_2 \geq \ldots \geq d_N \geq 1$$

Since $d_k = a_k a_k{}^*$, where $a_k$ is the last row of $A_k$ (see (9)), and $a_N$ = $(\eta_{1N}, \eta_{2N}, \ldots, \eta_{(m-1,N)}, 1)$, the last inequality in (10) is clear. Hence we have to show that $d_{(k-1)} \geq d_k$.

It follows from (9) that

$$a_{k-1} = a_k a_k^* a_k d^{-1} - b_k a_k^* a_k d^{-1} + b_k$$

where $b_k$ is the last but one row of $A_k$ (above $a_k$). Consequently

$$a_{(k-1)} = a_k - b_k \, (I - a_k{}^* \, a_k \, d^{-1})$$

and since

$b_k \left( I - a_k^* a_k d^{-1} \right) \perp$ (as $b_k \left( I - a_k^* a_k d^{-1} \right) a_k^* = b_k a_k^* - b_k a_k^* = 0$ ),

we have $a_{(k-1)} || \geq || a_k ||$ because of Pythagoras theorem.

## Numerical Simulations

To demonstrate how much the application of the displacement structure accelerates a solution of linear system of equations, the algorithm was coded in MATLAB. A PC with 2.40GHz Intel Quad Core CPU and 2GB RAM was used for numerical simulations. Several linear systems were randomly generated and solved using 1) standard MATLAB division tool, 2) the algorithm described in this paper. In all of the experiments the $L_\infty$ norm of the difference between the solutions produced by the two methods were calculated. The two solutions are quite close to precisions in which MATLAB carries out its calculations (the experiments were run in the standard double precision).

Additionally, elapsed time was measured for each of the procedures. The results are listed in the table1 below. As expected, the algorithm which utilizes the underlying displacement structure of the system works faster than the standard one, and the difference between the performance times is becoming more and more evident as m and $N$ grow, keeping $N$ sufficiently larger than $m$.

## Conclusion

A certain type of linear system of equations with displacement structure naturally arises in the process of realization of novel matrix spectral factorization algorithm introduced in (Janashia, Lagvilava, Epremidze, 2011).We have described an efficient UDL factorization algorithm of the coefficient matrix, and have presented numerical evidence that this algorithm has a noticeable advantage in computational speed, when compared to a standard computational tool.

## Acknowledgments

The second author was supported by the Shota Rustaveli National Science Foundation grant (Contract No. 31/47) during this research.

**Table 1.** Regression analysis results with respect to normal case

| Rank *m* | 3 | 5 | 15 | 15 | 15 | 20 |
|---|---|---|---|---|---|---|
| Order *N* | 10 | 100 | 500 | 1000 | 5000 | 5000 |
| Time (Disp. Str.) | 0.0043 | 0.0052 | 0.032 | 0.113 | 3.486 | 4.54 |
| Time (Matlab Alg.) | 0.0152 | 0.0178 | 0.268 | 1.857 | 196.87 | 245.03 |

## References

Anderson B.D.O., Moore J.B., (1979), Optimal Filtering. Prentice-Hall, New Jersey.

Cutain R. F., Zwart H., (1995), An Introduction to Infinite Dimensional Linear Systems Theory. Texts in Appl. Math., 21, Spinger, New York.

Ephremidze L., Lagvilava E., (2014), On Compact Wavelet Matrices of Rank m and of Order and Degree N.J. Fourier Anal. Appl.DOI 10.1007/s00041-013-9317-y.

Ephremidze L., Janashia G., & Lagvilava E., (2011), On Approximate Spectral Factorization of Matrix-Functions. J. Fourier Anal. Appl., 17(5), 976-990, DOI: 10.1007/s00041-010-9167-9.