Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 315 (2004) 453–468
Theoretical Computer Science
www.elsevier.com/locate/tcs

Split algorithms for skewsymmetric Toeplitz matrices with arbitrary rank profile☆

Georg Heinig[a,*], Karla Rost[b]

[a] Dept. of Math. and Comp. Sci., Kuwait University, P.O. Box 5969, Safat 13060, Kuwait
[b] Dept. of Mathematics, Chemnitz University of Technology, D-09107 Chemnitz, Germany

Abstract

Split Levinson-type and Schur-type algorithms for the solutions of linear systems with a nonsingular skewsymmetric Toeplitz matrix are designed. In contrast to previous ones, the algorithms work for any nonsingular skewsymmetric Toeplitz matrix. Moreover, generalizations of ZW- and WZ-factorizations of skewsymmetric Toeplitz matrices related to the new split algorithms are presented.

MSC: 65F05; 15A06; 15A23; 15A09

Keywords: Toeplitz matrix; Skewsymmetric matrix; Split algorithm; Levinson algorithm; Schur algorithm; WZ-factorization

1. Introduction

This paper is dedicated to fast algorithms for nonsingular skewsymmetric Toeplitz matrices, i.e. matrices of the form $T_N = [a_{i-j}]_{i,j=1}^N$ with $a_{-j} = -a_j$. We assume that the entries are from a field $\mathbb{F}$ of characteristic different from two.

A general linear system $T_N\mathbf{f} = \mathbf{b}$ with a nonsingular Toeplitz coefficient matrix can be solved "fast" with complexity $O(N^2)$ using Levinson-type or Schur-type algorithms. A problem is that the classical Levinson and Schur algorithms work only if the matrix T_N is *strongly nonsingular*, which means that all *leading principal submatrices* $T_k = [a_{i-j}]_{i,j=1}^k$ are nonsingular for $k = 1, \ldots, N$. This condition is never satisfied for a

☆ This work was supported by Research Grant SM05/02 of Kuwait University.
* Corresponding author.
E-mail addresses: georg@mcs.sci.kuniv.edu.kw (G. Heinig), krost@mathematik.tu-chemnitz.de (K. Rost).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2004.01.003

skewsymmetric Toeplitz matrix, since skewsymmetric matrices of odd order are always singular.

The problem of fast solving skewsymmetric Toeplitz systems was addressed in our recent paper [14]. In this paper fast algorithms were designed for skewsymmetric Toeplitz matrices which work under the condition that every leading principal submatrix of even order is nonsingular, which means the same as the nonsingularity of all *central submatrices* $T_{N-2\ell} = [a_{i-j}]_{i,j=\ell+1}^{N-\ell}$, $\ell = 0, 1, \ldots, N/2-1$. Matrices with the latter property are called *centro-nonsingular*. The algorithms in [14] are, in principle, split algorithms in the sense of Delsarte-Genin in [3,4]. Some algorithms in [14] are the skewsymmetric counterparts of double-steps split algorithms for symmetric Toeplitz matrices proposed in [16] and [8]. However, surprisingly, there are also algorithms for skewsymmetric Toeplitz matrices that do not have an obvious symmetric counterpart, which is due to some additional symmetry properties of skewsymmetric Toeplitz matrices.

An algorithm for Toeplitz matrices working without additional conditions was first proposed in [7]. A discussion of algorithms of this kind can also be found in [19]. But these algorithms are for general Toeplitz matrices and do not fully utilize additional symmetry properties like symmetry or skewsymmetry. Thus, the aim of the present paper is to design (split) algorithms that exploit both the Toeplitz structure as well as the skewsymmetry and work without assumption on the rank profile. Split algorithms for general symmetric Toeplitz matrices were designed in our recent paper [15]. Let us reiterate that the skewsymmetric case is not simply an analogue of the symmetric case but has some specific peculiarities.

Our approach is based on a look-ahead strategy. In the algorithms we consider only those submatrices T_n which are nonsingular. Let $n_1 < n_2 < \cdots < n_r = N$ be the set of all $n = n_k$ for which T_n is nonsingular, and let $\mathbf{u}^{(k)}$ be the vector spanning the (one-dimensional) nullspace of T_{n_k+1}. Here T_{N+1} means any skewsymmetric extension of T_N. The Levinson-type algorithm computes a vector $\mathbf{u}^{(k+1)}$ from $\mathbf{u}^{(k)}$ and $\mathbf{u}^{(k-1)}$ by a three-term recursion, and the Schur-type algorithm computes the corresponding residuals. The last two vectors $\mathbf{u}^{(k)}$ determine the inverse matrix via an "inversion formula" which allows to solve a linear system efficiently.

Note that a different approach for solving skewsymmetric Toeplitz systems will be discussed in a forthcoming paper [9]. The approach in [9] is based on the recursion of fundamental systems (see [13]). One of its advantages is that it can easily be generalized to the block case, which is not the case for the look-ahead approach.

Like the classical Schur algorithm is related to an LU-factorization of the Toeplitz matrix and the classical Levinson algorithm is related to a UL-factorization of its inverse, the split Schur algorithm for symmetric Toeplitz matrices is related to a ZW-factorization [1] of the matrix and the split Levinson algorithm to a WZ-factorization of its inverse. This was observed in [5]. Concerning WZ-factorization for general matrices we refer to [6,18] and references therein.

In [14] the structure of the ZW-factorization of centro-nonsingular skewsymmetric Toeplitz matrices was studied. It was shown that such a matrix T_N admits a represen-

[1] The definitions of Z- and W-matrices are given in Section 6.

tation $T_N = ZXZ^{\mathrm{T}}$ in which Z is a special unit Z-matrix and X is a skewsymmetric antidiagonal matrix (and a similar result for T_N^{-1}). In the present paper we show that, more general, any nonsingular skewsymmetric Toeplitz matrix admits such a representation in which X is a skewsymmetric block antidiagonal matrix and the blocks are multiples of the identity. The factors Z and X can be computed with the help of the generalized split Schur algorithm. The factorization combined with back substitution gives another possibility to solve linear systems without computing the vectors $\mathbf{u}^{(k)}$.

Besides the solution via inversion formula and factorization we also discuss the solution via direct recursion. We refrain from computing the computational complexities in all cases, since their exact values depend on the rank profile of the matrix. However, it can be pointed out that these values are in general not essentially higher and in most cases even lower than the corresponding values computed in [14] for the case of a centro-nonsingular skewsymmetric Toeplitz matrix.

Let us introduce some notations that will be used throughout the paper. We denote by J_n the $n \times n$ matrix of counteridentity, which has ones on the antidiagonal and zeros elsewhere. A vector $\mathbf{u} \in \mathbb{F}^n$ is called *symmetric* if $\mathbf{u} = J_n\mathbf{u}$ and *skewsymmetric* if $\mathbf{u} = -J_n\mathbf{u}$. An $n \times n$ matrix B is called *centrosymmetric* if $J_nBJ_n = B$ or *centro-skewsymmetric* if $J_nBJ_n = -B$. Let $\mathbb{F}^n_{\pm}$ be the subspaces of $\mathbb{F}^n$ consisting of all symmetric, skewsymmetric vectors, respectively.

Occasionally we will use polynomial language. For a matrix $A = [a_{ij}]$, $A(t,s)$ will denote the bivariate polynomial

$$A(t,s) = \sum_{i,j} a_{ij} t^{i-1} s^{j-1},$$

and for $\mathbf{u} = (u_i)_{i=1}^n$ we set $\mathbf{u}(t) = \sum_{i=1}^n u_i t^{i-1}$.

For a vector $\mathbf{u} = (u_i)_{i=1}^l$, let $M_k(\mathbf{u})$ denote the $(k+l-1) \times k$ matrix

$$M_k(\mathbf{u}) = \left.\begin{bmatrix} u_1 & & 0 \\ \vdots & \ddots & \\ u_l & & u_1 \\ & \ddots & \vdots \\ 0 & & u_l \end{bmatrix}\right\} k+l-1.$$

It is easily checked that, for $\mathbf{x} \in \mathbb{F}^k$, $(M_k(\mathbf{u})\mathbf{x})(t) = \mathbf{u}(t)\mathbf{x}(t)$.

Furthermore, $\mathbf{e}_k \in \mathbb{F}^n$ will denote the kth vector in the standard basis of $\mathbb{F}^n$, and $\mathbf{0}_k$ will denote a zero vector of length k. If the length of the vector is clear or irrelevant we omit the subscript.

2. Inversion formula

From now on, let $T_N = [a_{i-j}]_{i,j=1}^N$ be a nonsingular skewsymmetric Toeplitz matrix and T_{N+1} any skewsymmetric $(N+1) \times (N+1)$ Toeplitz extension of T_N. Clearly, N must be even and T_{N+1} and T_{N-1} have one-dimensional nullspaces. Let $\mathbf{u} \in \mathbb{F}^{N+1}$

and $\mathbf{u}' \in \mathbb{F}^{N-1}$ be the vectors spanning these nullspaces. In [13] (see also [14]) it was shown that the vectors $\mathbf{u}$ and $\mathbf{u}'$ are symmetric.

Since T_N is nonsingular, the last component of $\mathbf{u}$ is nonzero. Therefore, we may assume that it is equal to 1. Note that the last component of $\mathbf{u}'$ might be zero.

Let r be defined by

$$r = [a_{N-1} \quad \cdots \quad a_1]\mathbf{u}'.$$

Since T_N is nonsingular, we have $r \neq 0$. It is worth to mention that the vectors

$$\frac{1}{r}\begin{bmatrix} \mathbf{u}' \\ 0 \end{bmatrix}, \quad -\frac{1}{r}\begin{bmatrix} 0 \\ \mathbf{u}' \end{bmatrix}$$

are the last and the first columns of T_N^{-1}, respectively. We introduce the (symmetric) vector

$$\mathbf{x} = \frac{1}{r}\begin{bmatrix} 0 \\ \mathbf{u}' \\ 0 \end{bmatrix} \in \mathbb{F}^{N+1}$$

which is the solution of the equation $T_{N+1}\mathbf{x} = \mathbf{e}_{N+1} - \mathbf{e}_1$.

The following is a specification of a well-known inversion formula for general Toeplitz matrices (see [10,1]) for the case of skewsymmetric matrices and was discussed in [14].

Theorem 2.1. *The inverse of T_N is given by*

$$T_N^{-1}(t,s) = \frac{\mathbf{x}(t)\mathbf{u}(s) - \mathbf{u}(t)\mathbf{x}(s)}{1 - ts}. \tag{1}$$

Formula (1) can be expressed in matrix form in many ways. Let us present one of them, which is the "classical" Gohberg-Semencul formula built from triangular Toeplitz matrices.

For a vector $\mathbf{v} = (v_i)_{i=1}^{N+1}$, let $L(\mathbf{v})$ denote the $N \times N$ lower triangular Toeplitz matrix

$$L(\mathbf{v}) = \begin{bmatrix} v_1 & & 0 \\ \vdots & \ddots & \\ v_N & \cdots & v_1 \end{bmatrix}.$$

Corollary 2.2. *The inverse of T_N is given by*

$$T_N^{-1} = L(\mathbf{x})L(\mathbf{u})^{\mathrm{T}} - L(\mathbf{u})L(\mathbf{x})^{\mathrm{T}}. \tag{2}$$

The direct application of (2) has complexity $\mathrm{O}(N^2)$, but if $\mathbb{F}$ is the field of real or complex numbers fast algorithms with complexity $\mathrm{O}(N \log N)$ can be applied. Let us mention that there are formulas for T_N^{-1} that contain only diagonal matrices and discrete

Fourier or real trigonometric transformations, which are ready for implementation (see for example [11,12] and references therein).

Note also that formula (2) can be written in terms of polynomial multiplication, and polynomial multiplication can be carried out with complexity $O(N \log N \log\log N)$ in any field (see [17] and references therein).

3. Recursion background

We are going to show some facts which will be the basis for the split algorithms developed in the next sections. Besides the (nonsingular) matrix T_N and its extension T_{N+1} we consider its central submatrices. Recall that N is even and so all central submatrices of T_N have even order. These central submatrices coincide with the leading principal submatrices $T_k = [a_{i-j}]_{i,j=1}^k$ for even k.

Let T_n be nonsingular. Then T_{n+1} has the kernel dimension one. Let $\mathbf{u}_n$ span the kernel of T_{n+1}. Since the last component of $\mathbf{u}_n$ does not vanish we may assume that it is equal to 1. As mentioned above, $\mathbf{u}_n$ is symmetric.

We introduce the numbers

$$r_j = [a_{j+n} \quad \cdots \quad a_j]\mathbf{u}_n$$

for $j = 1, \dots, N-n$, which will be called *residuals* of $\mathbf{u}_n$.

Proposition 3.1. *Let $r_1 = \cdots = r_{d-1} = 0$, $r_d \neq 0$, and $m = n + 2d$. Then $T_{n+1}, \dots, T_{m-1}$ are singular and T_m is nonsingular.*

Proof. We have

$$T_m M_{2d}(\mathbf{u}_n) = \begin{bmatrix} O_{d\times d} & -R \\ O & O \\ R^{\mathrm{T}} & O_{d\times d} \end{bmatrix}, \tag{3}$$

where R denotes the $d \times d$ upper triangular Toeplitz matrix

$$R = \begin{bmatrix} r_d & \cdots & r_{2d-1} \\ & \ddots & \vdots \\ 0 & & r_d \end{bmatrix}.$$

Hence

$$T_{n+2k+1} \begin{bmatrix} \mathbf{0}_k \\ \mathbf{u}_n \\ \mathbf{0}_k \end{bmatrix} = \mathbf{0}$$

for $k = 0, \dots, d-1$, which means that the matrices $T_{n+1}, \dots, T_{m-1}$ are singular. Furthermore, we conclude from (3) that the vectors $\mathbf{e}_1, \dots, \mathbf{e}_d$ and $\mathbf{e}_{m-d+1}, \dots, \mathbf{e}_m$ belong to the range of T_m and also to the range of T_m^{T}.

Suppose that $T_m\mathbf{v}=\mathbf{0}$. Then $\mathbf{g}^{\mathrm{T}}\mathbf{v}=0$ for all vectors from the range of T_m^{T}. Hence the first and last d components of $\mathbf{v}$ vanish, and $\mathbf{v}$ is of the form $\mathbf{v}^{\mathrm{T}}=[\mathbf{0}_d \;\; \mathbf{v}'^{\mathrm{T}} \;\; \mathbf{0}_d]^{\mathrm{T}}$, where $\mathbf{v}'$ belongs to the kernel of T_n. Since T_n is nonsingular, we conclude that $\mathbf{v}=\mathbf{0}$. Thus T_m is nonsingular. □

Besides the vector $\mathbf{u}_n$ we consider a solution $\mathbf{x}_n$ of the equation $T_{n+1}\mathbf{x}_n=\mathbf{e}_{n+1}-\mathbf{e}_1$. Since $\mathbf{u}_n^{\mathrm{T}}(\mathbf{e}_{n+1}-\mathbf{e}_1)=0$, this equation has a (non-unique) solution $\mathbf{x}$, which is symmetric, due to the centro-skewsymmetry of T_n. We introduce numbers

$$s_j=[a_{j+n} \quad \cdots \quad a_j]\mathbf{x}_n$$

for $j=0,\dots,N-n$. In particular, $s_0=1$.

Let $\mathbf{x}_m$ be a solution of the equation $T_{m+1}\mathbf{x}_m=\mathbf{e}_{m+1}-\mathbf{e}_1$ and $\mathbf{u}_m$ the vector spanning the kernel of T_{m+1} with the last component equal to 1. We show now how $\mathbf{u}_m$ and $\mathbf{x}_m$ can be computed from $\mathbf{u}_n$ and $\mathbf{x}_n$.

From (3) we conclude that

$$T_{m+1}\begin{bmatrix}\mathbf{0}_d\\ \mathbf{u}_n\\ \mathbf{0}_d\end{bmatrix}=r_d(\mathbf{e}_{m+1}-\mathbf{e}_1).$$

Thus $\mathbf{x}_m$ can be chosen as

$$\mathbf{x}_m=\frac{1}{r_d}\begin{bmatrix}\mathbf{0}_d\\ \mathbf{u}_n\\ \mathbf{0}_d\end{bmatrix}. \tag{4}$$

To find $\mathbf{u}_m$ we observe that

$$T_{m+1}M_{2d+1}(\mathbf{u}_n)=\begin{bmatrix} 0 & \cdots & -r_d & \cdots & -r_{2d}\\ \vdots & & & \ddots & \vdots\\ 0 & & & & -r_d\\ \mathbf{0} & & \mathbf{0} & & \mathbf{0}\\ r_d & & & & 0\\ \vdots & \ddots & & & \vdots\\ r_{2d} & \cdots & r_d & \cdots & 0\end{bmatrix}. \tag{5}$$

Let $\tilde{R}$ denote the $(d+1)\times(d+1)$ upper triangular Toeplitz matrix

$$\tilde{R}=\begin{bmatrix} r_d & \cdots & r_{2d}\\ & \ddots & \vdots\\ 0 & & r_d\end{bmatrix},$$

and $\mathbf{c}=(c_i)_{i=1}^{d+1}$ the solution of the triangular Toeplitz system

$$\tilde{R}^{\mathrm{T}}\mathbf{c}=\mathbf{s},$$

where $\mathbf{s}=(s_{i-1})_{i=1}^{d+1}$.

Furthermore, let $\tilde{\mathbf{c}}=\begin{bmatrix}\mathbf{c}\\ \mathbf{c}'\end{bmatrix}\in\mathbb{F}_+^{2d+1}$ be the symmetric extension of $\mathbf{c}$, $q=1/c_1$, and $\mathbf{p}=q\tilde{\mathbf{c}}$. Then we have

$$T_{m+1}\left(M_{2d+1}(\mathbf{u}_n)\mathbf{p}-q\begin{bmatrix}\mathbf{0}_d\\ \mathbf{x}_n\\ \mathbf{0}_d\end{bmatrix}\right)=\mathbf{0}.$$

By construction, the last (and the first) component of $M_{2d+1}(\mathbf{u}_n)\mathbf{p}$ equals 1. We arrived at the relation

$$\mathbf{u}_m=M_{2d+1}(\mathbf{u}_n)\mathbf{p}-q\begin{bmatrix}\mathbf{0}_d\\ \mathbf{x}_n\\ \mathbf{0}_d\end{bmatrix}. \tag{6}$$

We write relations (4) and (6) in polynomial language and arrive at the following.

Proposition 3.2. *The vectors* $\mathbf{u}_m$ *and* $\mathbf{x}_m$ *can be computed from* $\mathbf{u}_n$ *and* $\mathbf{x}_n$ *via*

$$\begin{aligned}\mathbf{u}_m(t)&=\mathbf{p}(t)\mathbf{u}_n(t)-qt^d\mathbf{x}_n(t),\\ \mathbf{x}_m(t)&=\frac{1}{r_d}\,t^d\mathbf{u}_n(t).\end{aligned}\tag{7}$$

4. Split algorithms

We discuss now the algorithms emerging from the recursion described in Proposition 3.2. First we introduce some notation. Let $n_1<\cdots<n_\ell=N$ be the integers $n\in\{1,2,\ldots,N\}$ for which T_n is nonsingular, $d_k=\frac{1}{2}(n_{k+1}-n_k)$, and let $\mathbf{u}^{(k)}$ be the vector spanning the kernel of T_{n_k+1} with last component equal to 1 and $\mathbf{x}^{(k)}$ a solution of $T_{n_k+1}\mathbf{x}^{(k)}=\mathbf{e}_{n_k+1}-\mathbf{e}_1$. The residuals $r_j^{(k)}$ and $s_j^{(k)}$ of $\mathbf{u}^{(k)}$ and $\mathbf{x}^{(k)}$ are defined by

$$r_j^{(k)}=[a_{j+n_k}\quad\cdots\quad a_j]\mathbf{u}^{(k)},\quad s_j^{(k)}=[a_{j+n_k}\quad\cdots\quad a_j]\mathbf{x}^{(k)}, \tag{8}$$

respectively, for $j=0,\ldots,N-n_k$. Clearly, $r_0^{(k)}=0$ and $s_0^{(k)}=1$.

Our aim is to find $\mathbf{u}=\mathbf{u}^{(\ell)}$ and $\mathbf{x}=\mathbf{x}^{(\ell)}$. Then the solution of a linear system $T_N\mathbf{f}=\mathbf{b}$ can be computed using the formula from Corollary 2.2 or another inversion formula.

First let us note that according to (7)

$$\mathbf{x}^{(k)}=\frac{1}{r_{d_{k-1}}^{(k-1)}}\begin{bmatrix}\mathbf{0}_{d_{k-1}}\\ \mathbf{u}^{(k-1)}\\ \mathbf{0}_{d_{k-1}}\end{bmatrix}$$

and

$$s_j^{(k)} = \frac{1}{r_{d_{k-1}}^{(k-1)}} r_{j+d_{k-1}}^{(k-1)}.$$

That means it is sufficient to compute the residuals $r_j^{(k)}$ and to construct the vectors $\mathbf{u}^{(k)}$.

For initialization we set $n_0 = 0$ and $\mathbf{u}^{(0)} = 1$. Then $r_j^{(0)} = a_j$. If $a_1 = \cdots = a_{d-1} = 0$ and $a_d \neq 0$, then $n_1 = 2d$. The vector $\mathbf{u}^{(1)}$ is the normalized solution of the homogeneous system $T_{2d+1}\mathbf{v} = 0$.

We show how this solution can be found. We form the matrix

$$\tilde{R}^{(0)} = \begin{bmatrix} a_d & \cdots & a_{2d} \\ & \ddots & \vdots \\ 0 & & a_d \end{bmatrix}.$$

Let $\mathbf{c}$ be the solution of the triangular Toeplitz system $(\tilde{R}^{(0)})^{\mathrm{T}}\mathbf{c} = \mathbf{e}_1$ and $\mathbf{v} = \begin{bmatrix} \mathbf{c} \\ \mathbf{c}' \end{bmatrix} \in \mathbb{F}^{2d+1}$ its symmetric extension. Then $T_{2d+1}\mathbf{v} = \mathbf{0}$. Hence $\mathbf{u}^{(1)} = (1/c)\mathbf{v}$, where c is the first component of $\mathbf{c}$.

We assume now that n_{k-1}, n_k, $\mathbf{u}^{(k-1)}$ and $\mathbf{u}^{(k)}$ are given. We also need some of the values $r_j^{(k-1)}$ $(j = 1, \ldots, 2d_{k-1})$ that are computed in the previous step. Now n_{k+1} and $\mathbf{u}^{(k+1)}$ are computed as follows. If $r_1^{(k)} = \cdots = r_{d-1}^{(k)} = 0$ and $r_d^{(k)} \neq 0$, then $d_k = d$, i.e. $n_{k+1} = n_k + 2d$.

We compute the numbers $r_{d_k+1}^{(k)}, \ldots, r_{2d_k}^{(k)}$ and form the matrix $\tilde{R}^{(k)}$ as

$$\tilde{R}^{(k)} = \begin{bmatrix} r_{d_k}^{(k)} & \cdots & r_{2d_k}^{(k)} \\ & \ddots & \vdots \\ 0 & & r_{d_k}^{(k)} \end{bmatrix}.$$

If $d_k > d_{k-1}$, then it will be necessary to compute also the numbers $r_j^{(k-1)}$ for $j = 2d_{k-1} + 1, \ldots, d_k + d_{k-1}$ to form the vector $\mathbf{r}'^{(k-1)} = (r_j^{(k-1)})_{j=d_{k-1}}^{d_k+d_{k-1}}$.

Let $\mathbf{c}^{(k)}$ be the solution of the triangular Toeplitz system

$$(\tilde{R}^{(k)})^{\mathrm{T}}\mathbf{c}^{(k)} = \mathbf{r}'^{(k-1)},$$

$q^{(k)} = 1/c$, where c is the first component of $\mathbf{c}^{(k)}$, and $\mathbf{p}^{(k)} = q^{(k)} \begin{bmatrix} \mathbf{c}^{(k)} \\ \mathbf{c}'^{(k)} \end{bmatrix} \in \mathbb{F}_+^{2d_k+1}$ be the symmetric extension of $q^{(k)}\mathbf{c}^{(k)}$. Then

$$\mathbf{u}^{(k+1)} = M_{2d_k+1}(\mathbf{u}^{(k)})\mathbf{p}^{(k)} - q^{(k)} \begin{bmatrix} \mathbf{0}_{d_k+d_{k-1}} \\ \mathbf{u}^{(k-1)} \\ \mathbf{0}_{d_k+d_{k-1}} \end{bmatrix}.$$

In polynomial language the recursion can be written as follows.

Theorem 4.1. *The polynomials* $\mathbf{u}^{(k)}(t)$ *satisfy the three-term recursion*

$$\mathbf{u}^{(k+1)}(t) = \mathbf{p}^{(k)}(t)\mathbf{u}^{(k)}(t) - t^{d_k+d_{k-1}}q^{(k)}\mathbf{u}^{(k-1)}(t).$$

Example 1. Consider the skewsymmetric Toeplitz matrix $T_6 = [a_{i-j}]_{i,j=1}^6$, with $(a_k)_{k=1}^5 = (1,2,3,5,6)$. Since we need also an extension of T_6 we set $a_6 = 0$. The standard setting for initialization is $n_0 = 0$, $\mathbf{u}^{(0)} = 1$ and $r_j^{(0)} = a_j$. Since $r_1^{(0)} = 1 \neq 0$ we have $d_0 = 1$ and $n_1 = n_0 + 2d_0 = 2$. We obtain $\mathbf{x}^{(1)} = [0\ \ 1\ \ 0]^{\mathrm{T}}$ and $\mathbf{u}^{(1)} = [1,-2,1]^{\mathrm{T}}$. With $\mathbf{u}^{(0)}$ and $\mathbf{u}^{(1)}$ we can start the recursion.

We compute the residuals as $r_1^{(1)} = 0$, $r_2^{(1)} = 1$. Thus $d_1 = 2$, $n_2 = n_1 + 2d_1 = 6$, and $\mathbf{x}^{(2)} = [0,0,1,-2,1,0,0]^{\mathrm{T}}$. In order to form the matrix $\tilde{R}^{(1)}$ we find that $r_3^{(1)} = -1$ and $r_4^{(1)} = -7$, and in order to form the vector $\mathbf{r}'^{(0)}$ we observe that $r_2^{(0)} = a_2 = 2$, $r_3^{(0)} = a_3 = 3$. The solution of the system $(\tilde{R}^{(1)})^{\mathrm{T}}\mathbf{c}^{(1)} = \mathbf{r}'^{(0)}$ is $\mathbf{c}^{(1)} = [1,3,13]^{\mathrm{T}}$. Hence $\mathbf{p}^{(1)} = [1,3,13,3,1]^{\mathrm{T}}$, which gives

$$\mathbf{u}^{(2)} = [1,1,8,-21,8,1,1]^{\mathrm{T}}.$$

The inverse of T_6 is now given by Corollary 2.2 with $\mathbf{x} = \mathbf{x}^{(2)}$ and $\mathbf{u} = \mathbf{u}^{(2)}$. A check shows that this really gives the inverse matrix.

Let us discuss the complexity of the algorithm emerging from Theorem 4.1. Surprisingly, the existence of singular central submatrices does not increase the complexity, in many cases it even decreases it. For simplicity we assume that all d_k are equal to d, where d is small compared with N. We neglect lower order terms. The amount for inner product calculations will be almost independent of d. We have to compute about N inner products of a symmetric and a general vector. For this $\frac{1}{2}N^2$ additions and $\frac{1}{4}N^2$ multiplications are needed. Then we have in each step $2d+1$ vector additions of symmetric vectors and $d+1$ multiplications of a symmetric vector by a scalar. This results in $\left(\frac{1}{4}+\frac{1}{8d}\right)N^2$ additions and $\left(\frac{1}{8}+\frac{1}{8d}\right)N^2$ multiplications. Thus, the overall complexity is about $\left(\frac{3}{4}+\frac{1}{8d}\right)N^2$ additions and $\left(\frac{3}{8}+\frac{1}{8d}\right)N^2$ multiplications. That means the amount decreases when d increases. In the case $d = 1$, which is the centro-nonsingular case, Theorem 4.1 is just Theorem 3.2 in [14]. In this case the complexity is $\frac{7}{8}N^2$ additions and $\frac{1}{2}N^2$ multiplications (comp. [14]).

The algorithm just described is a split Levinson-type algorithm and includes the calculation of the residuals via long inner products, which might be not convenient in parallel computing. We show now that the residuals can also be computed by a Schur-type algorithm. The Schur-type algorithm is of independent interest, since it provides a factorization, which will be described in Section 6.

We consider the full residual vectors $\mathbf{r}^{(k)} = (r_j^{(k)})_{j=1}^{N-n_k}$ and the corresponding polynomials $\mathbf{r}^{(k)}(t)$. By the definition of the integer d_k, $\tilde{\mathbf{r}}^{(k)}(t) = t^{-d_k+1}\mathbf{r}^{(k)}(t)$ is a polynomial. The monic, symmetric polynomial $\mathbf{p}^{(k)}(t)$ and $q^{(k)} \in \mathbb{F}$ have been constructed in such way that the polynomial

$$\tilde{\mathbf{r}}^{(k)}(t)\mathbf{p}^{(k)}(t) - q^{(k)}\tilde{\mathbf{r}}^{(k-1)}(t)$$

has a zero of order d_k+1 at $t=0$. According to Theorem 4.1, the remainder will give us $\mathbf{r}^{(k+1)}(t)$. Let P_m denote the projector mapping a polynomial $\sum_{j=1}^{N} p_j t^{j-1}$ $(N \geqslant m)$ to $\sum_{j=1}^{m} p_j t^{j-1}$, i.e. cutting off high powers.

Theorem 4.1 gives us immediately the following recursion formula for the residuals.

Theorem 4.2. *The polynomials* $\mathbf{r}^{(k)}(t)$ *satisfy the recursion*

$$\mathbf{r}^{(k+1)}(t) = P_{N-n_{k+1}}(t^{-2d_k}\mathbf{p}^{(k)}(t)\mathbf{r}^{(k)}(t) - t^{-d_{k-1}-d_k}q^{(k)}\mathbf{r}^{(k-1)}(t)).$$

To write this recursion in matrix form we introduce the matrix $Q^{(k)}$ by

$$Q^{(k)} = [r^{(k)}_{2d_k+i-j+1}]_{i=1}^{\mu_k}\,{}_{j=1}^{2d_k+1},$$

where $\mu_k = N - n_{k+1} = N - n_k - 2d_k$. Now we have

$$\mathbf{r}^{(k+1)} = Q^{(k)}\mathbf{p}^{(k)} - q^{(k)}\check{\mathbf{r}}^{(k-1)},$$

where $\check{\mathbf{r}}^{(k-1)} = [r^{(k-1)}_{d_k+d_{k-1}+i}]_{i=1}^{\mu_k}$.

The recursion starts with $\check{\mathbf{r}}^{(-1)} = 0$, $\mathbf{r}^{(0)} = [a_j]_{j=1}^{N}$, $\mathbf{p}^{(0)} = \mathbf{u}^{(1)}$, and $Q^{(0)} = [a_{n_1+i-j+1}]_{i=1}^{N-n_1}\,{}_{j=1}^{n_1+1}$. The vector $\mathbf{u}^{(1)}$ will be computed as described in the initialization of the Levinson-type recursion via the solution of a triangular $(d_1+1)\times(d_1+1)$ Toeplitz system.

Theorem 4.2 can be combined with Theorem 4.1 to compute $\mathbf{u}$ and $\mathbf{x}$, the parameters for the inversion formula.

5. Solution of linear systems

In this section we show how to solve a linear system

$$T_N \mathbf{f}_N = \mathbf{b}_N$$

with a nonsingular $N\times N$ skewsymmetric Toeplitz coefficient matrix T_N recursively without using the inversion formula. We use all notations that were introduced in the previous section.

Suppose that $\mathbf{b} = [b_i]_{i=1}^{N}$. We set $\mathbf{b}^{(k)} = [b_i]_{i=(1/2)(N-n_k+2)}^{(1/2)(N+n_k)} \in \mathbb{F}^{n_k}$ We consider the systems

$$T^{(k)}\mathbf{f}^{(k)} = \mathbf{b}^{(k)},$$

where $T^{(k)} = T_{n_k}$. Our aim is to compute $\mathbf{f}^{(k+1)}$ from $\mathbf{f}^{(k)}$.

Since $T^{(k+1)}$ is of the form

$$T^{(k+1)} = \begin{bmatrix} * & -B_-^{(k)} & * \\ * & T^{(k)} & * \\ * & B_+^{(k)} & * \end{bmatrix},$$

where

$$B_+^{(k)} = \begin{bmatrix} a_{n_k} & \cdots & a_1 \\ \vdots & \ddots & \vdots \\ a_{n_k+d_k-1} & \cdots & a_{d_k} \end{bmatrix}, \quad B_-^{(k)} = J_{d_k} B_+^{(k)} J_{n_k}$$

we have

$$T^{(k+1)} \begin{bmatrix} \mathbf{0} \\ \mathbf{f}^{(k)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\beta_-^{(k)} \\ \mathbf{b}^{(k)} \\ \beta_+^{(k)} \end{bmatrix},$$

where $\beta_\pm^{(k)} = B_\pm^{(k)} \mathbf{f}^{(k)}$.

As in Section 4 we obtain

$$T^{(k+1)} M_{2d_k}(\mathbf{u}^{(k)}) = \begin{bmatrix} O & -R^{(k)} \\ O & O \\ (R^{(k)})^{\mathrm{T}} & O \end{bmatrix},$$

where

$$R^{(k)} = \begin{bmatrix} r_{d_k} & \cdots & r_{2d_k-1} \\ & \ddots & \vdots \\ & & r_{d_k} \end{bmatrix}.$$

Hence we have, for $\xi_\pm^{(k)} \in \mathbb{F}^{d_k}$,

$$T^{(k+1)} \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{f}^{(k)} \\ \mathbf{0} \end{bmatrix} + M_{2d_k}(\mathbf{u}^{(k)}) \begin{bmatrix} \xi_+^{(k)} \\ \xi_-^{(k)} \end{bmatrix} \right) = \begin{bmatrix} -R^{(k)} \xi_-^{(k)} - \beta_-^{(k)} \\ \mathbf{b}^{(k)} \\ (R^{(k)})^{\mathrm{T}} \xi_+^{(k)} + \beta_+^{(k)} \end{bmatrix}. \tag{9}$$

From this relation we conclude the following.

Theorem 5.1. *Suppose that*

$$\mathbf{b}^{(k+1)} = \begin{bmatrix} \mathbf{b}_-^{(k)} \\ \mathbf{b}^{(k)} \\ \mathbf{b}_+^{(k)} \end{bmatrix},$$

where $\mathbf{b}_\pm^{(k)} \in \mathbb{F}^{d_k}$ *and* $\xi_\pm^{(k)}$ *are the solutions of*

$$(R^{(k)})^{\mathrm{T}} \xi_+^{(k)} = \mathbf{b}_+^{(k)} - \beta_+^{(k)}, \quad R^{(k)} \xi_-^{(k)} = -\mathbf{b}_-^{(k)} - \beta_-^{(k)}. \tag{10}$$

Then the solution $\mathbf{f}^{(k+1)}$ *of* $T^{(k+1)}\mathbf{f}^{(k+1)} = \mathbf{b}^{(k+1)}$ *is given by*

$$\mathbf{f}^{(k+1)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f}^{(k)} \\ \mathbf{0} \end{bmatrix} + M_{2d_k}(\mathbf{u}^{(k)}) \begin{bmatrix} \xi_+^{(k)} \\ \xi_-^{(k)} \end{bmatrix}. \tag{11}$$

For one step of the recursion one has first to compute the vectors $\beta_\pm^{(k)}$ which require the multiplication of a vector by the $d_k \times n_k$ Toeplitz matrix $B_\pm^{(k)}$, then to solve two triangular $d_k \times d_k$ Toeplitz systems (with actually the same coefficient matrix) to get $\xi_\pm^{(k)}$ and finally to apply formula (11).

The computations of the vectors $\beta_\pm^{(k)}$ require long inner product calculations which can be avoided if the full residual vectors $\tilde{\beta}_\pm^{(k)} \in \mathbb{F}^{N-n_k}$ are considered. These vectors are given by

$$T_N \begin{bmatrix} \mathbf{0} \\ \mathbf{f}^{(k)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\tilde{\beta}_-^{(k)} \\ \mathbf{b}^{(k)} \\ \tilde{\beta}_+^{(k)} \end{bmatrix}.$$

Let $Q_\pm^{(k)}$ be defined by

$$Q_+^{(k)} = [r_{2d_k+i-j+1}^{(k)}]_{i=1}^{v_k}{}_{j=1}^{d_k}, \quad Q_-^{(k)} = J_{v_k} Q_+^{(k)} J_{d_k},$$

where $v_k = \frac{1}{2}(N - n_k)$. Then we conclude from (9) that

$$\tilde{\beta}_\pm^{(k+1)} = Q_\pm^{(k)} \xi_\pm^{(k)} + (\tilde{\beta}_\pm^{(k)})',$$

where here the prime at $\tilde{\beta}_+^{(k)}$ means that the first d_k components are deleted and at $\tilde{\beta}_-^{(k)}$ that the last d_k components are deleted.

6. Generalized ZW-factorization

Like the classical Schur algorithm for symmetric Toeplitz matrices is related to the LU-factorization of the matrix and the classical Levinson algorithm related to a UL-factorization of its inverse, the split Schur algorithm is related to a ZW-factorization of the matrix and the split Levinson algorithm to a WZ-factorizaton of the inverse. In [14] the latter factorizations were investigated for skewsymmetric Toeplitz matrices. It was shown that centro-nonsingular skewsymmetric matrices admit a ZW-factorization in which the factors possess some additional symmetry properties. We are going to generalize this result to arbitrary nonsingular skewsymmetric Toeplitz matrices. The factorization will lead to the possibility to solve a linear system by a pure Schur-type algorithm.

To be more precise, let us recall some concepts. A matrix $A = [a_{ij}]_{i,j=1}^n$ is called a *W-matrix* if $a_{ij} = 0$ for all (i,j) for which $i > j$ and $i + j > n$ or $i < j$ and $i + j \leqslant n$.

The matrix A will be called a *unit W-matrix* if, in addition, $a_{ii}=1$ for $i=1,\dots,n$ and $a_{i,n+1-i}=0$ for $i\neq(n+1)/2$. The transpose of a W-matrix is called a *Z-matrix*. A matrix which is both a Z- and a W-matrix will be called an *X-matrix*. The names come from the shapes of the set of all possible positions for nonzero entries, viz.

$$W=\begin{bmatrix}\bullet & & & & & \bullet\\ \bullet & \circ & & & \circ & \bullet\\ \bullet & \circ & \circ & \circ & \circ & \bullet\\ \bullet & \circ & \bullet & \bullet & \circ & \bullet\\ \bullet & \bullet & & & \bullet & \bullet\\ \bullet & & & & & \bullet\end{bmatrix},\quad Z=\begin{bmatrix}\bullet & \bullet & \bullet & \bullet & \bullet & \bullet\\ & \circ & \circ & \circ & \bullet & \\ & & \circ & \bullet & & \\ & & \bullet & \circ & & \\ & \bullet & \circ & \circ & \circ & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet\end{bmatrix}.$$

A unit Z- or W-matrix is obviously nonsingular and a linear system with such a coefficient matrix can be solved by back substitution with $n^2/2$ additions and $n^2/2$ multiplications.

A representation $A=ZXW$ in which Z is a unit Z-matrix, W is a unit W-matrix, and X a nonsingular X-matrix is called ZW-factorization. Analogously WZ-factorization is defined. A admits a ZW-factorization if and only if A is centro-nonsingular. Under the same condition A^{-1} admits a WZ-factorization.

That means if A is not centro-nonsingular, then no such a factorization exists and a generalization is not at hand. We show now that, nevertheless, in the special case of a skewsymmetric Toeplitz matrix there is a natural generalization of the factorization result in [14].

We introduce $N\times d_k$ matrices $W_{\pm}^{(k)}$ by

$$W_{-}^{(k)}=\begin{bmatrix}O_{\nu_k\times d_k}\\ M_{d_k}(\mathbf{u}^{(k)})\\ O_{d_k\times d_k}\\ O_{\nu_k\times d_k}\end{bmatrix},\quad W_{+}^{(k)}=\begin{bmatrix}O_{\nu_k\times d_k}\\ O_{d_k\times d_k}\\ M_{d_k}(\mathbf{u}^{(k)})\\ O_{\nu_k\times d_k}\end{bmatrix},$$

where $\nu_k=\frac{1}{2}(N-n_{k+1})$, and form the matrix

$$W=[W_{-}^{(\ell-1))}\quad\cdots\quad W_{-}^{(0)}W_{+}^{(0)}\quad\cdots\quad W_{+}^{(\ell-1)}]. \tag{12}$$

Recall that $\mathbf{u}^{(0)}=1$, $n_0=0$. Obviously, W is a centrosymmetric unit W-matrix.

We have

$$T_N W_{-}^{(k)}=\begin{bmatrix}-S_{-}^{(k)}\\ O_{(n_{k+1}-d_k)\times d_k}\\ (R^{(k)})^{\mathrm{T}}\\ S_{+}^{(k)}\end{bmatrix},\quad T_N W_{+}^{(k)}=\begin{bmatrix}-\hat{S}_{+}^{(k)}\\ -R^{(k)}\\ O_{(n_{k+1}-d_k)\times d_k}\\ \hat{S}_{-}^{(k)}\end{bmatrix},$$

where

$$S_{+}^{(k)}=[r_{2d_k+i-j}^{(k)}]_{i=1}^{\nu_k}{}_{j=1}^{d_k},\quad S_{-}^{(k)}=[r_{\nu_k-i+j}^{(k)}]_{i=1}^{\nu_k}{}_{j=1}^{d_k},$$

$\hat{S}_{\pm}^{(k)} = J_{v_k} S_{\pm}^{(k)} J_{d_k}$. We set $r^{(k)} = r_{d_k}^{(k)}$,

$$Z_{+}^{(k)} = \frac{1}{r^{(k)}} T_N W_{-}^{(k)}, \quad Z_{-}^{(k)} = -\frac{1}{r^{(k)}} T_N W_{+}^{(k)},$$

and form the matrix

$$Z = [Z_{-}^{(\ell-1)} \quad \cdots \quad Z_{-}^{(0)} Z_{+}^{(0)} \quad \cdots \quad Z_{+}^{(\ell-1)}]. \tag{13}$$

Then Z is a centrosymmetric unit Z-matrix. Furthermore,

$$T_N W = ZX,$$

where X is the skewsymmetric block antidiagonal matrix

$$X = \begin{bmatrix} 0 & & & & & -r^{(\ell-1)} I_{d_{\ell-1}} \\ & & & & \iddots & \\ & & & -r^{(0)} I_{d_0} & & \\ & & r^{(0)} I_{d_0} & & & \\ & \iddots & & & & \\ r^{(\ell-1)} I_{d_{\ell-1}} & & & & & 0 \end{bmatrix}. \tag{14}$$

This leads to the following.

Theorem 6.1. *A nonsingular skewsymmetric Toeplitz matrix and its inverse admit representations*

$$T_N = ZXZ^{\mathrm{T}}, \quad T_N^{-1} = WX^{-1}W^{\mathrm{T}},$$

where Z is a centrosymmetric Z-matrix given by (13), *W is a centrosymmetric W-matrix given by* (12), *and X is a skewsymmetric block antidiagonal matrix given by* (14).

Example 2. Let us illustrate the factorizations for the example of a nonsingular skew-symmetric Toeplitz matrix $T_6 = [a_{i-j}]_{i,j=1}^{6}$ with $a_1 \neq 0$ for which T_4 is singular. That means we have $n_1 = 2$ and $N = n_2 = 6$. Let $\mathbf{u}^{(1)} = [1 \ u \ 1]^{\mathrm{T}}$ span the nullspace of T_3. Then the factors of the generalized ZW-factorization of T_6 and generalized WZ-factorization of T_6^{-1} are given by

$$W = \left[\begin{array}{cc|c|c|cc} 1 & & & & & 0 \\ u & 1 & & & 0 & 0 \\ 1 & u & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & u & 1 \\ 0 & 0 & & & 1 & u \\ 0 & & & & & 1 \end{array}\right], \quad Z = \left[\begin{array}{cc|c|c|cc} 1 & \frac{r_3}{r_2} & \frac{a_3}{a_1} & -\frac{a_2}{a_1} & 0 & 0 \\ & 1 & \frac{a_2}{a_1} & -1 & 0 & \\ & & 1 & 0 & & \\ & & 0 & 1 & & \\ & 0 & -1 & \frac{a_2}{a_1} & 1 & \\ 0 & 0 & -\frac{a_2}{a_1} & \frac{a_3}{a_1} & \frac{r_3}{r_2} & 1 \end{array}\right],$$

$$X = \left[\begin{array}{cc|c|c|cc} & & & & -r_2 & 0 \\ & & & & 0 & -r_2 \\ & & & -a_1 & & \\ & & a_1 & & & \\ r_2 & 0 & & & & \\ 0 & r_2 & & & & \end{array}\right],$$

where $r_2 = a_4 + a_3u + a_2$ and $r_3 = a_5 + a_4u + a_3$.

Let us point out that the factorization of T_N can be computed with the help of the Schur-type algorithm emerging from Theorem 4.2. and the factorization of T_N^{-1} with the help of the Levinson-type algorithm emerging from Theorem 4.1. Thus these algorithms can be used to solve linear systems via factorization and back substitution or matrix multiplication, respectively.

7. Concluding remarks

The algorithms described in the previous sections lead to several methods for solving a linear system $T_N\mathbf{f} = \mathbf{b}$ with a nonsingular, skewsymmetric Toeplitz coefficient matrix. There are three possibilities, namely (a) via inversion formula, (b) via direct recursion, and (c) via factorization. For each possibility there is a Levinson-type and a Schur-type version. That means we have six methods. In [14] these six methods (and two more) are described in detail and compared from the view point of complexity in sequential processing in the centro-nonsingular case. In the general case complexity matters are more complicated, since the complexity heavily depends on the rank profile but the comparison will give, in principle, the same result.

It turned out that the Levinson-type algorithm combined with the inversion formula is the most efficient one from the complexity point of view, provided that for matrix-vector multiplication a fast algorithm is used. If it is carried out in the classical way, then direct recursion and factorization are preferable.

Let us point out that complexity is not the only criterion for estimating the performance of an algorithm. In floating point arithmetics stability is an important issue. It is well known that, as a rule, Schur-type algorithms are more stable than Levinson-type algorithms (see [2]). From this point of view a solution via ZW-factorization and back substitution might be preferable over the other methods. Furthermore, all Schur-type versions are preferable in parallel computing, since they avoid inner product calculations.

References

[1] D.A. Bini, V.Y. Pan, Matrix and Polynomial Computations 1: Fundamental Algorithms, Birkhäuser Verlag, Basel, Boston, Berlin, 1994.

[2] R.P. Brent, Stability of fast algorithms for structured linear systems, in: T. Kailath, A.H. Sayed (Eds.), Fast Reliable Algorithms for Matrices with Structure, SIAM, Philadelphia, 1999.
[3] P. Delsarte, Y. Genin, The split Levinson algorithm, IEEE Trans. Acoust. Speech Signal Process. ASSP-34 (1986) 470–477.
[4] P. Delsarte, Y. Genin, On the splitting of classical algorithms in linear prediction theory, IEEE Trans. Acoust. Speech Signal Process. ASSP-35 (1987) 645–653.
[5] C.J. Demeure, Bowtie factors of Toeplitz matrices by means of split algorithms, IEEE Trans. Acoust. Speech Signal Process. ASSP-37 (10) (1989) 1601–1603.
[6] D.J. Evans, M. Hatzopoulos, A parallel linear systems solver, Internat. J. Comput. Math. 7 (3) (1979) 227–238.
[7] G. Heinig, Inversion of Toeplitz and Hankel matrices with singular sections, Wiss. Zeitschr. d. TH Karl-Marx-Stadt 25 (3) (1983) 326–333.
[8] G. Heinig, Chebyshev-Hankel matrices and the splitting approach for centrosymmetric Toeplitz-plus-Hankel matrices, Linear Algebra Appl. 327 (1–3) (2001) 181–196.
[9] G. Heinig, A. Al-Rashidi, Fast algorithms for skewsymmetric Toeplitz matrices based on recursion of fundamental systems, in preparation.
[10] G. Heinig, K. Rost, Algebraic Methods for Toeplitz-like Matrices and Operators, Birkhäuser Verlag, Basel, Boston, Stuttgart, 1984.
[11] G. Heinig, K. Rost, DFT representations of Toeplitz-plus-Hankel Bezoutians with application to fast matrix-vector multiplication, Linear Algebra Appl. 284 (1998) 157–175.
[12] G. Heinig, K. Rost, Efficient inversion formulas for Toeplitz-plus-Hankel matrices using trigonometric transformations, in: V. Olshevsky (Ed.), Structured Matrices in Mathematics, Computer Science, and Engineering, Vol. 2, AMS-Series, Providence, RI, Contemp. Math. 281 (2001) 247–264.
[13] G. Heinig, K. Rost, Centrosymmetric and centro-skewsymmetric Toeplitz matrices and Bezoutians, Linear Algebra Appl. 343–344 (2001) 195–209.
[14] G. Heinig, K. Rost, Fast algorithms for skewsymmetric Toeplitz matrices, Oper. Theory Adv. Appl. 135 (2002) 193–208.
[15] G. Heinig, K. Rost, Split algorithms for symmetric Toeplitz matrices with arbitrary rank profile, Numer. Linear Algebra Appl., to appear.
[16] A. Melman, A two-step even-odd split Levinson algorithm for Toeplitz systems, Linear Algebra Appl. 338 (2001) 219–237.
[17] V.Y. Pan, Structured Matrices and Polynomials, Birkhäuser Verlag, Boston, Springer, New York, 2001.
[18] S. Chandra Sekhara Rao, Existence and uniqueness of *WZ* factorization, Parallel Comp. 23 (8) (1997) 1129–1139.
[19] V.V. Voevodin, E.E. Tyrtyshnikov, Numerical Processes with Toeplitz Matrices, Nauka, Moscow, 1987 (in Russian).