LINEAR ALGEBRA AND ITS APPLICATIONS

ELSEVIER Linear Algebra and its Applications 343–344 (2002) 419–450

www.elsevier.com/locate/laa

A modification of the Dewilde–van der Veen method for inversion of finite structured matrices[☆]

Y. Eidelman[*], I. Gohberg

School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel

Received 30 July 2000; accepted 14 April 2001

Submitted by V. Olshevsky

Abstract

We study a class of block structured matrices $R = \{R_{ij}\}_{i,j=1}^{N}$ with a property that the solution of the corresponding system $Rx = y$ of linear algebraic equations may be performed for $\mathrm{O}(N)$ arithmetic operations. In this paper for finite invertible matrices we analyze in detail factorization and inversion algorithms. These algorithms are related to those suggested by P.M. Dewilde and A.J. van der Veen (Time-varying Systems and Computations, Kluwer Academic Publishers, New York, 1998) for a class of finite and infinite matrices with a small Hankel rank. The algorithms presented here are more transparent and are a modification of the algorithms from the above reference. The approach and the proofs are essentially different from those in the above-mentioned reference. The paper contains also analysis of complexity and results of numerical experiments. © 2002 Elsevier Science Inc. All rights reserved.

Keywords: Structured matrices; Linear complexity algorithms; Inversion algorithms; Factorization algorithms; Solution of linear equations

1. Introduction

We study a class of block structured matrices $R = \{R_{ij}\}_{i,j=1}^{N}$ with a property that the solution of the corresponding system $Rx = y$ of linear algebraic equations

☆ This research was supported in part by The Israel Science Foundation founded by The Israel Academy of Sciences and Humanities.

* Corresponding author.

E-mail addresses: eideyu@math.tau.ac.il (Y. Eidelman), gohberg@math.tau.ac.il (I. Gohberg).

may be performed for $\mathrm{O}(N)$ arithmetic operations. As is well known, the standard methods for the solution of linear systems, as for instance the Gaussian elimination, require $\mathrm{O}(N^3)$ operations. For some classes of structured matrices such as Toeplitz, Cauchy, Vandermonde and others it takes $\mathrm{O}(N^2)$ operations. We consider a special class of matrices which admit linear complexity algorithms. These classes of matrices appear in different problems in which discretizations of kernels which are Green functions of differential equations are used, as well as in signal processing (Kalman filter, see [11,12]). More precisely we consider block matrices whose entries are specified as follows:

$$R_{ij} = \begin{cases} p_i a_{i-1} \cdots a_{j+1} q_j, & 1 \leqslant j < i \leqslant N, \\ d_i, & 1 \leqslant i = j \leqslant N, \\ g_i b_{i+1} \cdots b_{j-1} h_j, & 1 \leqslant i < j \leqslant N. \end{cases} \tag{1.1}$$

Here p_i $(i = 2, \ldots, N)$, q_j $(j = 1, \ldots, N-1)$, and a_k $(k = 2, \ldots, N-1)$ are matrices of sizes $m_i \times r'_{i-1}$, $r'_j \times n_j$, and $r'_k \times r'_{k-1}$, respectively; these elements are said to be *lower generators of the matrix R* with *orders* r'_k $(k = 1, \ldots, N-1)$. The elements g_i $(i = 1, \ldots, N-1)$, h_j $(j = 2, \ldots, N)$, and b_k $(k = 2, \ldots, N-1)$ are matrices of sizes $m_i \times r''_i$, and $r''_{j-1} \times n_j$, and $r''_{k-1} \times r''_k$, respectively; these elements are said to be *upper generators of the matrix R* with *orders* r''_k $(k = 1, \ldots, N-1)$. The matrices d_k $(k = 1, \ldots, N)$ of sizes $m_k \times n_k$ are said to be *diagonal entries* of the matrix R.

The class which we consider contains at least three well-known classes: diagonal plus semiseparable matrices, band matrices and unitary Hessenberg matrices. For band matrices linear complexity inversion algorithms are presented in various papers and monographs (see for instance [13]). For diagonal plus semiseparable matrices, probably for the first time a linear complexity inversion algorithm was suggested by Gohberg et al. in [11,12] with the assumption that a matrix is strongly regular, i.e., all its principal leading minors are non-vanishing. Another approach to inversion of diagonal plus semiseparable matrices which is based on the system theory was suggested by Gohberg and Kaashoek in [10]. Using the Gohberg–Kaashoek inversion formula the authors in [3,4] obtained inversion formulas and linear complexity inversion algorithms for diagonal plus semiseparable matrices of general form. Analysis of representations obtained in [3,4] showed that inverse to a diagonal plus semiseparable matrix is in general a matrix of the form (1.1), i.e., it belongs to a more general class. We started the detailed study of this class in our paper [5]. In this paper and also in [6,7] we developed linear complexity inversion algorithms which are based on computation of generators of the inverse matrix.

Another approach which is based on factorization representations was suggested by Dewilde and van der Veen. Dewilde and van der Veen in [1] (see also [2]) considered a class of finite and infinite matrices with a small Hankel rank. In particular in [1] a method for factorization and inversion of such matrices was suggested. In this paper we consider only the case of finite invertible matrices. This case is analyzed in detail, and a systematical description of factorization and inversion algorithms is presented. These algorithms are more transparent and are modification and simplification of the algorithms suggested in [1]. The approach and the proofs are essentially different from those in [1]. It allows us to avoid the requirement of the minimality of generators which represents numerical difficulties in using of the algorithm. The paper contains also analysis of the complexity and results of numerical experiments.

We will now explain the main idea of our derivation of the algorithms. Assume that the matrix R has a block row of the form

$$R^k = \begin{pmatrix} pq & A & gh \end{pmatrix},$$

where p is a matrix of the sizes $m \times n,\ m > n$ and m, n are small numbers, and the diagonal block A and the matrix g have small sizes also. By an orthogonal transformation V one can transform the matrix p to the form

$$Vp = X_1 = \begin{pmatrix} X \\ 0_{(m-n)\times n} \end{pmatrix}.$$

For the whole row we obtain

$$VR^k = \begin{pmatrix} X_1 & VA & (Vg)h \end{pmatrix}.$$

It means that for a small number of operations we obtain a large number of zeros in one part of the matrix and elements of the same structure in another part. The structure of quasiseparable matrices allows us to apply such transformations successively and to derive an algorithm on this basis. The suggested derivation of the algorithms is completely different from the derivation in [1] but leads to the same results. The same idea was used in the particular case of diagonal plus semiseparable matrices by Mastronardi et al. in their recent paper [14].

The paper consists of eight sections. Section 1 is the introduction. In Section 2 we give definitions and some auxiliary relations. In Sections 3 and 4 we consider algorithms for computing of the product of a matrix by a vector and solution of triangular systems via generators. In Section 5 we obtain some factorization relations for triangular matrices which are used for derivation of the main algorithms of the paper. In Section 6 we present the detailed description of the inversion method. This section contains a general description, two factorization algorithms which are a basis for the method, an application of the obtained results to the solution of linear systems, and analysis of complexity. In Section 7 we consider separately the case of matrices with scalar entries. In Section 8 we present results of numerical experiments.

2. Definitions

Let $\{a_k\}$, $k = 1, \ldots, N$, be a family of matrices of sizes $r_k \times r_{k-1}$. For positive integers $i, j,\ i > j$, define the operation $a^{\times}_{ij}$ as follows: $a^{\times}_{ij} = a_{i-1} \cdots a_{j+1}$ for $i > j+1$, $a^{\times}_{j+1,j} = I_{r_j}$.

Let $\{b_k\}$, $k = 1, \ldots, N$, be a family of matrices of sizes $r_{k-1} \times r_k$. For positive integers $i, j,\ j > i$, define the operation $b^{\times}_{ij}$ as follows: $b^{\times}_{ij} = b_{i+1} \cdots b_{j-1}$ for $j > i + 1$, $b^{\times}_{i,i+1} = I_{r_i}$.

It is easy to see that

$$a^{\times}_{i+1,j} = a_i a^{\times}_{i,j} \tag{2.1}$$

and

$$b^{\times}_{i-1,j} = b_i b^{\times}_{i,j}. \tag{2.2}$$

Let $R = \{R_{ij}\}^{N}_{i,j=1}$ be a matrix with block entries R_{ij} of sizes $m_i \times n_j$. Assume that the entries of this matrix are represented in the form

$$R_{ij} = \begin{cases} p_i a^{\times}_{ij} q_j, & 1 \leqslant j < i \leqslant N, \\ d_i, & 1 \leqslant i = j \leqslant N, \\ g_i b^{\times}_{ij} h_j, & 1 \leqslant i < j \leqslant N. \end{cases} \tag{2.3}$$

Here $p_i\ (i = 2, \ldots, N)$, $q_j\ (j = 1, \ldots, N-1)$, and $a_k\ (k = 2, \ldots, N-1)$ are matrices of sizes $m_i \times r'_{i-1}$, $r'_j \times n_j$, and $r'_k \times r'_{k-1}$, respectively; these elements are said to be *lower generators of the matrix R* with *orders* $r'_k\ (k = 1, \ldots, N-1)$. The elements $g_i\ (i = 1, \ldots, N-1)$, $h_j\ (j = 2, \ldots, N)$, and $b_k\ (k = 2, \ldots, N-1)$ are matrices of sizes $m_i \times r''_i$, $r''_{j-1} \times n_j$, and $r''_{k-1} \times r''_k$, respectively; these elements are said to be *upper generators of the matrix R* with *orders* $r''_k\ (k = 1, \ldots, N-1)$. The matrices $d_k\ (k = 1, \ldots, N)$ of sizes $m_k \times n_k$ are said to be *diagonal entries* of the matrix R. We define also orders of generators $r'_k,\ r''_k$ for $k = -1,\ 0,\ N,\ N+1$ setting them to be zeros.

Formally, we use some calculation rules with matrices that have blocks with dimension zero. Aside from obvious rules, the product of an "empty" matrix of dimension $m \times 0$ and an empty matrix of dimension $0 \times n$ is a matrix of dimension $m \times n$ with all elements equal to 0. All further rules of block matrix multiplication remain consistent. Such operations are used in MATLAB.

The class of matrices which we are considering contains at least three well-known classes of structured matrices: band matrices, diagonal plus semiseparable matrices, and unitary Hessenberg matrices. Assume that blocks of the matrix R are square and the orders of generators are constant: $r'_k = r_1,\ r''_k = r_2\ (k = 1, \ldots, N-1)$. If in (2.3) $a_k = a,\ b_k = b\ (k = 2, \ldots, N-1)$ and $a^{r_1} = 0,\ b^{r_2} = 0$, then R is a band matrix. If $a_k = I_{r_1},\ b_k = I_{r_2}\ (k = 2, \ldots, N-1)$, then we obtain a diagonal plus semiseparable matrix. Assume now that in (2.3) $m_1 = 1,\ n_1 = 0,\ m_k = n_k = 1\ (k = 2, \ldots, N-1),\ m_N = 0,\ n_N = 1$ and $r'_k = 0\ (k = 1, \ldots, N-1)$. Then R is an upper Hessenberg matrix. If moreover $r''_k = 1\ (k = 1, \ldots, N-1)$ with some additional assumptions (see Section 5 below), we obtain a unitary Hessenberg matrix.

Generators of a matrix $R = \{R_{ij}\}^{N}_{i,j=1}$ may be obtained by its entries as follows:

$$p_i = \begin{bmatrix} R_{i1} & \cdots & R_{i,i-1} & * & \cdots & * \end{bmatrix}, \quad 2 \leqslant i \leqslant N,$$

$$q_j = \begin{bmatrix} 0_{n_i \times n_j} \\ \cdots \\ I_{n_j} \\ \cdots \\ 0_{n_N \times n_j} \end{bmatrix}, \quad 1 \leqslant j \leqslant N-1,$$

$$g_i = \begin{bmatrix} * & * & \cdots & R_{i,i+1} & \cdots & R_{i,N} \end{bmatrix}, \quad 1 \leqslant i \leqslant N-1,$$

$$h_j = \begin{bmatrix} 0_{n_i \times n_j} \\ \cdots \\ I_{n_j} \\ \cdots \\ 0_{n_N \times n_j} \end{bmatrix}, \quad 2 \leqslant j \leqslant N,$$

$$a_k = b_k = I_{\tilde{N}},\ k = 2, \ldots, N-1, \quad \tilde{N} = \sum_{i=1}^{N} n_i.$$

Such defined generators have orders $\tilde{N}$ which are not the minimal. Generators with minimal orders may be obtained by entries of a matrix for $\mathrm{O}(N^3)$ operations using an algorithm suggested in [1, p. 56]. In the case of a unitary Hessenberg matrix this cost may be reduced to $\mathrm{O}(N^2)$ operations (see [8]). If generators of a matrix are given, then one can obtain by them generators with minimal orders using an algorithm suggested in [1, p. 101]. Next we assume that generators of a matrix are given and their orders are essentially less than sizes of a matrix.

Let $R(k,:)$ be the kth block row of the matrix R. From the definition of generators it directly follows that

$$\begin{aligned} R(1,:) &= \begin{pmatrix} d_1 & g_1 H_2 \end{pmatrix}, \\ R(k,:) &= \begin{pmatrix} p_k Q_{k-1} & d_k & g_k H_{k+1} \end{pmatrix}, \\ R(N,:) &= \begin{pmatrix} p_N Q_{N-1} & d_N \end{pmatrix}, \end{aligned} \tag{2.4}$$

where matrices $Q_k,\ H_k$ are defined as follows:

$$Q_1 = q_1; \quad Q_k = \begin{pmatrix} a_k \cdots a_2 q_1 & a_k \cdots a_3 q_2 & \ldots & a_k q_{k-1} & q_k \end{pmatrix}, \\ 2 \leqslant k \leqslant N-1; \tag{2.5}$$

$$H_N = h_N, \quad H_k = \begin{pmatrix} h_k & b_k h_{k+1} & b_k b_{k+1} h_{k+2} & \ldots & b_k \cdots b_N h_N \end{pmatrix}, \\ N-1 \geqslant k \geqslant 2. \tag{2.6}$$

It is easy to see that these matrices may be defined equivalently via recursive relations

$$Q_1 = q_1; \quad Q_k = \begin{pmatrix} a_k Q_{k-1} & q_k \end{pmatrix}, \quad 2 \leqslant k \leqslant N-1; \tag{2.7}$$

$$H_N = h_N, \quad H_k = \begin{pmatrix} h_k & b_k H_{k+1} \end{pmatrix}, \quad N-1 \geqslant k \geqslant 2. \tag{2.8}$$

Define the QR factorization of a matrix A of sizes $m \times n$ as the representation $A = VX$, where V is a unitary $m \times m$ matrix and X is a matrix of sizes $m \times n$. In the case $m > n$ the matrix X has the form $X = \binom{X_0}{0}$, where X_0 is a matrix of sizes $n \times n$. Such a factorization is computed using the standard MATLAB function.

Complexity of computations is expressed via a number of flops, i.e., arithmetic operations of the form $a \pm bc,\ a \pm b/c$. Submatrices are indicated in MATLAB style, i.e., for a matrix $A, A(m:n, t:s)$ selects block rows m to n of block columns t to s, and a colon without an index range selects all the rows and columns.

3. Multiplication by a vector

Let $R = \{R_{ij}\}_{i,j=1}^{N}$ be a matrix with block entries R_{ij} of sizes $m_i \times n_j$ with given lower generators $p_i\ (i = 2, \ldots, N),\ q_j (j = 1, \ldots, N-1),\ a_k (k = 2, \ldots, N-1)$ of orders $r'_k\ (k = 1, \ldots, N-1)$, upper generators $g_i\ (i = 1, \ldots, N-1),\ h_j\ (j = 2, \ldots, N),\ b_k\ (k = 2, \ldots, N-1)$ of orders $r''_k\ (k = 1, \ldots, N-1)$ and diagonal entries $d_k\ (k = 1, \ldots, N)$. The multiplication of the matrix by a vector may be performed as follows. Let $x = \operatorname{col}(x_i)_{i=1}^{N}$ be a vector with column coordinates x_i of sizes n_i. The product $y = Rx$ of the matrix R by the vector x is found as $y = y^{\mathrm{L}} + y^{\mathrm{D}} + y^{\mathrm{U}}$, where $y^{\mathrm{L}} = R_{\mathrm{L}}x$, $y^{\mathrm{D}} = R_{\mathrm{D}}x$, $y^{\mathrm{U}} = R_{\mathrm{U}}x$, and $R_{\mathrm{L}},\ R_{\mathrm{D}}$ and R_{U} are the corresponding strictly lower triangular, diagonal and strictly upper triangular parts of the matrix R.

For y^{L} we have $y_1^{\mathrm{L}} = 0$ and for $i \geqslant 2$ using the first of relations (2.3) we obtain

$$y_i^{\mathrm{L}} = \sum_{j=1}^{i-1} R_{ij} x_j = \sum_{j=1}^{i-1} p_i a_{ij}^{\times} q_j x_j = p_i z_i,$$

where

$$z_i = \sum_{j=1}^{i-1} a_{i,j}^{\times} q_j x_j.$$

From equalities (2.1) and $a_{i+1,i}^{\times} = I$ it follows that z_i satisfies the recursive relations

$$z_{i+1} = \sum_{j=1}^{i} a_{i+1,j}^{\times} q_j x_j = a_i \sum_{j=1}^{i-1} a_{ij}^{\times} q_j x_j + a_{i+1,i}^{\times} q_i x_i = a_i z_i + q_i x_i.$$

For y^{U} we have $y_N^{\mathrm{U}} = 0$ and for $i \leqslant N-1$ using the third of relations (2.3) we obtain

$$y_i^{\mathrm{U}} = \sum_{j=i+1}^{N} R_{ij} x_j = \sum_{j=i+1}^{N} g_i b_{ij}^{\times} h_j x_j = g_i w_i,$$

where

$$w_i = \sum_{j=i+1}^{N} b^{\times}_{i,j} h_j x_j. \tag{3.1}$$

From equalities (2.2) and $b^{\times}_{i-1,i} = I$ it follows that w_i satisfies the recursive relations

$$\begin{aligned} w_{i-1} &= \sum_{j=i}^{N} b^{\times}_{i-1,j} h_j x_j \\ &= b_i \sum_{j=i+1}^{N} b^{\times}_{ij} h_j x_j + b^{\times}_{i-1,i} h_i x_i \\ &= b_i w_i + h_i x_i. \end{aligned} \tag{3.2}$$

For y^{D} it is obvious that $y^{\mathrm{D}}_i = d_i x_i,\ i = 1, \ldots, N$.

From these relations we obtain the following algorithm for computing the product $y = Rx$.

Algorithm 3.1.

1. Start with $y^{\mathrm{L}}_1 = 0,\ z_2 = q_1 x_1,\ y^{\mathrm{L}}_2 = p_2 z_2$ and for $i = 3, \ldots, N$ compute recursively

   $$z_i = a_{i-1} z_{i-1} + q_{i-1} x_{i-1},$$

   $$y^{\mathrm{L}}_i = p_i z_i.$$
2. Compute for $i = 1, \ldots, N$

   $$y^{\mathrm{D}}_i = d_i x_i.$$
3. Start with $y^{\mathrm{U}}_N = 0,\ w_{N-1} = h_N x_N,\ y^{\mathrm{U}}_{N-1} = g_{N-1} w_{N-1}$ and for $i = N-2, \ldots, 1$ compute recursively

   $$w_i = b_{i+1} w_{i+1} + h_{i+1} x_{i+1},$$

   $$y^{\mathrm{U}}_i = g_i w_i.$$
4. Compute vector y

   $$y = y^{\mathrm{L}} + y^{\mathrm{D}} + y^{\mathrm{U}}.$$

In this algorithm, computation of the products $a_{i-1} z_{i-1},\ q_{i-1} x_{i-1},\ p_i z_i,\ d_i x_i,\ b_{i+1} w_{i+1},\ h_{i+1} x_{i+1}$, and $g_i w_i$ costs, respectively, $r'_{i-1} r'_{i-2},\ r'_{i-1} n_{i-1},\ m_i r'_{i-1},\ m_i n_i,\ r''_i r''_{i+1},\ r''_i n_{i+1}$, and $m_i r''_i$ flops. Hence the total complexity of Algorithm 3.1 is expressed as follows:

$$c = \sum_{k=1}^{N} \left[m_k (r'_{k-1} + r''_k) + n_{k-1} r'_{k-1} + n_{k+1} r''_k + r'_{k-1} r'_{k-2} + r''_k r''_{k+1} + m_k n_k \right]. \tag{3.3}$$

If the sizes of a matrix $m_k,\ n_k$ and the orders of its generators $r'_k,\ r''_k$ are bounded by the numbers *m* and *r*, respectively, we obtain the estimate

$$c \leqslant N(4mr + 2r^2 + m^2).$$

In this case multiplication of a matrix by a vector costs $\mathrm{O}(N)$ operations in contrast to $\mathrm{O}(N^2)$ for a matrix of general form.

4. Solution of triangular systems

Let R be a block upper triangular matrix $R = \{R_{ij}\}_{i,j=1}^N$ with block entries R_{ij} of sizes $n_i \times n_j$ with given upper generators $g_i\ (i = 1, \dots, N-1),\ h_j\ (j = 2, \dots, N),\ b_k\ (k = 2, \dots, N-1)$ of orders $\rho'_k\ (k = 1, \dots, N-1)$ and invertible diagonal entries $d_k\ (k = 1, \dots, N)$. An algorithm similar to Algorithm 3.1 can be obtained for the solution of the system $Rx = y$. Solution of an upper triangular system is obtained as follows:

$$x_N = R_{NN}^{-1} y_N, \quad x_i = R_{ii}^{-1}\left(y_i - \sum_{j=i+1}^{N} R_{ij} x_j\right),\ i = N-1, \dots, 1.$$

Using the second and the third relations from (2.3) we obtain

$$x_N = d_N^{-1} y_N$$

and

$$\begin{aligned} x_i &= d_i^{-1}\left(y_i - \sum_{j=i+1}^{N} g_i b_{ij}^{\times} h_j x_j\right) \\ &= d_i^{-1}(y_i - g_i w_i), \quad i = N-1, \dots, 1, \end{aligned}$$

where the auxiliary variable w_k is given by (3.1). From (3.2) it follows that w_k satisfies the recursive relation $w_k = b_{k+1} w_{k+1} + h_{k+1} x_{k+1},\ k = N-2, \dots, 1$.

Thus we obtain the following algorithm.

Algorithm 4.1.

1. Start with $x_N = d_N^{-1} y_N,\ \ w_{N-1} = h_N x_N,\ x_{N-1} = d_{N-1}^{-1}(y_{N-1} - g_{N-1} w_{N-1})$.
2. For $i = N-2, \dots, 1$ compute recursively

$$w_i = b_{i+1} w_{i+1} + h_{i+1} x_{i+1},$$

$$x_i = d_i^{-1}(y_i - g_i w_i).$$

In this algorithm, computation of the products $b_{i+1} w_{i+1},\ h_{i+1} x_{i+1}$, and $g_i w_i$ costs, respectively, $\rho'_i \rho'_{i+1},\ \rho'_i n_{i+1}$, and $n_i \rho'_i$ flops. The total complexity of Algorithm 4.1 is expressed as follows:

$$c = \sum_{k=1}^{N} \left[n_k \rho'_k + n_{k+1} \rho'_k + \rho'_k \rho'_{k+1} + \zeta(n_k) \right]. \tag{4.1}$$

Here $\zeta(n)$ is a complexity of solution of $n \times n$ linear system by the standard Gauss method. If the sizes of a matrix $m_k,\ n_k$ and the orders of its generators $r'_k,\ r''_k$ are bounded by the numbers m and r, respectively, we obtain the estimate

$$c \leqslant N(2mr + r^2 + \zeta(m)).$$

In this case, similarly to the multiplication by vector, the solution of a triangular system costs $\mathrm{O}(N)$ operations in contrast to $\mathrm{O}(N^2)$ for a matrix of general form.

5. Factorization of triangular matrices

In this section we obtain some factorization relations for triangular matrices using their generators. These results turn out to be useful for derivation of the main algorithms of this paper. Similar relations were used earlier (see [8,9]) for unitary Hessenberg matrices.

Lemma 5.1. *Let $R = \{R_{ij}\}_{i,j=1}^{N}$ be a block lower triangular matrix with entries of sizes $m_i \times n_j$ and lower generators p_i $(i = 2, \dots, N)$, q_j $(j = 1, \dots, N-1)$, a_k $(k = 2, \dots, N-1)$ and diagonal entries d_k $(k = 1, \dots, N)$. By generators and diagonal entries define matrices*

$$\begin{aligned} R_1 &= \begin{bmatrix} d_1 \\ q_1 \end{bmatrix}, \\ R_k &= \begin{bmatrix} p_k & d_k \\ a_k & q_k \end{bmatrix}, \quad k = 2, \dots, N-1, \\ R_N &= \begin{bmatrix} p_N & d_N \end{bmatrix}, \end{aligned} \tag{5.1}$$

and next set

$$\begin{aligned} \tilde{R}_1 &= \mathrm{diag}\{R_1, I_{\gamma_1}\}, \\ \tilde{R}_k &= \mathrm{diag}\{I_{\eta_k}, R_k, I_{\gamma_k}\}, \quad k = 2, \dots, N-1, \\ \tilde{R}_N &= \mathrm{diag}\{I_{\eta_N}, R_N\}, \end{aligned} \tag{5.2}$$

where $\eta_k = \sum_{i=1}^{k-1} m_i,\ \gamma_k = \sum_{i=k+1}^{N} n_i$.

Then the equality

$$R = \tilde{R}_N \cdot \tilde{R}_{N-1} \cdots \tilde{R}_1 \tag{5.3}$$

holds.

Proof. Let us prove by induction the validity of the relations

$$\tilde{R}_k \cdots \tilde{R}_1 = \begin{pmatrix} R(1:k,1:k) & 0 \\ Q_k & 0 \\ 0 & I_{\gamma_k} \end{pmatrix}, \quad k=1,\ldots,N-1, \tag{5.4}$$

where matrices Q_k are given by (2.5).

For $k=1$, relation (5.4) is obvious. Suppose (5.4) holds for k with $1 \leqslant k \leqslant N-2$. Then

$$\begin{aligned} &\tilde{R}_{k+1}\tilde{R}_k \cdots \tilde{R}_1 \\ &\quad = \begin{pmatrix} I_{\eta_{k+1}} & 0 & 0 & 0 \\ 0 & p_{k+1} & d_{k+1} & 0 \\ 0 & a_{k+1} & q_{k+1} & 0 \\ 0 & 0 & 0 & I_{\gamma_{k+1}} \end{pmatrix} \begin{pmatrix} R(1:k,1:k) & 0 & 0 \\ Q_k & 0 & 0 \\ 0 & I_{n_{k+1}} & 0 \\ 0 & 0 & I_{\gamma_{k+1}} \end{pmatrix} \\ &\quad = \begin{pmatrix} R(1:k,1:k) & 0 & 0 \\ p_{k+1}Q_k & d_{k+1} & 0 \\ a_{k+1}Q_k & q_{k+1} & 0 \\ 0 & 0 & I_{\gamma_{k+1}} \end{pmatrix}. \end{aligned}$$

Using relations (2.7) and

$$\begin{pmatrix} R(1:k,1:k) & 0 \\ p_{k+1}Q_k & d_{k+1} \end{pmatrix} = R(1:k+1,1:k+1)$$

we conclude that

$$\tilde{R}_{k+1}\tilde{R}_k \cdots \tilde{R}_1 = \begin{pmatrix} R(1:k+1,1:k+1) & 0 \\ Q_{k+1} & \\ 0 & I_{\gamma_{k+1}} \end{pmatrix}.$$

Relation (5.4) with $k=N-1$ yields

$$\begin{aligned} \tilde{R}_N \cdots \tilde{R}_1 &= \begin{pmatrix} I_{\eta_N} & 0 & 0 \\ 0 & p_N & d_N \end{pmatrix} \begin{pmatrix} R(1:N-1,1:N-1) & 0 \\ Q_{N-1} & 0 \\ 0 & I_{n_N} \end{pmatrix} \\ &= \begin{pmatrix} R(1:N-1,1:N-1) & 0 \\ p_N Q_{N-1} & 0 \end{pmatrix} = R(1:N,1:N) = R. \qquad \square \end{aligned}$$

Lemma 5.2. *Let R be a block upper triangular matrix with entries of sizes $m_i \times n_j$, upper generators g_i $(i=1,\ldots,N-1)$, h_j $(j=2,\ldots,N)$, b_k $(k=2,\ldots,N-1)$ and diagonal entries d_k $(k=1,\ldots,N)$. By generators and diagonal entries define matrices*

$$
\begin{aligned}
&R_1 = \begin{bmatrix} d_1 & g_1 \end{bmatrix},\\
&R_k = \begin{bmatrix} h_k & b_k \\ d_k & g_k \end{bmatrix}, \quad k = 2, \dots, N-1,\\
&R_N = \begin{bmatrix} h_N \\ d_N \end{bmatrix},
\end{aligned}
\tag{5.5}
$$

and next set

$$
\begin{aligned}
&\tilde{R}_1 = \operatorname{diag}\{R_1, I_{\phi_1}\},\\
&\tilde{R}_k = \operatorname{diag}\{I_{\chi_k}, R_k, I_{\phi_k}\}, \quad k = 2, \dots, N-1,\\
&\tilde{R}_N = \operatorname{diag}\{I_{\chi_N}, R_N\},
\end{aligned}
\tag{5.6}
$$

where $\chi_k = \sum_{i=1}^{k-1} n_i,\ \phi_k = \sum_{i=k+1}^{N} m_i$.

Then the equality

$$R = \tilde{R}_1 \cdot \tilde{R}_2 \cdots \tilde{R}_N \tag{5.7}$$

holds.

Lemma 5.2 is obtained from Lemma 5.1 by passing to adjoint matrices.

Let us notice that the adjoint matrix R^* is a block lower triangular matrix with entries of sizes $n_i \times m_j$ and lower generators $h_i^*\ (i = 2, \dots, N),\ g_j^*\ (j = 1, \dots, N-1),\ b_k^*\ (k = 2, \dots, N-1)$ and diagonal entries $d_k^*\ (k = 1, \dots, N)$.

If $m_1 = 1,\ n_1 = 0,\ m_k = n_k = 1\ (k = 2, \dots, N-1),\ m_N = 0,\ n_N = 1,$ the orders of generators of the matrix R equal one and all the matrices R_k in (5.5) are unitary, then R is a unitary Hessenberg matrix and factorization (5.6) is similar to the one used in [8].

6. Description of the method

6.1. General description

Let R be a block invertible matrix. The method suggested by Dewilde and van der Veen in [1] consists of construction of the factorization of the form

$$R = VUS, \tag{6.1}$$

where $V,\ U$ are block unitary matrices, V is block lower triangular, U is block upper triangular, and S is a block upper triangular matrix with square invertible blocks on the main diagonal. The matrices $V,\ U,\ S$ are given by their generators which are computed via generators of the original matrix R. If the generators of the matrices $V,\ U,\ S$ have just been computed, then the solution of the system of linear algebraic equations $Rx = y$ may be determined by $x = S^{-1}U^*V^*y$ using Algorithms 3.1 and 4.1.

On the first stage we compute the factorization $R = VT$, where V is a block lower triangular unitary matrix, and T is a block upper triangular matrix. Following the terminology of [1] we call this stage inner coprime factorization. The matrix T obtained in the first stage has in general rectangular blocks on the main diagonal. In order to obtain matrices which are convenient for inversion we compute for the matrix T the factorization $T = US$, where U is a block upper triangular unitary matrix, and S is a block upper triangular matrix with square invertible blocks on the main diagonal. This stage also following the terminology of [1] we call inner–outer factorization. Below we present the description of both stages with the detailed justification. The proofs are based on Lemmas 5.1 and 5.2 on factorization of triangular matrices and differ completely from the proofs given in [1].

Notice that in [1] instead of (6.1) a more general factorization $R = VUSW$ with an additional block triangular unitary factor W was used. In our case of finite invertible matrices the factor W does not appear and thus the amount of computations is reduced.

6.2. Inner coprime factorization

Let R be a block matrix with given generators. We present here an algorithm for computing generators and diagonal entries of unitary block lower triangular matrix V and block upper triangular matrix T such that $R = VT$. This algorithm is a generalization of an algorithm suggested in [1, p. 131,170] with some additional assumptions which are equivalent to the conditions

$$\operatorname{rank} p_N = r'_{N-1}, \quad \operatorname{rank} \begin{pmatrix} p_k \\ a_k \end{pmatrix} = r'_{k-1}, \; k = N-1, \ldots, 2. \tag{6.2}$$

Theorem 6.1. *Let $R = \{R_{ij}\}_{i,j=1}^N$ be a block matrix with entries of sizes $m_i \times n_j$, lower generators p_i $(i=2,\ldots,N)$, q_j $(j=1,\ldots,N-1)$, a_k $(k=2,\ldots,N-1)$ of orders r'_k $(k=1,\ldots,N-1)$, upper generators g_i $(i=1,\ldots,N-1)$, h_j $(j=2,\ldots,N)$, b_k $(k=2,\ldots,N-1)$ of orders r''_k $(k=1,\ldots,N-1)$ and diagonal entries d_k $(k=1,\ldots,N)$.*

The matrix R admits the factorization

$$R = V \cdot T, \tag{6.3}$$

where V is a block lower triangular unitary matrix with block entries of sizes $m_i \times \nu_j$ $(i,j=1,\ldots,N)$, lower generators $(p_V)_i$ $(i=2,\ldots,N)$, $(q_V)_j$ $(j=1,\ldots,N-1)$, $(a_V)_k$ $(k=2,\ldots,N-1)$ of orders ρ_k $(k=1,\ldots,N-1)$ and diagonal entries $(d_V)_k$ $(k=1,\ldots,N)$ and T is a block upper triangular matrix with block entries of sizes $\nu_i \times n_j$ $(i,j=1,\ldots,N)$, upper generators $(g_T)_i$ $(i=1,\ldots,N-1)$, $(h_T)_j$ $(j=2,\ldots,N)$, $(b_T)_k$ $(k=2,\ldots,N-1)$ of orders $\rho'_k = r''_k + \rho_k$ $(k=1,\ldots,N-1)$ and diagonal entries $(d_T)_k$ $(k=1,\ldots,N)$.

The sizes ν_k, the orders of generators ρ_k, generators and diagonal entries of the matrices V, T are determined using the following algorithm.

1. *Compute $\rho_{N-1} = \min\{m_N,\ r'_{N-1}\}$, $\nu_N = m_N - \rho_{N-1}$. Compute the QR factorization*

$$p_N = V_N \begin{pmatrix} X_N \\ 0 \end{pmatrix}, \tag{6.4}$$

where V_N is a unitary matrix of sizes $m_N \times m_N$ and X_N is a matrix of sizes $\rho_{N-1} \times r'_{N-1}$. Determine matrices $(p_V)_N$ and $(d_V)_N$ of sizes $m_N \times \rho_{N-1}$, and $m_N \times \nu_N$ from the partition

$$V_N = \begin{bmatrix} (p_V)_N & (d_V)_N \end{bmatrix}. \tag{6.5}$$

Compute

$$h'_N = (p_V)^*_N d_N, \quad (h_T)_N = \begin{bmatrix} h_N \\ h'_N \end{bmatrix}, \quad (d_T)_N = (d_V)^*_N d_N. \tag{6.6}$$

2. *For $k = N-1, \ldots, 2$ we perform the following. Compute $\rho_{k-1} = \min\{m_k + \rho_k,\ r'_{k-1}\},\ \nu_k = m_k + \rho_k - \rho_{k-1}$. Compute the QR factorization*

$$\begin{bmatrix} p_k \\ X_{k+1}a_k \end{bmatrix} = V_k \begin{pmatrix} X_k \\ 0 \end{pmatrix}, \tag{6.7}$$

where V_k is a unitary matrix of sizes $(m_k + \rho_k) \times (m_k + \rho_k)$, and X_k is a matrix of sizes $\rho_{k-1} \times r'_{k-1}$. Determine matrices $(p_V)_k,\ (a_V)_k,\ (d_V)_k$, and $(q_V)_k$ of sizes $m_k \times \rho_{k-1},\ \rho_k \times \rho_{k-1},\ m_k \times \nu_k$ and $\rho_k \times \nu_k$ from the partition

$$V_k = \begin{bmatrix} (p_V)_k & (d_V)_k \\ (a_V)_k & (q_V)_k \end{bmatrix}. \tag{6.8}$$

Compute

$$\begin{aligned} h'_k &= (p_V)^*_k d_k + (a_V)^*_k X_{k+1} q_k, \quad (h_T)_k = \begin{bmatrix} h_k \\ h'_k \end{bmatrix}, \\ (b_T)_k &= \begin{pmatrix} b_k & 0 \\ (p^*_V)_k g_k & (a_V)^*_k \end{pmatrix}, \\ (g_T)_k &= \begin{bmatrix} (d_V)^*_k g_k & (q_V)^*_k \end{bmatrix}, \\ (d_T)_k &= (d_V)^*_k d_k + (q_V)^*_k X_{k+1} q_k. \end{aligned} \tag{6.9}$$

3. *Compute $\nu_1 = m_1 + \rho_1$. Choose a unitary matrix V_1 of sizes $\nu_1 \times \nu_1$. Define matrices $(d_V)_1$, and $(q_V)_1$ of sizes $m_1 \times \rho_1$, and $\rho_1 \times \nu_1$ from the partition*

$$V_1 = \begin{bmatrix} (d_V)_1 \\ (q_V)_1 \end{bmatrix}. \tag{6.10}$$

Compute

$$(d_T)_1 = (d_V)^*_1 d_1 + (q_V)^*_1 X_2 q_1, \quad (g_T)_1 = \begin{bmatrix} (d_V)^*_1 g_1 & (q_V)^*_1 \end{bmatrix}. \tag{6.11}$$

Proof. We consider a lower triangular matrix $V = \{V_{ij}\}_{i,j=1}^N$ and an upper triangular matrix $T = \{T_{ij}\}_{i,j=1}^N$ with generators given by the algorithm, i.e.,

$$V_{ij} = \begin{cases} (p_V)_i (a_V)^{\times}_{ij} (q_V)_j, & 1 \leqslant j < i \leqslant N, \\ (d_V)_i, & 1 \leqslant i = j \leqslant N, \\ 0, & 1 \leqslant i < j \leqslant N, \end{cases}$$

$$T_{ij} = \begin{cases} 0, & 1 \leqslant j < i \leqslant N, \\ (d_T)_i, & 1 \leqslant i = j \leqslant N, \\ (g_T)_i (b_T)^{\times}_{ij} (h_T)_j, & 1 \leqslant i < j \leqslant N. \end{cases}$$

Let us show that for such defined matrices $V,\ T$, relation (6.3) holds. Let the matrices V_k $(k = 1, \ldots, N)$ be given by relations (6.5), (6.8) and (6.10). Set

$$\begin{aligned} \tilde{V}_1 &= \operatorname{diag}\{V_1, I_{\phi_1}\}, \\ \tilde{V}_k &= \operatorname{diag}\{I_{\eta_k}, V_k, I_{\phi_k}\}, \quad k = 2, \ldots, N-1, \\ \tilde{V}_N &= \operatorname{diag}\{I_{\eta_N}, V_N\}, \end{aligned}$$

where

$$\eta_k = \sum_{i=1}^{k-1} m_i, \quad \phi_k = \sum_{i=k+1}^{N} \nu_i.$$

From Lemma 5.1 it follows that

$$V = \tilde{V}_N \tilde{V}_{N-1} \cdots \tilde{V}_1. \tag{6.12}$$

Since all the matrices V_k are unitary, all the matrices $\tilde{V}_k$ are also unitary and hence the matrix V is unitary as a product of unitary matrices.

Let us prove by induction that

$$\tilde{V}_k^* \cdots \tilde{V}_N^* R = \begin{pmatrix} R(1:k-1,:) \\ \tilde{X}_k \\ T(k:N,:) \end{pmatrix}, \quad k = N, \ldots, 2, \tag{6.13}$$

where

$$\begin{aligned} \tilde{X}_N &= \begin{pmatrix} X_N Q_{N-1} & h'_N \end{pmatrix}, \\ \tilde{X}_k &= \begin{pmatrix} X_k Q_{k-1} & h'_k & \Lambda_k \end{pmatrix}, \quad k = N-1, \ldots, 2, \end{aligned}$$

the elements h'_k and X_k are given in (6.6), (6.9) and (6.4), (6.7),

$$\Lambda_k = \begin{pmatrix} (p_V^*)_k g_k & (a_V)_k^* \end{pmatrix} (H_T)_{k+1}, \tag{6.14}$$

and matrices Q_k and $(H_T)_k$ are defined by (2.5) and (2.6).

For $k = N$ we represent the matrix R in the form

$$R = \begin{pmatrix} R(1:N-1,:) \\ R(N,:) \end{pmatrix}.$$

The last relation from (2.4) yields $R(N,:) = \begin{pmatrix} p_N Q_{N-1} & d_N \end{pmatrix}$. From (6.4) we obtain

$$V_N^* p_N Q_{N-1} = \begin{pmatrix} X_N Q_{N-1} \\ 0 \end{pmatrix}.$$

Next, from relations (6.6) it follows that

$$V_N^* d_N = \begin{pmatrix} (p_V)_N^* \\ (d_V)_N^* \end{pmatrix} d_N = \begin{pmatrix} h'_N \\ (d_T)_N \end{pmatrix}.$$

Thus we obtain

$$V_N^* R(N,:) = \begin{pmatrix} X_N Q_{N-1} & h'_N \\ 0 & (d_T)_N \end{pmatrix} = \begin{pmatrix} \tilde{X}_N \\ T(N,:) \end{pmatrix}.$$

Hence it follows that

$$\begin{aligned}\tilde{V}_N^* R &= \begin{pmatrix} I_{\eta_N} & 0 \\ 0 & V_N^* \end{pmatrix} \begin{pmatrix} R(1:N-1,:) \\ R(N,:) \end{pmatrix} \\ &= \begin{pmatrix} R(1:N-1,:) \\ V_N^* R(N,:) \end{pmatrix} \\ &= \begin{pmatrix} R(1:N-1,:) \\ \tilde{X}_N \\ T(N,:) \end{pmatrix}.\end{aligned}$$

Suppose (6.13) holds for k with $N \geqslant k \geqslant 3$. Then

$$\begin{aligned}\tilde{V}_{k-1}^* \tilde{V}_k^* \cdots \tilde{V}_N^* R &= \tilde{V}_{k-1}^* \begin{pmatrix} R(1:k-1,:) \\ \tilde{X}_k \\ T(k:N,:) \end{pmatrix} \\ &= \begin{pmatrix} I\eta_{k-1} & 0 & 0 \\ 0 & V_{k-1}^* & 0 \\ 0 & 0 & I_{\phi_{k-1}} \end{pmatrix} \begin{pmatrix} R(1:k-2,:) \\ R(k-1,:) \\ \tilde{X}_k \\ T(k:N,:) \end{pmatrix} \\ &= \begin{pmatrix} R(1:k-2,:) \\ V_{k-1}^* \begin{pmatrix} R(k-1,:) \\ \tilde{X}_k \end{pmatrix} \\ T(k:N,:) \end{pmatrix}.\end{aligned}$$

From the definition of matrices $\tilde{X}_k$ and relations (2.7) we obtain

$$\tilde{X}_k = \begin{pmatrix} X_k a_{k-1} Q_{k-2} & X_k q_{k-1} & h'_k & \Lambda_k \end{pmatrix}.$$

Relations (2.4) and (2.8) yield

$$R(k-1,:) = \begin{pmatrix} p_{k-1} Q_{k-2} & d_{k-1} & g_{k-1} h_k & g_{k-1} b_k H_{k+1} \end{pmatrix}.$$

Thus we have

$$\begin{pmatrix} R(k-1,:) \\ \tilde{X}_k \end{pmatrix} = \begin{pmatrix} p_{k-1} Q_{k-2} & d_{k-1} & g_{k-1} h_k & g_{k-1} b_k H_{k+1} \\ X_k a_{k-1} Q_{k-2} & X_k q_{k-1} & h'_k & \Lambda_k \end{pmatrix}. \tag{6.15}$$

By virtue of (6.7) we obtain

$$V_{k-1}^{*}\begin{pmatrix} p_{k-1}Q_{k-2} \\ X_k a_{k-1}Q_{k-2}\end{pmatrix} = \begin{pmatrix} X_{k-1}Q_{k-2} \\ 0\end{pmatrix}. \tag{6.16}$$

Relations (6.8) and (6.9) yield

$$V_{k-1}^{*}\begin{pmatrix} d_{k-1} \\ X_k q_{k-1}\end{pmatrix} = \begin{bmatrix} (p_V)_{k-1}^{*} & (a_V)_{k-1}^{*} \\ (d_V)_{k-1}^{*} & (q_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} d_{k-1} \\ X_k q_{k-1}\end{pmatrix} = \begin{pmatrix} h'_{k-1} \\ (d_T)_{k-1}\end{pmatrix}. \tag{6.17}$$

Using (6.9) we obtain

$$\begin{aligned}\begin{bmatrix}(d_V)_{k-1}^{*} & (q_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} g_{k-1}h_k \\ h'_k\end{pmatrix} &= \begin{bmatrix}(d_V)_{k-1}^{*}g_{k-1} & (q_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} h_k \\ h'_k\end{pmatrix} \\ &= (g_T)_{k-1}(h_T)_k.\end{aligned} \tag{6.18}$$

Next we have

$$\begin{bmatrix}(d_V)_{k-1}^{*} & (q_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} g_{k-1}b_k H_{k+1} \\ \Lambda_k\end{pmatrix} = (g_T)_{k-1}\begin{pmatrix} b_k H_{k+1} \\ \Lambda_k\end{pmatrix}. \tag{6.19}$$

Using (6.14), the relation

$$(H_T)_{k+1} = \begin{pmatrix} H_{k+1} \\ * \end{pmatrix}$$

and (6.9), we obtain

$$\begin{pmatrix} b_k H_{k+1} \\ \Lambda_k\end{pmatrix} = \begin{pmatrix} b_k H_{k+1} \\ \left((p_V^{*})_k g_k \quad (a_V)_k^{*}\right)(H_T)_{k+1}\end{pmatrix} = (b_T)_k (H_T)_{k+1}. \tag{6.20}$$

Thus from (6.18) and (6.19) we obtain

$$\begin{aligned}&\begin{bmatrix}(d_V)_{k-1}^{*} & (q_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} g_{k-1}h_k & g_{k-1}b_k H_{k+1} \\ h'_k & \Lambda_k\end{pmatrix} \\ &\quad = (g_T)_{k-1}\left((h_T)_k \quad (b_T)_k (H_T)_{k+1}\right),\end{aligned}$$

and relation (2.8) for the matrix $(H_T)_k$ yields

$$\begin{bmatrix}(d_V)_{k-1}^{*} & (q_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} g_{k-1}h_k & g_{k-1}b_k H_{k+1} \\ h'_k & \Lambda_k\end{pmatrix} = (g_T)_{k-1}(H_T)_k. \tag{6.21}$$

Next, using relations (6.9), (6.20), (6.14) and (2.8) we obtain

$$\begin{aligned}&\begin{bmatrix}(p_V)_{k-1}^{*} & (a_V)_{k-1}^{*}\end{bmatrix}\begin{pmatrix} g_{k-1}h_k & g_{k-1}b_k H_{k+1} \\ h'_k & \Lambda_k\end{pmatrix} \\ &\quad = \begin{bmatrix}(p_V)_{k-1}^{*}g_{k-1} & (a_V)_{k-1}^{*}\end{bmatrix}(H_T)_k = \Lambda_{k-1}.\end{aligned} \tag{6.22}$$

Thus from relations (6.15)–(6.17), (6.21), (6.22) and relation (2.4) for the upper triangular matrix T we conclude that

$$V_{k-1}^*\begin{pmatrix} R(k-1,:) \\ \tilde{X}_k \end{pmatrix} = \begin{pmatrix} X_{k-1}Q_{k-2} & h'_{k-1} & \Lambda_{k-1} \\ 0 & (d_T)_{k-1} & (g_T)_{k-1}(H_T)_k \end{pmatrix}$$
$$= \begin{pmatrix} \tilde{X}_{k-1} \\ T(k-1,:) \end{pmatrix}$$

which completes the proof of (6.13).

Using (6.13) with $k = 2$ we obtain

$$V^*R = \tilde{V}_1^*\tilde{V}_2^*\cdots\tilde{V}_N^*R$$
$$= \begin{pmatrix} V_1^* & 0 \\ 0 & I_{\phi_1} \end{pmatrix}\begin{pmatrix} R(1,:) \\ \tilde{X}_2 \\ T(2:N,:) \end{pmatrix} = \begin{pmatrix} V_1^*\begin{pmatrix} R(1,:) \\ \tilde{X}_2 \end{pmatrix} \\ T(2:N,:) \end{pmatrix}.$$

Next, the first of relations (2.4) and (2.8) yield

$$\begin{pmatrix} R(1,:) \\ \tilde{X}_2 \end{pmatrix} = \begin{pmatrix} d_1 & g_1h_2 & g_1H_2 \\ X_2q_1 & h'_2 & \Lambda_2 \end{pmatrix}.$$

By virtue of (6.10), (6.11) and (6.21) and the first of relations (2.4) we obtain

$$V_1^*\begin{pmatrix} R(1,:) \\ \tilde{X}_2 \end{pmatrix} = \begin{bmatrix} (d_V)_1^* & (q_V)_1^* \end{bmatrix}\begin{pmatrix} d_1 & g_1h_2 & g_1H_2 \\ X_2q_1 & h'_2 & \Lambda_2 \end{pmatrix}$$
$$= \begin{pmatrix} (d_T)_1 & (g_T)_1(H_T)_2 \end{pmatrix} = T(1,:).$$

Thus we obtain

$$V^*R = \begin{pmatrix} T(1,:) \\ T(2:N,:) \end{pmatrix} = T.$$

Hence (6.3) follows. □

If relations (6.2) hold, then the algorithm given in Theorem 6.1 is simplified since in this case $\rho_k = r'_k,\ 1 \leqslant k \leqslant N-1$. In order to check the last relations, notice that from (6.2) it follows that $m_N \leqslant r'_{N-1},\ r'_{k-1} \leqslant m_k + r'_k$ and hence $\rho_{N-1} = \min\{m_N,\ r'_{N-1}\} = r'_{N-1}$ and next from $\rho_k = r'_k$ it follows that $\rho_{k-1} = \min\{m_k + \rho_k,\ r'_{k-1}\} = r'_{k-1}$. Generators of a matrix satisfying (6.2) may be obtained by the original ones using an algorithm presented in [1, p. 101]. However computational framework of this algorithm is not clear to the authors.

6.3. Inner–outer factorization

In this section we consider block invertible upper triangular matrices with given upper generators. If a block triangular matrix is invertible, the following conditions on the sizes of its entries are valid.

Lemma 6.2. *Let $T = \{T_{ij}\}_{i,j=1}^{N}$ be a block upper triangular matrix with entries of sizes $\nu_i \times n_j$.*

Then

$$\sum_{i=1}^{k}(\nu_i - n_i) \geqslant 0, \quad k = 1, \ldots, N-1,$$

$$\sum_{i=1}^{N}(\nu_i - n_i) = 0.$$

Proof. Let us consider the submatrix $T(:, 1:k)$ which is composed of the first k block columns of the matrix T. We have

$$T(:, 1:k) = \begin{pmatrix} T_k \\ 0 \end{pmatrix},$$

where T_k is a matrix of sizes

$$\left(\sum_{i=1}^{k}\nu_i\right) \times \left(\sum_{i=1}^{k}n_i\right).$$

From the invertibility of the matrix T it follows that rank $T_k = \sum_{i=1}^{k} n_i$ and thus

$$\sum_{i=1}^{k} n_i \leqslant \sum_{i=1}^{k}\nu_i, \quad 1 \leqslant k \leqslant N-1.$$

From the invertibility of T it follows that T is a square matrix. Hence

$$\sum_{i=1}^{N}\nu_i = \sum_{i=1}^{N} n_i. \qquad \square$$

In order to transform a matrix to a form which is convenient for the inversion, we use a specification of an algorithm suggested in [1, p. 171].

Theorem 6.3. *Let $T = \{T_{ij}\}_{i,j=1}^{N}$ be a block upper triangular invertible matrix with entries of sizes $\nu_i \times n_j$, upper generators $(g_T)_i$ $(i = 1, \ldots, N-1)$, $(h_T)_j$ $(j = 2, \ldots, N)$, $(b_T)_k$ $(k = 2, \ldots, N-1)$ of orders ρ'_k $(k = 1, \ldots, N-1)$ and diagonal entries $(d_T)_k$ $(k = 1, \ldots, N)$.*

The matrix T admits the factorization

$$T = US, \tag{6.23}$$

where U is a block upper triangular unitary matrix with block entries of sizes $\nu_i \times n_j$ $(i, j = 1, \ldots, N)$, upper generators $(g_{\mathrm{U}})_i$ $(i = 1, \ldots, N-1)$, $(h_{\mathrm{U}})_j$ $(j = 2, \ldots, N)$, $(b_{\mathrm{U}})_k$ $(k=2, \ldots, N-1)$ of orders $s_k = \sum_{i=1}^{k}(\nu_i - n_i)$ $(k=1, \ldots, N-1)$ and diagonal entries $(d_{\mathrm{U}})_k$ $(k = 1, \ldots, N)$ and S is a block upper triangular invertible matrix with block entries of sizes $n_i \times n_j$ $(i, j = 1, \ldots, N)$, upper generators

$(g_S)_i$ $(i=1,\dots,N-1)$, $(h_S)_j$ $(j=2,\dots,N)$, $(b_S)_k$ $(k=2,\dots,N-1)$ of orders ρ'_k $(k=1,\dots,N-1)$ and invertible diagonal entries $(d_S)_k$ $(k=1,\dots,N)$.

Generators and diagonal entries of the matrices U, S are determined using the following algorithm.

1. *Compute $s_1=\nu_1-n_1$. Compute the QR factorization*
$$\begin{bmatrix}(d_T)_1 & (g_T)_1\end{bmatrix}=U_1\begin{bmatrix}(d_S)_1 & (g_S)_1\\ 0 & Y_1\end{bmatrix}, \tag{6.24}$$
where U_1 is a unitary matrix of sizes $\nu_1\times\nu_1$, $(d_S)_1$ is an upper triangular matrix of sizes $n_1\times n_1$, and Y_1 is a matrix of sizes $s_1\times\rho'_1$. Determine matrices $(d_{\mathrm{U}})_1$, and $(g_{\mathrm{U}})_1$ of sizes $\nu_1\times n_1$, and $\nu_1\times\rho'_1$ from the partition
$$U_1=\begin{bmatrix}(d_{\mathrm{U}})_1 & (g_{\mathrm{U}})_1\end{bmatrix}. \tag{6.25}$$
2. *For $k=2,\dots,N-1$ perform the following. Compute $s_k=s_{k-1}+\nu_k-n_k$. Compute the QR factorization*
$$\begin{bmatrix}Y_{k-1}(h_T)_k & Y_{k-1}(b_T)_k\\ (d_T)_k & (g_T)_k\end{bmatrix}=U_k\begin{bmatrix}(d_S)_k & (g_S)_k\\ 0 & Y_k\end{bmatrix}, \tag{6.26}$$
where U_k is a unitary matrix of sizes $(\nu_k+s_{k-1})\times(\nu_k+s_{k-1})$, $(d_S)_k$ is an upper triangular matrix of sizes $n_k\times n_k$, and Y_k is a matrix of sizes $s_k\times\rho'_k$. Set $(h_S)_k=(h_T)_k$, $(b_S)_k=(b_T)_k$.
Determine matrices $(d_{\mathrm{U}})_k$, and $(g_{\mathrm{U}})_k$, $(h_{\mathrm{U}})_k$, and $(b_{\mathrm{U}})_k$ of sizes $\nu_k\times n_k$, $\nu_k\times s_k$, $s_{k-1}\times n_k$, and $s_{k-1}\times s_k$ from the partition
$$U_k=\begin{bmatrix}(h_{\mathrm{U}})_k & (b_{\mathrm{U}})_k\\ (d_{\mathrm{U}})_k & (g_{\mathrm{U}})_k\end{bmatrix}. \tag{6.27}$$
3. *Compute the QR factorization*
$$\begin{bmatrix}Y_{N-1}(h_T)_N\\ (d_T)_N\end{bmatrix}=U_N(d_S)_N, \tag{6.28}$$
where U_N is a unitary matrix of sizes $(\nu_N+s_{N-1})\times(\nu_N+s_{N-1})$, and $(d_S)_N$ is an upper triangular matrix of sizes $n_N\times n_N$. Set $(h_S)_N=(h_T)_N$.
Determine matrices $(d_{\mathrm{U}})_N$ and $(h_{\mathrm{U}})_N$ of sizes $\nu_N\times n_N$ and $s_{N-1}\times n_N$ from the partition
$$U_N=\begin{bmatrix}(h_{\mathrm{U}})_N\\ (d_{\mathrm{U}})_N\end{bmatrix}. \tag{6.29}$$

Proof. From Lemma 6.2 it follows that all the numbers s_k are nonnegative and $\nu_N+s_{N-1}=n_N$.

We consider upper triangular matrices $U=\{U_{ij}\}_{i,j=1}^N$ and $S=\{S_{ij}\}_{i,j=1}^N$ with generators given by the algorithm, i.e.,

$$U_{ij} = \begin{cases} 0, & 1 \leqslant j < i \leqslant N, \\ (d_{\mathrm{U}})_i, & 1 \leqslant i = j \leqslant N, \\ (g_{\mathrm{U}})_i (b_{\mathrm{U}})^{\times}_{ij} (h_{\mathrm{U}})_j, & 1 \leqslant i < j \leqslant N, \end{cases}$$

and

$$S_{ij} = \begin{cases} 0, & 1 \leqslant j < i \leqslant N, \\ (d_S)_i, & 1 \leqslant i = j \leqslant N, \\ (g_S)_i (b_S)^{\times}_{ij} (h_S)_j, & 1 \leqslant i < j \leqslant N. \end{cases}$$

Let us show that for such defined matrices U and S, relation (6.23) holds. Let the matrices U_k $(k = 1, \dots, N)$ be given by relations (6.25), (6.27) and (6.29). Set

$$\begin{aligned} \tilde{U}_1 &= \operatorname{diag}\{U_1, I_{\phi_1}\}, \\ \tilde{U}_k &= \operatorname{diag}\{I_{\chi_k}, U_k, I_{\phi_k}\}, \quad k = 2, \dots, N-1, \\ \tilde{U}_N &= \operatorname{diag}\{I_{\chi_N}, U_N\}, \end{aligned}$$

where

$$\chi_k = \sum_{i=1}^{k-1} n_i, \quad \phi_k = \sum_{i=k+1}^{N} \nu_i.$$

From Lemma 5.2 it follows that

$$U = \tilde{U}_1 \tilde{U}_2 \cdots \tilde{U}_N. \tag{6.30}$$

Moreover U is unitary as a product of unitary matrices.

Let us prove by induction that

$$\tilde{U}_k^* \cdots \tilde{U}_1^* T = \begin{pmatrix} S(1:k, :) \\ \tilde{Y}_k \\ T(k+1:N, :) \end{pmatrix}, \quad k = 1, \dots, N-1, \tag{6.31}$$

where

$$\tilde{Y}_k = \begin{pmatrix} 0_{s_k \times n_{k+1}} & Y_k (H_T)_{k+1} \end{pmatrix}$$

and matrices $(H_T)_k$ are defined by (2.6).

For $k = 1$ we represent the matrix T in the form

$$T = \begin{pmatrix} T(1, :) \\ T(2:N, :) \end{pmatrix}.$$

The first of relations (2.4) yields $T(1, :) = ((d_T)_1 \;\; (g_T)_1 (H_T)_2)$. From (6.24) we obtain

$$U_1^* T(1, :) = \begin{pmatrix} (d_S)_1 & (g_S)_1 (H_T)_2 \\ 0 & Y_1 (H_T)_2 \end{pmatrix} = \begin{pmatrix} S(1, :) \\ \tilde{Y}_1 \end{pmatrix}.$$

Hence

$$\tilde{U}_1^* T = \begin{pmatrix} U_1 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} T(1,:) \\ T(2,:) \end{pmatrix} = \begin{pmatrix} S(1,:) \\ \tilde{Y}_1 \\ T(2,:) \end{pmatrix}.$$

Suppose (6.31) holds for k with $1 \leqslant k \leqslant N-1$. Then

$$\begin{aligned} \tilde{U}_{k+1}^* \tilde{U}_k^* \cdots \tilde{U}_1^* T &= \tilde{U}_{k+1}^* \begin{pmatrix} S(1:k,:) \\ \tilde{Y}_k \\ T(k+1:N,:) \end{pmatrix} \\ &= \begin{pmatrix} I_{\chi_{k+1}} & 0 & 0 \\ 0 & U_{k+1}^* & 0 \\ 0 & 0 & I_{\phi_{k+1}} \end{pmatrix} \begin{pmatrix} S(1:k,:) \\ \tilde{Y}_k \\ T(k+1,:) \\ T(k+2:N,:) \end{pmatrix} \\ &= \begin{pmatrix} S(1:k,:) \\ U_k^* \begin{pmatrix} \tilde{Y}_k \\ T(k+1,:) \end{pmatrix} \\ T(k+2:N,:) \end{pmatrix}. \end{aligned}$$

Relation (2.8) yields

$$\tilde{Y}_k = \begin{pmatrix} 0_{s_k \times \chi_{k+1}} & Y_k (h_T)_{k+1} & Y_k (b_T)_{k+1} (H_T)_{k+2} \end{pmatrix}.$$

Relations (2.4) for the upper triangular matrix T yield

$$T(k+1,:) = \begin{pmatrix} 0_{\nu_k \times \chi_{k+1}} & (d_T)_{k+1} & (g_T)_{k+1} (H_T)_{k+2} \end{pmatrix}.$$

Thus we have

$$\begin{pmatrix} \tilde{Y}_k \\ T(k+1,:) \end{pmatrix} = \begin{pmatrix} 0_{s_k \times \chi_{k+1}} & Y_k (h_T)_{k+1} & Y_k (b_T)_{k+1} (H_T)_{k+2} \\ 0_{\nu_k \times \chi_{k+1}} & (d_T)_{k+1} & (g_T)_{k+1} (H_T)_{k+2} \end{pmatrix}. \tag{6.32}$$

By virtue of (6.26) we obtain

$$\begin{aligned} U_{k+1}^* \begin{pmatrix} \tilde{Y}_k \\ T(k+1,:) \end{pmatrix} &= \begin{pmatrix} 0_{n_k \times \chi_{k+1}} & (d_S)_k & (g_S)_{k+1} (H_T)_{k+2} \\ 0_{s_{k+1} \times \chi_{k+1}} & 0_{s_{k+1} \times n_k} & Y_{k+1} (H_T)_{k+2} \end{pmatrix} \\ &= \begin{pmatrix} S(k+1,:) \\ \tilde{Y}_{k+1} \end{pmatrix}, \end{aligned}$$

which completes the proof of (6.31).

Using (6.31) with $k = N-1$ we obtain

$$\tilde{U}_N^* \tilde{U}_{N-1}^* \cdots \tilde{U}_1^* T = \begin{pmatrix} I_{\chi_N} & 0 \\ 0 & U_N^* \end{pmatrix} \begin{pmatrix} S(1:N-1,:) \\ \tilde{Y}_k \\ T(N,:) \end{pmatrix}$$

$$= \begin{pmatrix} S(1:N-1,:) \\ U_N^* \begin{pmatrix} \tilde{Y}_{N-1} \\ T(N,:) \end{pmatrix} \end{pmatrix}.$$

Since

$$\begin{pmatrix} \tilde{Y}_{N-1} \\ T(N,:) \end{pmatrix} = \begin{pmatrix} 0_{s_{N-1}\times\chi_N} & Y_{N-1}(h_T)_N \\ 0_{\nu_N\times\chi_N} & (d_T)_N \end{pmatrix}$$

we obtain from (6.29) that

$$U_N^* \begin{pmatrix} \tilde{Y}_{N-1} \\ T(N,:) \end{pmatrix} = \begin{pmatrix} 0_{n_N\times\chi_N} & (d_S)_N \end{pmatrix} = S(N,:).$$

Thus we conclude that $U^*T = S$, i.e., (6.23) holds.

Invertibility of the matrix S and of the matrices $(d_S)_k$ follows from the invertibility of the matrix T. □

The algorithm suggested in [1, p. 171] handles block matrices which are not necessarily invertible. The difference between Algorithm 6.3 and the algorithm from [1] is the following. Instead of (6.26) the factorization

$$\begin{bmatrix} Y_{k-1}(h_T)_k & Y_{k-1}(b_T)_k \\ (d_T)_k & (g_T)_k \end{bmatrix} = U_k \begin{bmatrix} (d_S)_k & (g_S)_k \\ 0 & Y_k \\ 0 & 0 \end{bmatrix} \tag{6.33}$$

with unitary matrix U_k is used. The partitioning of the second factor in (6.33) is such that matrices $(d_S)_k$, and Y_k containing n_k and ρ_k' columns have full row rank. This also defines the row sizes of $(d_S)_k$, and Y_k.

Notice that in the case of invertible matrix T all the matrices $(d_S)_k$ are invertible and from relations (6.31) and (6.32) it follows that the matrix in the left-hand side of (6.33) has full row rank. Hence in this case conditions of full row rank for the matrices $(d_S)_k$, and Y_k are valid automatically and the zero rows in the second factor in (6.33) are absent.

6.4. Solution of linear systems

Let now us consider the system $Rx = y$ of linear algebraic equations with block invertible matrix R with given generators. Using results of Theorems 6.1 and 6.3 and Algorithms 3.1 and 4.1 we obtain the following algorithm.

Algorithm 6.4. Let $R = \{R_{ij}\}_{i,j=1}^N$ be a block matrix with entries of sizes $m_i \times n_j$, lower generators p_i $(i=2,\dots,N)$, q_j $(j=1,\dots,N-1)$, a_k $(k=2,\dots,N-1)$ of orders r_k' $(k=1,\dots,N-1)$, upper generators g_i $(i=1,\dots,N-1)$, h_j $(j=2,\dots,N)$, b_k $(k=2,\dots,N-1)$ of orders r_k'' $(k=1,\dots,N-1)$ and diagonal entries d_k $(k=1,\dots,N)$. Then solution of the system $Rx = y$ is given as follows.

1. Using algorithm from Theorem 6.1 compute generators

$$(p_V)_i \ (i=2,\dots,N), \quad (q_V)_j \ (j=1,\dots,N-1),$$

$$(a_V)_k \ (k=2,\dots,N-1), \quad (g_T)_i \ (i=1,\dots,N-1),$$

$$(h_T)_j \ (j=2,\dots,N), \quad (b_T)_k \ (k=2,\dots,N-1)$$

and diagonal entries $(d_V)_k,\ (d_T)_k\ (k=1,\dots,N)$ of the block lower triangular unitary matrix V and the block upper triangular matrix T such that $R=VT$.
2. Using algorithm from Theorem 6.3 compute generators

$$(g_{\mathrm{U}})_i \ (i=1,\dots,N-1), \quad (h_{\mathrm{U}})_j \ (j=2,\dots,N),$$

$$(b_{\mathrm{U}})_k \ (k=2,\dots,N-1), \quad (g_S)_i \ (i=1,\dots,N-1),$$

$$(h_S)_j \ (j=2,\dots,N), \quad (b_S)_k \ (k=2,\dots,N-1)$$

and diagonal entries $(d_{\mathrm{U}})_k,\ (d_S)_k\ (k=1,\dots,N)$ of the block upper triangular unitary matrix U and the block upper triangular matrix S with invertible diagonal entries such that $T=US$.
3. Compute the product $\tilde{x}=V^*y$ as follows: start with $\tilde{x}_N=(d_V)_N^*y_N,\ w_{N-1}=(p_V)_N^*y_N,\ \tilde{x}_{N-1}=(q_V)_{N-1}^*w_{N-1}+(d_V)_{N-1}^*y_{N-1}$ and for $i=N-2,\dots,1$ compute recursively

$$w_i=(a_V)_{i+1}^*w_{i+1}+(p_V)_{i+1}^*y_{i+1}, \quad \tilde{x}_i=(q_V)_i^*w_i+(d_V)_i^*y_i.$$

4. Compute the product $x'=U^*\tilde{x}$ as follows: start with $x_1'=(d_{\mathrm{U}})_1^*\tilde{x}_1,\ z_2=(g_{\mathrm{U}})_1^*\tilde{x}_1,\ x_2'=(h_{\mathrm{U}})_2^*z_2+(d_{\mathrm{U}})_2^*\tilde{x}_2$ and for $i=3,\dots,N$ compute recursively

$$z_i=(b_{\mathrm{U}})_i^*z_{i-1}+(g_{\mathrm{U}})_{i-1}^*\tilde{x}_{i-1}, \quad x_i'=(d_{\mathrm{U}})_i^*\tilde{x}_i+(h_{\mathrm{U}})_i^*z_i.$$

5. Compute the solution x of the equation $Sx=x'$ as follows: start with $x_N=(d_S)_N^{-1}x_N',\ \ w_{N-1}=(h_S)_Nx_N',\ x_{N-1}=(d_S)_{N-1}^{-1}(x_{N-1}'-(g_S)_{N-1}w_{N-1})$ and for $i=N-2,\dots,1$ compute recursively

$$w_i=(b_S)_{i+1}w_{i+1}+(h_S)_{i+1}x_{i+1}, \quad x_i=(d_S)_i^{-1}(x_i'-(g_S)_iw_i).$$

Here in Stages 3 and 4 we used Algorithm 3.1 for the upper triangular matrix V^* with upper generators $(q_V)_i^*\ (i=1,\dots,N-1),\ (p_V)_j^*\ (j=2,\dots,N),\ (a_V)_k^*\ (k=2,\dots,N-1)$ and diagonal entries $(d_V)_k^*\ (k=1,\dots,N)$ and for the lower triangular matrix U^* with lower generators $(h_{\mathrm{U}})_i^*\ (i=2,\dots,N),\ (g_{\mathrm{U}})_j^*\ (j=1,\dots,N-1),\ (b_{\mathrm{U}})_k^*\ (k=2,\dots,N-1)$ and diagonal entries $(d_{\mathrm{U}})_k^*\ (k=1,\dots,N)$. Computations in Stages 3 and 4 may be performed also based on relation (6.12) for the matrix V and relation (6.30) for the matrix U. In Stage 5 we apply Algorithm 4.1 to the upper triangular matrix S.

6.5. Complexity

We consider here the costs of computations in Algorithm 6.4 presented above. In Stage 1, i.e., in the algorithm from Theorem 6.1, costs are determined by relations (6.7) and (6.9). In (6.7), computing of the product $X_{k+1}a_k$ requires $\rho_k r'_k r'_{k-1}$ flops and the QR factorization costs $\vartheta(m_k+\rho_k, r'_{k-1})$ flops. Here $\vartheta(m,n)$ means complexity of QR factorization for a matrix of sizes $m \times n$. In (6.9), computing of the products $(p_V)^*_k d_k,\ (a_V)^*_k X_{k+1}q_k,\ (p^*_V)_k g_k,\ (d_V)^*_k g_k,\ (d_V)^*_k d_k$, and $(q_V)^*_k X_{k+1}q_k$ costs, respectively, $\rho_{k-1}m_k n_k,\ \rho_{k-1}\rho_k r'_k n_k,\ \rho_{k-1}m_k r''_k,\ \nu_k m_k r''_k, \nu_k m_k n_k$, and $\nu_k\rho_k r'_k n_k$ flops. Thus the total complexity of Stage 1 is

$$c_1=\sum_{k=1}^{N}[\,\vartheta(m_k+\rho_k, r'_{k-1})+\rho_{k-1}m_k n_k+\rho_{k-1}\rho_k r'_k n_k+\rho_{k-1}m_k r''_k \\ +\nu_k m_k r''_k+\nu_k m_k n_k+\nu_k\rho_k r'_k n_k]$$

flops. In Stage 2, i.e., in the algorithm from Theorem 6.3, costs are determined by relation (6.26). Computing of the products $Y_{k-1}(h_T)_k$, and $Y_{k-1}(b_T)_k$ costs $s_{k-1}\rho'_{k-1}n_k$, and $s_{k-1}\rho'_{k-1}\rho'_k$ flops, respectively, computing of the QR factorization costs $\vartheta(s_{k-1}+\nu_k, n_k+\rho'_k)$ flops. Thus the total complexity of Stage 2 is

$$c_2=\sum_{k=1}^{N}[\vartheta(s_{k-1}+\nu_k, n_k+\rho'_k)+s_{k-1}\rho'_{k-1}(n_k+\rho'_k)]$$

flops. In Stage 3, we apply to the upper triangular matrix V^* relation (3.3) with $m_k=\nu_k,\ n_k=m_k,\ r''_k=\rho_k,\ r'_k=0$ and obtain the complexity

$$c_3=\sum_{k=1}^{N}[\nu_k\rho_k+m_{k+1}\rho_k+\rho_k\rho_{k+1}+\nu_k m_k]$$

flops. In Stage 4, we apply to the lower triangular matrix U^* relation (3.3) with $m_k=n_k,\ n_k=\nu_k,\ r''_k=0,\ r'_k=s_k$ and obtain the complexity

$$c_4=\sum_{k=1}^{N}[n_k s_{k-1}+\nu_{k-1}s_{k-1}+s_{k-1}s_{k-2}+n_k\nu_k]$$

flops. And finally complexity of Stage 5 is given by (4.1):

$$c_5=\sum_{k=1}^{N}[n_k\rho'_k+n_{k+1}\rho'_k+\rho'_k\rho'_{k+1}+\zeta(n_k)],$$

where $\tilde{\zeta}(n)$ is a complexity of solution of $n \times n$ linear triangular system by the standard method. The total complexity of Algorithm 6.4 is the sum $c=c_1+c_2+c_3+c_4+c_5$.

Assume that the sizes of blocks $m_k, n_k,$ the orders of generators r'_k, r''_k of the matrix R and the values $\sum_{i=1}^{k}(m_i - n_i)$ are bounded by the numbers $m, r,$ and s_0, respectively, i.e.,

$$m_k, n_k \leqslant m, \quad k = 1, \ldots, N,$$
$$r'_k, r''_k \leqslant r, \quad \sum_{i=1}^{k}(m_i - n_i) \leqslant s_0, \quad k = 1, \ldots, N-1.$$

Then the following estimates are obtained. From the relation $\rho_{k-1} = \min\{m_k + \rho_k,\ r'_{k-1}\}$ it follows that $\rho_k \leqslant r$ and from the equality $\rho'_k = r''_k + \rho_k$ we conclude that $\rho'_k \leqslant 2r$. Next we have

$$\sum_{k=1}^{N} \nu_k = \sum_{k=1}^{N} m_k \leqslant mN$$

and from $\nu_k = m_k + \rho_k - \rho_{k-1}$ we conclude that

$$\begin{aligned} s_k = \sum_{i=1}^{k}(\nu_i - n_i) &= \sum_{i=1}^{k}(m_i + \rho_i - \rho_{i-1} - n_i) \\ &= \rho_k + \sum_{i=1}^{k}(m_i - n_i) \leqslant r + s_0, \end{aligned}$$

$$\nu_k + s_{k-1} = m_k + \rho_k + \sum_{i=1}^{k-1}(m_i - n_i) \leqslant m + r + s_0.$$

Using these relations the complexities c_1, c_2, c_3, c_4, c_5 are estimated as follows:

$$c_1 \leqslant N(\vartheta(m + r, r) + 2rm^2 + r^3m + r^2m + m^3 + r^2m^2),$$

$$c_2 \leqslant N(\vartheta(m + r + s_0, n + 2r) + (2rm + 4r^2)s_0 + 2r^2m + 4r^3),$$

$$c_3 \leqslant N(2mr + r^2 + m^2),$$

$$c_4 \leqslant N(2mr + r^2 + s_0(2mr + 2r + s_0) + m^2),$$

$$c_5 \leqslant N(4mr + 4r^2 + \tilde{\zeta}(m)).$$

Thus the total complexity of Algorithm 6.4 is estimated as follows:

$$\begin{aligned} c \leqslant N(&\vartheta(m + r, r) + \vartheta(r + m + s_0, m + r) + \tilde{\zeta}(m) + 2rm^2 + r^3m + 3r^2m \\ &+ r^2m^2 + m^3 + 4r^3 + 8mr + 6r^2 + 2m^2 + s_0(4mr + 4r^2 + 2r + s_0)). \end{aligned}$$

Thus in this case Algorithm 6.4 has a linear $\mathrm{O}(N)$ complexity.

Assume that sizes of the blocks of the matrix R satisfy $m_k = n_k,\ k = 1, \dots, N$. Then since $s_0 = 0$ we conclude that

$$c \leqslant N(\vartheta(m+r,r) + \vartheta(r+m,m+r) + \tilde{\zeta}(m) + 2rm^2 + r^3m + 3r^2m + r^2m^2 + m^3 + 4r^3 + 8mr + 6r^2 + 2m^2). \tag{6.34}$$

The coefficient in N here is bigger than the coefficient in the corresponding formula in [7]. This fact is confirmed by the measuring of the time in numerical experiments (see Section 8).

7. The case of scalar matrices

We consider here the case of a matrix $R = \{R_{ij}\}_{i,j=1}^N$ with scalar entries, i.e., $m_k = n_k = 1$. Let r'_k $(k = 1, \dots, N-1)$ be orders of lower generators of R. In factorization (6.1) the matrix S is a scalar upper triangular matrix, V, U are unitary matrices. Thus we have here a special form of the QR factorization in which the unitary factor is represented as the product VU. The matrix $V = \{v_{ij}\}_{i,j=1}^N$ with scalar entries v_{ij} may be treated by Theorem 6.1 as a block lower triangular matrix with blocks of sizes $\nu_k = 1 + \rho_k - \rho_{k-1}$ $(k = 1, \dots, N)$, where ρ_k are orders of lower generators of the block matrix V which are defined by the relations $\rho_N = 0,\ \rho_{k-1} = \min\{1 + \rho_k,\ r'_{k-1}\}$ $(k = N-1, \dots, 1)$. The fact that V is a block lower triangular matrix means that $v_{ij} = 0$ for $j > \sum_{k=1}^i \nu_k = i + \rho_i$. Similarly for the unitary matrix $U = \{u_{ij}\}_{i,j=1}^N$ from Theorem 6.3 it follows that $u_{ij} = 0$ for $i > j + \rho_j$. Moreover, from Theorem 6.3 it follows that orders of upper generators of U are equal to $s_k = \sum_{i=1}^k (\nu_i - 1) = \sum_{i=1}^k (\rho_i - \rho_{i-1}) = \rho_k$, i.e., coincide with the orders of generators of the matrix V. If for some r holds $r'_k \leqslant r,\ k = 1, \dots, N-1$, we obtain $\rho_k \leqslant r$ and hence the matrices V and U satisfy the relations $v_{ij} = 0$ for $j > i + r$ and $u_{ij} = 0$ for $i > j + r$.

For convenience we present here for scalar matrices a factorization theorem and an algorithm which are obtained directly from Theorems 6.1 and 6.3 and the notes from the beginning of this section.

Theorem 7.1. *Let $R = \{R_{ij}\}_{i,j=1}^N$ be a scalar matrix with lower generators p_i $(i = 2, \dots, N), q_j$ $(j = 1, \dots, N-1),$ a_k $(k = 2, \dots, N-1)$ of orders r'_k $(k = 1, \dots, N-1),$ upper generators g_i $(i = 1, \dots, N-1),$ h_j $(j = 2, \dots, N),$ b_k $(k = 2, \dots, N-1)$ of orders r''_k $(k = 1, \dots, N-1)$ and diagonal entries d_k $(k=1, \dots, N)$. Let us define the numbers ρ_k via recursive relations $\rho_N = 0,$ and $\rho_{k-1} = \min\{1 + \rho_k,\ r'_{k-1}\},\ k = N, \dots, 2.$*

The matrix R admits the factorization

$$R = VUS,$$

where $V = \{v_{ij}\}_{i,j=1}^N$, *and* $U = \{u_{ij}\}_{i,j=1}^N$ *are unitary matrices satisfying the relations* $v_{ij} = 0$ *for* $j > i + \rho_i$ *and* $u_{ij} = 0$ *for* $i > j + \rho_j$, *and S is an invertible upper triangular matrix. Moreover, the matrices V and U admit the factorizations*

$$V = \tilde{V}_N \tilde{V}_{N-1} \cdots \tilde{V}_1, \quad U = \tilde{U}_1 \tilde{U}_2 \cdots \tilde{U}_N,$$

where

$$\tilde{V}_k = \text{diag}\{I_{k-1}, V_k, I_{N-k-\rho_k}\},$$
$$\tilde{U}_k = \text{diag}\{I_{k-1}, U_k, I_{N-k-\rho_k}\}, \quad k = 1, \ldots, N$$

with unitary matrices V_k *and* U_k *of sizes* $(1 + \rho_k) \times (1 + \rho_k)$; *the matrix S has upper generators of orders* ρ'_k $(k = 1, \ldots, N - 1)$.

The matrices V_k, U_k, *generators* $(g_S)_i$ $(i = 1, \ldots, N-1)$, $(h_S)_j$ $(j = 2, \ldots, N)$, $(b_S)_k$ $(k = 2, \ldots, N-1)$ *and diagonal entries* $(d_S)_k$ $(k = 1, \ldots, N)$ *of the matrix S are determined using the following algorithm.*

1.1. *Set* $V_N = 1$. *If* $r'_{N-1} > 0$, *set*

$$X_N = p_N, (h_S)_N = \begin{bmatrix} h_N \\ d_N \end{bmatrix}, \quad (d_T)_N$$

to be 0×1 *empty matrix*; *if* $r'_{N-1} = 0$, *set* X_N *to be* 0×0 *empty matrix*, $(h_S)_N = h_N$, $(d_T)_N = d_N$.

1.2. *For* $k = N - 1, \ldots, 2$ *perform the following. Compute the QR factorization*

$$\begin{bmatrix} p_k \\ X_{k+1}a_k \end{bmatrix} = V_k \begin{pmatrix} X_k \\ 0 \end{pmatrix},$$

where V_k *is a unitary matrix of sizes* $(1 + \rho_k) \times (1 + \rho_k)$, *and* X_k *is a matrix of sizes* $\rho_{k-1} \times r'_{k-1}$. *Determine matrices* $(p_V)_k$, $(a_V)_k$, $(d_V)_k$, *and* $(q_V)_k$ *of sizes* $1 \times \rho_{k-1}$, $\rho_k \times \rho_{k-1}$, $1 \times (1 + \rho_k - \rho_{k-1})$, *and* $\rho_k \times (1 + \rho_k - \rho_{k-1})$ *from the partition*

$$V_k = \begin{bmatrix} (p_V)_k & (d_V)_k \\ (a_V)_k & (q_V)_k \end{bmatrix}.$$

Compute

$$h'_k = (p_V)_k^* d_k + (a_V)_k^* X_{k+1} q_k, \quad (h_S)_k = \begin{bmatrix} h_k \\ h'_k \end{bmatrix},$$

$$(b_S)_k = \begin{pmatrix} b_k & 0 \\ (p_V^*)_k g_k & (a_V)_k^* \end{pmatrix}, \quad (g_T)_k = \begin{bmatrix} (d_V)_k^* g_k & (q_V)_k^* \end{bmatrix},$$

$$(d_T)_k = (d_V)_k^* d_k + (q_V)_k^* X_{k+1} q_k.$$

1.3. *Set*

$$V_1 = I_{1+\rho_1}, \ (d_T)_1 = \begin{pmatrix} d_1 \\ X_2 q_1 \end{pmatrix}, \ (g_T)_1 = \begin{pmatrix} g_1 & 0 \\ 0 & I_{\rho_1} \end{pmatrix}.$$

Thus we have computed the matrices V_k and generators $(b_S)_k$, $(h_S)_k$ of the matrix S.

2.1. *Compute the QR factorization*

$$\begin{bmatrix}(d_T)_1 & (g_T)_1\end{bmatrix} = U_1 \begin{bmatrix}(d_S)_1 & (g_S)_1 \\ 0 & Y_1\end{bmatrix},$$

where U_1 is a unitary matrix of sizes $(1+\rho_1)\times(1+\rho_1)$, $(d_S)_1$ is a number, $(g_S)_1$ is a row of size ρ_1', and Y_1 is a matrix of sizes $\rho_1 \times \rho_1'$.

2.2. *For $k = 2, \dots, N-1$ perform the following. Compute the QR factorization*

$$\begin{bmatrix}Y_{k-1}(h_S)_k & Y_{k-1}(b_S)_k \\ (d_T)_k & (g_T)_k\end{bmatrix} = U_k \begin{bmatrix}(d_S)_k & (g_S)_k \\ 0 & Y_k\end{bmatrix},$$

where U_k is a unitary matrix of sizes $(1+\rho_k)\times(1+\rho_k)$, $(d_S)_k$ is a number, $(g_S)_k$ is a row of size ρ_k', and Y_k is a matrix of sizes $\rho_k \times \rho_k'$.

2.3. *Set $U_N = 1$. Compute*

$$(d_S)_N = \begin{bmatrix}Y_{N-1}(h_S)_N \\ (d_T)_N\end{bmatrix}.$$

Thus we have computed generators $(g_S)_k$ and diagonal entries $(d_S)_k$ of the matrix S.

For a matrix with scalar entries we obtain the following algorithm for solution of the system of linear algebraic equations.

Algorithm 7.2. Let $R = \{R_{ij}\}_{i,j=1}^N$ be a matrix with scalar entries with lower generators p_i $(i = 2, \dots, N)$, q_j $(j = 1, \dots, N-1)$, a_k $(k = 2, \dots, N-1)$ of orders r_k' $(k = 1, \dots, N-1)$, upper generators g_i $(i = 1, \dots, N-1)$, h_j $(j=2, \dots, N)$, b_k $(k = 2, \dots, N-1)$ of orders r_k'' $(k = 1, \dots, N-1)$ and diagonal entries d_k $(k = 1, \dots, N)$. Then solution of the system $Rx = y$ is given as follows.

1. Set $\rho_N = 0$ and for $k = N-1, \dots, 2$ compute $\rho_{k-1} = \min\{1+\rho_k,\ r_{k-1}'\}$. Using algorithm from Theorem 7.1 compute unitary matrices $V_1, \dots, V_N$, U_1, $\dots, U_N$, generators $(g_S)_i$ $(i = 1, \dots, N-1)$, $(h_S)_j$ $(j = 2, \dots, N)$, $(b_S)_k$ $(k=2, \dots, N-1)$ and diagonal entries $(d_S)_k$ $(k = 1, \dots, N)$ of the upper triangular matrix S such that

$$R = VUS, \quad V = \tilde{V}_N\tilde{V}_{N-1}\cdots\tilde{V}_1, \quad U = \tilde{U}_1\tilde{U}_2\cdots\tilde{U}_N,$$

where

$$\tilde{V}_k = \operatorname{diag}\{I_{k-1}, V_k, I_{N-k-\rho_k}\},$$

$$\tilde{U}_k = \operatorname{diag}\{I_{k-1}, U_k, I_{N-k-\rho_k}\},\ k = 1, \dots, N.$$

2. Determine matrices $(p_V)_k$, $(a_V)_k$, $(d_V)_k$, and $(q_V)_k$ of sizes $1\times\rho_{k-1}$, $\rho_k\times\rho_{k-1}$, $1\times(1+\rho_k-\rho_{k-1})$, and $\rho_k\times(1+\rho_k-\rho_{k-1})$ from the partitions

$$V_N = \begin{bmatrix}(p_V)_N & (d_V)_N\end{bmatrix},$$

$$V_k = \begin{bmatrix} (p_V)_k & (d_V)_k \\ (a_V)_k & (q_V)_k \end{bmatrix}, \quad k = N-1, \ldots, 2,$$

$$V_1 = \begin{bmatrix} (d_V)_1 \\ (q_V)_1 \end{bmatrix}.$$

Compute the product $\tilde{x} = V^* y$ as follows: start with $\tilde{x}_N = (d_V)_N^* y_N,\ w_{N-1} = (p_V)_N^* y_N,\ \tilde{x}_{N-1} = (q_V)_{N-1}^* w_{N-1} + (d_V)_{N-1}^* y_{N-1}$ and for $i = N-2, \ldots, 1$ compute recursively

$$w_i = (a_V)_{i+1}^* w_{i+1} + (p_V)_{i+1}^* y_{i+1}, \quad \tilde{x}_i = (q_V)_i^* w_i + (d_V)_i^* y_i.$$

3. Determine matrices $(d_U)_k,\ (g_U)_k,\ (h_U)_k$, and $(b_U)_k$ of sizes $(1 + \rho_k - \rho_{k-1}) \times 1,\ (1 + \rho_k - \rho_{k-1}) \times \rho_k,\ \rho_{k-1} \times 1$, and $\rho_{k-1} \times \rho_k$ from the partitions

$$U_1 = \begin{bmatrix} (d_U)_1 & (g_U)_1 \end{bmatrix},$$

$$U_k = \begin{bmatrix} (h_U)_k & (b_U)_k \\ (d_U)_k & (g_U)_k \end{bmatrix}, \quad k = 2, \ldots, N-1,$$

$$U_N = \begin{bmatrix} (h_U)_N \\ (d_U)_N \end{bmatrix}.$$

Compute the product $x' = U^* \tilde{x}$ as follows: start with $x'_1 = (d_U)_1^* \tilde{x}_1,\ z_2 = (g_U)_1^* \tilde{x}_1,\ x'_2 = (h_U)_2^* z_2 + (d_U)_2^* \tilde{x}_2$ and for $i = 3, \ldots, N$ compute recursively

$$z_i = (b_U)_i^* z_{i-1} + (g_U)_{i-1}^* \tilde{x}_{i-1}, \quad x'_i = (d_U)_i^* \tilde{x}_i + (h_U)_i^* z_i.$$

4. Compute the solution x of the equation $Sx = x'$ as follows: start with $x_N = (d_S)_N^{-1} x'_N,\ w_{N-1} = (h_S)_N x'_N,\ x_{N-1} = (d_S)_{N-1}^{-1}(x'_{N-1} - (g_S)_{N-1} w_{N-1})$ and for $i = N-2, \ldots, 1$ compute recursively

$$w_i = (b_S)_{i+1} w_{i+1} + (h_S)_{i+1} x_{i+1}, \quad x_i = (d_S)_i^{-1}(x'_i - (g_S)_i w_i).$$

Inequality (6.34) for a scalar matrix yields the following estimate for the complexity of Algorithm 7.2:

$$c \leqslant N(\vartheta(1 + r, r) + \vartheta(r + 1, r + 1) + 5r^3 + 10r^2 + 10r + 4).$$

8. Numerical experiments

As an illustration we present here the results of computer experiments with designed algorithms. We investigate their behavior in floating point arithmetic and compare them with other available algorithms. We solved linear systems $Rx = y$ for random values of input data $p,\ q,\ g,\ h,\ d,\ y,\ a,\ b$. The following algorithms were used:

(1)	GEPP	Gaussian eliminations with partial pivoting.
(2)	DV	Algorithm 6.4 presented above.
(3)	GE1	Algorithm obtained in [7] for block matrices.
(4)	GE2	Algorithm obtained in [6] for matrices with scalar entries and generators of order one.

All algorithms (1)–(4) were implemented in the system MATLAB, version 5.0.0.4064 with unit round-off error 2.2204×10^{-16}. The accuracy of the solutions obtained was estimated by the relations

$$\varepsilon = \frac{\|x - x_{QR}\|}{\|x_{QR}\|}, \quad \varepsilon_y = \frac{\|Rx - y\|}{\|y\|},$$

where x is the solution obtained by the corresponding algorithm, x_{QR} is the solution obtained using QR factorization which we assume to be exact. In each case the condition number $\kappa_2(R)$ of the original matrix was also computed.

In all experiments performed the input data were taken randomly using the random-function. The values of elements of $p, q, g, h, y,$ were chosen in the range 0–10, the values of a, b were in the range 0–1 and the values of the diagonal d were taken from the range 0–100.

The data on time required by the above algorithms are also presented here. The authors have to make a proviso that the test programs were not completely optimized for time performance. At the same time these data can provide an approximation for the real complexities of the compared algorithms.

1. The first series of experiments was performed for block matrices with square blocks of a fixed size and the same orders of generators. We compare here GEPP, DV and GE1 algorithms. The results of computations are presented in Table 1.

The corresponding data of time required are presented in Table 2.

Table 1
$m_k = n_k = 2,\ r'_k = r''_k = 2$

N	$\kappa_2(R)$	GEPP		GEI		DV	
		ε	ε_y	ε	ε_y	ε	ε_y
20	2e + 3	3e−14	7e−15	6e−15	1e−14	7e−15	1e−14
50	1e + 4	3e−14	6e−15	4e−13	1e−12	3e−14	6e−15
100	1e + 6	2e−12	8e−13	2e−12	2e−11	2e−12	7e−13
150	4e + 5	1e−12	3e−13	1e−12	3e−12	1e−12	3e−13
200	5e + 5	2e−12	3e−13	2e−12	3e−12	2e−12	1e−13
500	2e + 6	9e−12	6e−13	1e−11	3e−10	1e−13	2e−13
1000	8e + 8	5e−11	2e−11	1e−10	7e−12	1e−12	7e−12

Table 2
Time (seconds)

N	GE1	DV	MATLB's \
20	0.23	0.87	0.0058
50	0.65	2.48	0.042
100	1.24	5.51	0.29
150	1.77	6.71	1.82
200	2.45	8.99	4.88
500	6.01	22.37	71.23
1000	12.46	44.49	678.35

Table 3

N	$\kappa_2(R)$	GEPP		GEI		DV	
		ε	ε_y	ε	ε_y	ε	ε_y
40	$3e+2$	$2e{-}14$	$2e{-}15$	$1e{-}14$	$3e{-}15$	$2e{-}14$	$2e{-}15$
100	$2e+2$	$9e{-}15$	$1e{-}15$	$8e{-}15$	$1e{-}15$	$9e{-}15$	$1e{-}15$
200	$3e+3$	$9e{-}15$	$8e{-}15$	$4e{-}14$	$1e{-}12$	$1e{-}14$	$8e{-}15$
300	$1e+4$	$2e{-}13$	$1e{-}14$	$2e{-}13$	$1e{-}13$	$2e{-}13$	$7e{-}15$
500	$4e+3$	$8e{-}14$	$4e{-}15$	$8e{-}14$	$2e{-}14$	$8e{-}14$	$4e{-}15$

Table 4
Time (seconds)

N	GE2	DV	MATLB's \
40	0.12	1.51	0.007
100	0.28	3.84	0.042
200	0.58	7.92	0.31
300	0.84	12.08	1.85
500	1.56	22.70	188.34

2. The second series of experiments was performed for matrices with scalar entries and generators of order one. We compare here GEPP, DV and GE2 algorithms. The results of computations are presented in Table 3.

The corresponding data of time required are presented in Table 4.

From these tables it follows that for the examples discussed here accuracy is about the same for all the algorithms. However for large matrices the DV, GE1, GE2 algorithms are much faster. In the examples presented the DV algorithm turned out to be more accurate but the GE1 and GE2 algorithms are faster.

References

[1] P.M. Dewilde, A.J. van der Veen, Time-varying Systems and Computations, Kluwer Academic Publishers, New York, 1998.

[2] P.M. Dewilde, A.J. van der Veen, Inner–outer factorization and the inversion of locally finite systems of equations, Linear Algebra Appl. 313 (2000) 53–100.

[3] Y. Eidelman, I. Gohberg, Inversion formulas and linear complexity algorithm for diagonal plus semiseparable matrices, Comput. Math. Appl. 33 (1997) 69–79.

[4] Y. Eidelman, I. Gohberg, Fast inversion algorithms for diagonal plus semiseparable matrices, Integral Equations Operator Theory 27 (1997) 165–183.

[5] Y. Eidelman, I. Gohberg, On a new class of structured matrices, Integral Equations Operator Theory 34 (1999) 293–324.

[6] Y. Eidelman, I. Gohberg, Linear complexity inversion algorithms for a class of structured matrices, Integral Equations Operator Theory 35 (1999) 28–52.

[7] Y. Eidelman, I. Gohberg, Fast inversion algorithms for a class block structured matrices, in press.

[8] W.B. Gragg, The QR algorithm for unitary Hessenberg matrices, J. Comput. Appl. Math. 16 (1986) 1–8.

[9] P.E. Gill, G.H. Golub, W. Murray, M.A. Saunders, Methods for modifying matrix factorization, Math. Comp. 126 (1974) 505–535.

[10] I. Gohberg, M.A. Kaashoek, Time varying linear systems with boundary conditions and integral operators, 1. The transfer operator and its properties, Integral Equations Operator Theory 7 (1984) 325–391.

[11] I. Gohberg, T. Kailath, I. Koltracht, Linear complexity algorithms for semiseparable matrices, Integral Equations Operator Theory 8 (1985) 780–804.

[12] I. Gohberg, T. Kailath, I. Koltracht, A note on diagonal innovation, Acoustics Speech and Signal Processing 7 (1987) 1068–1069.

[13] G.H. Golub, C.F. Van Loan, Matrix Computations, John Hopkins, Baltimore, MD, 1983.

[14] N. Mastronardi, S. Chandrasekaran, S. Van Huffel, Fast and stable two-way algorithm for diagonal plus semi-separable systems of linear equations, Numer. Linear Algebra Appl. 8 (2001) 7–12.