Computing with quasiseparable matrices
Clément Pernet

To cite this version:

HAL Id: hal-01264131
https://hal.archives-ouvertes.fr/hal-01264131v2
Submitted on 19 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Computing with Quasiseparable Matrices

Clément Pernet
Univ. Grenoble Alpes Laboratoire LIP Inria, Université de Lyon
46, Allée d’Italie, F69364 Lyon Cedex 07, France
Clement.Pernet@imag.fr

ABSTRACT

The class of quasiseparable matrices is defined by a pair of bounds, called the quasiseparable orders, on the ranks of the sub-matrices entirely located in their strictly lower and upper triangular parts. These arise naturally in applications, as e.g., the inverse of band matrices, and are widely used for they admit structured representations allowing to compute with them in time linear in the dimension. We show, in this paper, the connection between the notion of quasiseparability and a matrix invariant, the rank profile matrix, that we introduced in [6]. The contribution of this paper, is to make the connection between the notion of quasiseparability and a matrix invariant, the rank profile matrix, that we introduced in [6].

1. INTRODUCTION

The inverse of a tridiagonal matrix, when it exists, is a dense matrix with the property that all sub-matrices entirely below or above its diagonal have rank at most one. This property and many generalizations of it, defining the semiseparable and quasiseparable matrices, have been extensively studied over the past 80 years. We refer to [16] and [17] for a broad bibliographic overview on the topic. In this paper, we will focus on the class of quasiseparable matrices, introduced in [8]:

**Definition 1.** An $n \times n$ matrix $M$ is $(r_L, r_U)$-quasiseparable if its strictly lower and upper triangular parts satisfy

the following low rank structure: for all $1 \leq k \leq n - 1$,

\[
\begin{align*}
rank(M_{k+1..n, 1..k}) & \leq r_L, \\
rank(M_{1..k, k+1..n}) & \leq r_U.
\end{align*}
\]

The values $r_L$ and $r_U$ define the quasiseparable orders of $M$.

Quasiseparable matrices can be represented with fewer than $n^2$ coefficients, using a structured representation, called a generator. The most commonly used generator [8, 16, 17, 9, 1] for a matrix $M$, consists of $(n - 1)$ pairs of vectors $p(i), q(i)$ of size $r_L$, $(n - 1)$ pairs of vectors $g(i), h(i)$ of size $r_U$, $n - 1$ matrices $a(i)$ of dimension $r_L \times r_L$, and $n - 1$ matrices $b(i)$ of dimension $r_U \times r_U$ such that

\[
M_{i,j} = \begin{cases} 
    p(i)^T a_{ij}^T g(j), & 1 \leq j < i \leq n \\
    d(i), & 1 \leq i = j \leq n \\
    g(i)^T b_{ij}^T h(j), & 1 \leq i < j \leq n 
\end{cases}
\]

where $a_{ij}^T = a(i - 1) \ldots a(j + 1)$ for $j > i$, $a_{j+1,j}^T = 1$, and $b_{ij}^T = b(i + 1) \ldots b(i - 1)$ for $i > j$, $b_{i+1,i}^T = 1$. This representation, of size $O(n(r_L^2 + r_U^2))$ makes it possible to apply a vector in $O(n(r_L^2 + r_U^2))$ field operations, multiply two quasiseparable matrices in time $O(n \max(r_L, r_U)^3)$ and also compute the inverse in time $O(n \max(r_L, r_U)^3)$ [8].

The contribution of this paper, is to make the connection between the notion of quasiseparability and a matrix invariant, the rank profile matrix, that we introduced in [6]. More precisely, we show that the PLUQ decompositions of the lower and upper triangular parts of a quasiseparable matrix, using a certain class of pivoting strategies, also have a structure ensuring that their memory footprint and the time complexity to compute them does not depend on the rank of the matrix but on the quasiseparable order (which can be arbitrarily lower). Note that we will assume throughout the paper that the PLUQ decomposition algorithms mentioned have the ability to reveal ranks. This is the case when computing with exact arithmetic (e.g., finite fields or multiprecision rationals), but not always with finite precision floating point arithmetic. In the latter context, a special care need to be taken for the pivoting of LU decompositions [10, 14], and QR or SVD decompositions are often more commonly used [2, 3]. This study is motivated by the design of new algorithms on polynomial matrices where quasiseparable matrices naturally occur, and more generally by the framework of the LinBox library [15] for black-box exact linear algebra.

After defining and recalling the properties of the rank profile matrix in Section 2, we propose in Section 3 an algorithm computing the quasiseparable orders $(r_L, r_U)$ in time $O(n^2 \omega^{-2})$ where $s = \max(r_L, r_U)$ and $\omega$ the exponent of
matrix multiplication. We then present in Section 4 two new structured representations, a binary tree of PLUQ decompositions, and the Bruhat generator, using respectively $O(ns\log \frac{n}{s})$ and $O(ns)$ field elements instead of $O(ns^2)$ for the previously known generators. We present in Section 5 algorithms computing them in time $O(n^2s^{2/3})$. These representations support a matrix-vector product in time linear in the size of their representation. Lastly we show how to multiply two such structured matrices in time $O(n^2s^{2/3})$.

Throughout the paper, $A_{i,j,k,l}$ will denote the sub-matrix of $A$ of row indices between $i$ and $j$ and column indices between $k$ and $l$. The matrix of the canonical basis, with a one at position $(i,j)$ will be denoted by $\Delta^{(i,j)}$.

2. PRELIMINARIES

2.1 Left triangular matrices

We will make intensive use of matrices with non-zero elements only located above the main anti-diagonal. We will refer to these matrices as left triangular, to avoid any confusion with upper triangular matrices.

**Definition 2.** A left triangular matrix is any $m \times n$ matrix $A$ such that $A_{i,j} = 0$ for all $i > n - j$.

The left triangular part of a matrix $A$, denoted by $Left(A)$ will refer to the left triangular matrix extracted from it. We will need the following property on the left triangular part of the product of a matrix by a triangular matrix.

**Lemma 1.** Let $A = BU$ be an $m \times n$ matrix where $U$ is $n \times n$ upper triangular. Then $Left(A) = Left(Left(B)U)$.

**Proof.** Let $\tilde{A} = Left(A), \tilde{B} = Left(B)$. For $j \leq n - i$, we have $\tilde{A}_{i,j} = \sum_{k=0}^{i-1} B_{i,k} \cdot U_{k,j} = \sum_{k=0}^{i-1} B_{i,k} \cdot U_{k,j}$ as $U$ is upper triangular. Now for $k \leq j \leq n - i$, $B_{i,k} = \tilde{B}_{i,k}$, which proves that the left triangular part of $A$ is that of $Left(B)U$. \qed

Applying Lemma 1 on $A^T$ yields Lemma 2

**Lemma 2.** Let $A = LB$ be an $m \times n$ matrix where $L$ is $m \times m$ lower triangular. Then $Left(A) = Left(LLeft(B))$.

Lastly, we will extend the notion of quasiseparable order to left triangular matrices, in the natural way: the left quasiseparable order is the maximal rank of any leading $k \times (n-k)$ sub-matrix. When no confusion may occur, we will abuse the definition and simply call it the quasiseparable order.

2.2 The rank profile matrix

We will use a matrix invariant, introduced in [6, Theorem 1], that summarizes the information on the ranks of any leading sub-matrices of a given input matrix.

**Definition 3.** [6, Theorem 1] The rank profile matrix of an $m \times n$ matrix $A$ of rank $r$ is the unique $m \times n$ matrix $R_A$, with only $r$ non-zero coefficients, all equal to one, located on distinct rows and columns such that any leading sub-matrices of $R_A$ has the same rank as the corresponding leading submatrix in $A$.

This invariant can be computed in just one Gaussian elimination of the matrix $A$, at the cost of $O(nm\omega^{-2})$ field operations [6], provided some conditions on the pivoting strategy being used. It is obtained from the corresponding PLUQ decomposition as the product

$$R_A = P \begin{bmatrix} 1_r & 0_{(m-r) \times (n-r)} \\ \end{bmatrix} Q.$$

We also recall in Theorem 1 an important property of such PLUQ decompositions revealing the rank profile matrix.

**Theorem 1** ([7, Th. 24], [5, Th. 1]). Let $A = PLU$ be a PLUQ decomposition revealing the rank profile matrix of $A$. Then, $P \begin{bmatrix} L & 0_{m \times (m-r)} \end{bmatrix} PT$ is lower triangular and $Q^T \begin{bmatrix} U & 0_{(n-r) \times n} \end{bmatrix} Q$ is upper triangular.

**Lemma 3.** The rank profile matrix invariant is preserved by multiplication

1. to the left with an invertible lower triangular matrix,
2. to the right with an invertible upper triangular matrix.

**Proof.** Let $B = LA$ for an invertible lower triangular matrix $L$. Then $\text{rank}(B_{i+1,i,j}) = \text{rank}(A_{i+1,i,j})$ for any $i \leq m, j \leq n$. Hence $R_B = R_A$. \qed

3. COMPUTING THE QUASISEPARABLE ORDERS

Let $M$ be an $m \times n$ matrix of which one want to determine the quasiseparable orders $(r_L, r_U)$. Let $L$ and $U$ be respectively the lower triangular part and the upper triangular part of $M$.

Let $J_m$ be the unit anti-diagonal matrix. Multiplying on the left by $J_m$ reverts the row order while multiplying on the right by $J_m$ reverts the column order. Hence both $J_mL$ and $UJ_n$ are left triangular matrices. Remark that the conditions (1) and (2) state that all leading $k \times (n-k)$ sub-matrices of $J_mL$ and $UJ_n$ have rank no greater than $r_L$ and $r_U$ respectively. We will then use the rank profile matrix of these two left triangular matrices to find these parameters.

3.1 From a rank profile matrix

First, note that the rank profile matrix of a left triangular matrix is not necessarily left triangular. For example, the rank profile matrix of $\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$ is $\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$. However, only the left triangular part of the rank profile matrix is sufficient to compute the left quasiseparable orders.

Suppose for the moment that the left-triangular part of the rank profile matrix of a left triangular matrix is given (returned by a function LT-RPM). It remains to enumerate all leading $k \times (n-k)$ sub-matrices and find the one with the largest number of non-zero elements. Algorithm 1 shows how to compute the largest rank of all leading sub-matrices of such a matrix. Run on $J_mL$ and $UJ_n$, it returns successively the quasiseparable orders $r_L$ and $r_U$.

This algorithm runs in $O(n)$ provided that the rank profile matrix $R$ is stored in a compact way, e.g. using a vector of $r$ pairs of pivot indices $\{(i_1, j_1), \ldots, (i_r, j_r)\}$.

3.2 Computing the rank profile matrix of a left triangular matrix

We now deal with the missing component: computing the left triangular part of the rank profile matrix of a left triangular matrix.
3.2.1 From a PLUQ decomposition

A first approach is to run any Gaussian elimination algorithm that can reveal the rank profile matrix, as described in [6]. In particular, the PLUQ decomposition algorithm of [5] computes the rank profile matrix of \(A\) in \(O(n^r - 2)\), where \(r = \text{rank}(A)\). However this estimate is pessimistic as it does not take into account the left triangular shape of the matrix. Moreover, this estimate does not depend on the left quasiseparable order \(s\) but on the rank \(r\), which may be much higher.

Remark 1. The discrepancy between the rank \(r\) of a left triangular matrix and its quasiseparable order arises from the location of the pivots in its rank profile matrix. Pivots located near the top left corner of the matrix are shared by many leading sub-matrices, and are therefore likely to contribute to the quasiseparable order. On the other hand, pivots near the anti-diagonal can be numerous, but do not add up to a large quasiseparable order. As an illustration, consider the two following extreme cases:

1. a matrix \(A\) with generic rank profile. Then the leading \(r \times r\) sub-matrix of \(A\) has rank \(r\) and the quasiseparable order is \(s \geq r\).

2. the matrix with \(n-1\) ones right above the anti-diagonal. It has rank \(r = n - 1\) but quasiseparable order 1.

Remark 1 indicates that in the unlucky cases when \(r \gg s\), the computation should reduce to instances of smaller sizes, hence a trade-off should exist between, on one hand, the discrepancy between \(r\) and \(s\), and on the other hand, the dimension \(n\) of the problems. All contributions presented in the remaining of the paper are based on such trade-offs.

3.2.2 A dedicated algorithm

In order to reach a complexity depending on \(s\) and not \(r\), we adapt in Algorithm 2 the tile recursive algorithm of [5], so that the left triangular structure of the input matrix is preserved and can be used to reduce the amount of computation.

Algorithm 2 does not assume that the input matrix is left triangular, as it will be called recursively with arbitrary matrices, but guarantees to return the left triangular part of the rank profile matrix. While the top left quadrant \(A_1\) is eliminated using any PLUQ decomposition algorithm revealing the rank profile matrix, the top right and bottom left quadrants are handled recursively.

**Theorem 2.** Given an \(n \times n\) input matrix \(A\) with left quasiseparable order \(s\), Algorithm 2 computes the left triangular part of the rank profile matrix of \(A\) in \(O(n^s s^2)\).

**Proof.** First remark that

\[
\begin{bmatrix}
D & F \\
L & T
\end{bmatrix}
\begin{bmatrix}
P_1 & 0 \\
0 & P_2
\end{bmatrix}
\begin{bmatrix}
U_1 & V_1 \\
0 & I
\end{bmatrix}
= \begin{bmatrix}
A_1 & A_2 \\
A_3 & 0
\end{bmatrix}.
\]

Hence

\[
L \begin{bmatrix}
A_1 & A_2 \\
A_3 & 0
\end{bmatrix} = P_1 \begin{bmatrix}
U_1 & V_1 \\
0 & I
\end{bmatrix} = \begin{bmatrix}
P_1 & 0 \\
0 & P_2
\end{bmatrix} \begin{bmatrix}
P_1 & 0 \\
0 & P_2
\end{bmatrix} T = \begin{bmatrix}
P_1 D F
\end{bmatrix}.
\]

By a similar reasoning, \([R_1, R_2]^T\) is the left triangular part of the rank profile matrix of \([A_1, A_2]^T\), which shows that the algorithm is correct.

Let \(s_1\) be the left quasiseparable order of \(H\) and \(s_2\) that of \(I\). The number of field operations to run Algorithm 2 is

\[
T(n, s) = \omega T(n, s_1) + T_{\text{LT-RPM}}(n/2, s_1) + T_{\text{LT-RPM}}(n/2, s_2)
\]

where

\[
\omega = \begin{cases}
\omega(\text{FieldOp}) & \text{for } s_1 = s_2 = 1 \\
\omega(\text{FieldOp}) & \text{for } s_1 = 2, s_2 = 1 \\
\omega(\text{FieldOp}) & \text{for } s_1 = 1, s_2 = 2 \\
\omega(\text{FieldOp}) & \text{for } s_1 = s_2 = 2
\end{cases}
\]
for a positive constant $\alpha$. We will prove by induction that $T(n, s) \leq 2\alpha s^{2-2}n^2$.

Again, since $L$ is lower triangular, the rank profile matrix of $LA_k$ is that of $A_2$ and the quasiseparable orders of the two matrices are the same. Now $H$ is the matrix $LH$ with some rows zeroed out, hence $s_1$, the quasiseparable order of $H$ is no greater than that of $A_2$ which is less or equal to $s$. Hence $\max(r_1, s_1, s_2) \leq s$ and we obtain $T(n, s) \leq \alpha s^{2-2}n^2 + 4\alpha s^{2-2}(n/2)^2 = 2\alpha s^{2-2}n^2$. □

4. MORE COMPACT GENERATORS

Taking advantage of their low rank property, quasiseparable matrices can be represented by a structured representation allowing to compute efficiently with them, as for example in the context of QR or QZ elimination [9, 1].

The most commonly used generator, as described in [8, 1] and in the introduction, represents an $(r_L, r_U)$-quasiseparable matrix of order $n$ by $O(n(r_L^2 + r_U^2))$ field coefficients.

Alternatively, hierarchically semiseparable representations (HSS) [18, 11] use numerical rank revealing factorizations of the off-diagonal blocks in a divide and conquer approach, reducing the size to $O(\max(r_L, r_U) n \log n)$ [11].

A third approach, based on Givens or unitary weights [4], performs another kind of elimination so as to compact the low rank off-diagonal blocks of the input matrix.

We propose, in this section, two alternative generators, based on an exact PLUQ decomposition revealing the rank profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix. The first one matches the best space based on an exact PLUQ decomposition revealing the rank profile matrix. The second one improves the time profile matrix.

4.1 A binary tree of PLUQ decompositions

Following the divide and conquer scheme of Algorithm 2, we propose a first generator requiring

$$O(n(r_L \log \frac{n}{r_L} + r_U \log \frac{n}{r_U}))$$ (3)

coefficients.

For a left triangular matrix $A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$, the sub-matrix $A_1$ is represented by its PLUQ decomposition $(P_1, L_1, U_1, Q_1)$, which requires $2n_1 \frac{s}{2} \leq sn_1$ field coefficients for $L_1$ and $U_1$ and $2n_2$ indices for $P$ and $Q$. This scheme is then recursively applied for the representation of $A_2$ and $A_3$. These matrices have quasiseparable order at most $s$, therefore the following

Note that the statement of $O(n(r_L + r_U))$ for the same generator in [9] is erroneous.

4.2 The Bruhat generator

We propose an alternative generator inspired by the generalized Bruhat decomposition [13, 12, 7]. Contrarily to the former one, it is not depending on a specific recursive cutting of the matrix.

Given a left triangular matrix $A$ of quasiseparable order $s$ and a PLUQ decomposition of it, revealing its rank profile matrix $E$, the generator consists in the three matrices

$$L = \text{Left}(P [ L ] 0 | Q), \quad E = \text{Left}(E), \quad U = \text{Left}(P [ U ] 0 | Q).$$ (4) (5) (6)

Lemma 4 shows that these three matrices suffice to recover the initial left triangular matrix.

**Lemma 4.** $A = \text{Left}(LE^T U)$

**Proof.** $A = P [ L ] 0_{0 \times (n-r)} Q Q^T \begin{bmatrix} U \\ 0_{(n-s) \times n} \end{bmatrix} Q.$ From Theorem 1, the matrix $Q^T \begin{bmatrix} U \\ 0 \end{bmatrix} Q$ is upper triangular and the matrix $P [ L ] 0 P^T$ is lower triangular. Applying Lemma 1 yields $A = \text{Left}(A) = \text{Left}(Q^T \begin{bmatrix} U \\ 0 \end{bmatrix} Q) = \text{Left}(LE^P \begin{bmatrix} U \\ 0 \end{bmatrix} Q)$, where $E = P [ I ] Q$. Then, as $LE^T$ is the matrix $P [ L ] 0 P^T$ with some coefficients zeroed out, it is lower triangular, hence applying again Lemma 2 yields

$$A = \text{Left}(LE^T U).$$ (7)

Consider any non-zero coefficient $c_{j,i}$ of $E^T$ that is not in its the left triangular part, i.e. $j > n - i$. Its contribution to the product $LE^T$, is only of the form $L_{k,j}c_{j,i}$. However the leading coefficient in column $j$ of $P [ L ] 0 Q$ is precisely at position $(i, j)$. Since $i > n - j$, this means that the $j$-th column of $L$ is all zero, and therefore $c_{j,i}$ has no contribution to the product. Hence we finally have $A = \text{Left}(LE^T U)$. □

We now analyze the space required by this generator.

**Lemma 5.** Consider an $n \times n$ left triangular rank profile matrix $R$ with quasiseparable order $s$. Then a left triangular matrix $L$ all zero except at the positions of the pivots of $R$ and below these pivots, does not contain more than $s(n - s)$ non-zero coefficients.
Proof. Let \( p(k) = \text{rank}(R_{1-k,1-n-k}) \). The value \( p(k) \) indicates the number of non-zero columns located in the \( k \times n - k \) leading sub-matrix of \( L \). Consequently the sum \( \sum_{k=1}^{n-1} p(k) \) is an upper bound on the number of non-zero coefficients in \( L \). Since \( p(k) \leq s \), it is bounded by \( sn \). More precisely, there is no more than \( k \) pivots in the first \( k \) columns and the first \( k \) rows, hence \( p(k) \leq k \) and \( p(n-k) \leq k \) for \( k \leq s \). The bound becomes \( s(s+1) + (n-2s-1)s = s(n-s) \). \( \square \)

Corollary 1. The generator \( (L, E, U) \) uses \( 2s(n-s) \) field coefficients and \( O(n) \) additional indices.

Proof. The leading column elements of \( L \) are located at the pivot positions of the left triangular rank profile matrix \( E \). Lemma 5 can therefore be applied to show that this matrix occupies no more than \( s(n-s) \) non-zero coefficients. The same argument applies to the matrix \( U \). \( \square \)

Figure 1 illustrates this generator on a left triangular matrix of quasiseparable order 5. As the supports of \( L \) and \( U \) are disjoint, the two matrices can be shown on the same left triangular matrix. The pivots of \( E \) (black) are the leading coefficients of every non-zero row of \( U \) and non-zero column of \( L \).

Corollary 2. Any \((r_L, r_U)\)-quasiseparable matrix of dimension \( n \times n \) can be represented by a generator using no more than \( 2n(r_L + r_U) + n - 2(r_L^2 - 2r_U^2) \) field elements.

4.3 The compact Bruhat generator

The sparse structure of the Bruhat generator makes it not amenable to the use of fast matrix arithmetic. We therefore propose here a slight variation of it, that we will use in section 5 for fast complexity estimates. We will first describe this compact representation for the \( L \) factor of the Bruhat generator.

First, remark that there exists a permutation matrix \( Q \) moving the non-zero columns of \( L \) to the first \( r \) positions, sorted by increasing leading row index, i.e. such that \( EQ \) is in column echelon form. The matrix \( LE \) or \( LEQ \) is now compacted, but still has \( r = \text{rank}(A) \) columns, which may exceed \( s \) and thus preventing to reach complexities in terms of \( n \) and \( s \) only. We will again use the argument of Lemma 5 to produce a more compact representation with only \( O(ns) \) non-zero elements, stored in dense blocks. Algorithm 3 shows how to build such a representation composed of a block diagonal matrix and a block sub-diagonal matrix, where all blocks have column dimension \( s \):

\[
\begin{bmatrix}
S_1 & D_2 & S_3 & D_4 & \cdots & \cdots & \cdots \\
S_2 & & & & & & \\
S_3 & & & & & & \\
& & & & & & \\
& & & & & & \\
& & & & & & \\
& & & & & & \\
& & & & & & \\
& & & & & & \\
\end{bmatrix}
\]

Algorithm 3 Compressing the Bruhat generator

Input: \( L \): the first matrix of the Bruhat generator
Output: \( D, S, T, Q \): the compression of \( L \)

1: \( Q \leftarrow \text{a permutation s.t. } EQ \text{ is in column echelon form} \)
2: \( C \leftarrow LQ \lfloor \frac{r}{s} \rfloor \) where \( r = \text{rank}(L) \)
3: Split \( C \) in column slices of width \( s \).
4: \( D \leftarrow \text{Diag}(C_{11}, \ldots, C_{n1}) \)
5: \( C \leftarrow C - D \)
6: \( T \leftarrow \text{I}_{n} \)
7: for \( i = 3 \ldots t \) do
8: \( \text{for each non zero column } j \) of \( C_{i,i-1} \) do
9: \( \text{Let } k \text{ be a zero column of } C_{i,i-1} \)
10: \( \text{Move col. } j \text{ to col. } k \text{ in } C_{i,i-1} \)
11: end for
12: end for
13: \( S \leftarrow C \)
14: \( \text{Return } (D, S, T, Q) \)

Lemma 6. Algorithm 3 computes a tuple \((D, S, T, Q)\) where \( Q \) is a permutation matrix putting \( L \) in column echelon form, \( T \in \{0,1\}^{r \times r}, D = \text{Diag}(D_1, \ldots, D_s) \), \( S = \begin{bmatrix} 0 & D_2 & 0 & \cdots & 0 \\
S_2 & \cdots & \cdots & \cdots & \cdots \\
S_3 & \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
& \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} \)

where each \( D_i \) and \( S_i \) is \( k_i \times s \) for \( k_i \geq s \) and \( \sum_{i=1}^{t} k_i = n \). This tuple is the compact Bruhat generator for \( L \) and satisfies \( L = [D + ST \ 0_{n \times (n-r)}]Q^T \).

Proof. First, note that for every \( i \), the dimensions of the blocks \( S_i \) and \( D_i \) are that of the block \( C_{ii} \). This block contains \( s \) pivots, hence \( k_i \geq s \). We then prove that there always exists a zero column to pick at step 10. The loci of the possible non-zero elements in \( L \) are column segments below a pivot and above the anti-diagonal. From Lemma 5, these segments have the property that each row of \( L \) is intersected by no more than \( s \) of them. This property is preserved by column permutation, and still holds on the matrix \( C \). In the first row of \( [C_{11} \ldots C_{1t-1}] \), there is a pivot located in the block \( C_{11} \). Hence there is at most \( s-1 \) such segments intersecting \( [C_{11} \ldots C_{1t-1}] \). These \( s-1 \) columns can all be gathered in the block \( C_{1t-1} \) of column dimension \( s \).
5. COST OF COMPUTING WITH THE NEW GENERATORS

5.1 Computation of the generators

5.1.1 The binary tree generators

Let $T_1(n, s)$ denote the cost of the computation of the binary tree generator for an $n \times n$ matrix of order of quasiseparability $s$. It satisfies the recurrence relation $T_1(n, s) = K_\omega s^{\omega-2} \left(\frac{s}{2}\right)^2 + 2T_1(n/2, s)$, which solves in

$$T(n, s) = \frac{K_\omega}{2} s^{\omega-2} n^2$$

with $K_\omega = \frac{2^{\omega-2}}{(2^{\omega-2} - 1)}$. Here $C_\omega$ is the leading constant of the complexity of matrix multiplication [5].

5.1.2 The Bruhat generator

We propose in Algorithm 4 an evolution of Algorithm 2 to compute the factors of the Bruhat generator.

Algorithm 4 LT-Bruhat

Input: $A$: an $n \times n$ matrix
Output: $(\mathcal{L}, \mathcal{E}, \mathcal{U})$: a Bruhat generator for the left triangular part of $A$
1: if $n = 1$ then return $([0], [0], [0])$
2: Split $A = [A_1, A_2]$ where $A_1$ is $[\frac{n}{2}] \times [\frac{n}{2}]$
3: Decompose $A_1 = P_1 \begin{bmatrix} L_1 & M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1$ \quad \triangleright PLU(A_1)
4: $R_1 \leftarrow P_1 \begin{bmatrix} I_1 & 0 \end{bmatrix} Q_1$ where $r_1 = \text{rank}(A_1)$.
5: $[B_1, B_2] \leftarrow P_1^T A_2$ \quad \triangleright PermR(A_2, P_1^T)
6: $[C_1, C_2] \leftarrow A_2 Q_1^T$ \quad \triangleright PermC(A_3, Q_1^T)
7: Here $A = \begin{bmatrix} L_1 \cup U_1 & V_1 & I & B_1 \\ M_1 & 0 & B_2 \\ C_1 & C_2 \end{bmatrix}$
8: $D \leftarrow L_1^{-1} B_1$ \quad \triangleright TRSM(L_1, B_1)
9: $E \leftarrow C_1 U_1$ \quad \triangleright TRSM(C_1, U_1)
10: $F \leftarrow B_2 - M_1$ \quad \triangleright MM(B_2, M_1, D)
11: $G \leftarrow C_2 - EV_1$ \quad \triangleright MM(C_2, E, V_1)
12: Here $A = \begin{bmatrix} L_1 \cup U_1 & V_1 & I & D \\ M_1 & 0 & F \\ C_1 & C_2 \end{bmatrix}$
13: $H \leftarrow P_1 \begin{bmatrix} I_{r_1} & 0 \\ 0 & 1 \end{bmatrix} \quad \triangleright \text{TRSM}(L_1, B_1)$
14: $I \leftarrow [0_{n-r_1}, 1, \frac{n}{2}] \quad \triangleright \text{TRSM}(C_1, U_1)$
15: $(\mathcal{L}_2, \mathcal{E}_2, \mathcal{U}_2) \leftarrow \text{LT-Bruhat}(H)$
16: $(\mathcal{L}_3, \mathcal{E}_3, \mathcal{U}_3) \leftarrow \text{LT-Bruhat}(I)$
17: $\mathcal{L} \leftarrow \begin{bmatrix} P_1 & 0 \\ I_{r} & 0 \end{bmatrix} \begin{bmatrix} L_1 & M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1$ \quad \triangleright $\mathcal{L}'(P_1, A)$
18: $\mathcal{U} \leftarrow \begin{bmatrix} P_1 & 0 \\ I_{r} & 0 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1$ \quad \triangleright $\mathcal{U}'(P_1, A)$
19: $\mathcal{E} \leftarrow [\mathcal{E}_1, \mathcal{E}_2]$
20: return $(\mathcal{L}, \mathcal{E}, \mathcal{U})$

Theorem 3. For any $n \times n$ matrix $A$ with a left triangular part of quasiseparable order $s$, Algorithm 4 computes the Bruhat generator of the left triangular part of $A$ in $O(s^{\omega-2} n^2)$ field operations.
Proof. The correctness of $\mathcal{L}$ is proven in Theorem 2. We will prove by induction the correctness of $\mathcal{U}$, noting that the correctness of $\mathcal{L}$ works similarly.

Let $H = P_2 L_2 U_2 Q_2$ and $I = P_3 L_3 U_3 Q_3$ be PLUQ decompositions of $H$ and $I$ revealing their rank profile matrices. Assume that Algorithm LT-Bruhat is correct in the two recursive calls 15 and 16, that is

$$U_2 = \text{Left}(P_2 \begin{bmatrix} U_2^T & 0 \end{bmatrix} Q_2), \quad U_3 = \text{Left}(P_3 \begin{bmatrix} U_3^T & 0 \end{bmatrix} Q_3),$$

$$L_2 = \text{Left}(P_2 \begin{bmatrix} L_2^T & 0 \end{bmatrix} Q_2), \quad L_3 = \text{Left}(P_3 \begin{bmatrix} L_3^T & 0 \end{bmatrix} Q_3).$$

At step 7, we have

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} A_3 \end{bmatrix}^* \begin{bmatrix} \begin{bmatrix} P_1 & P_3 \end{bmatrix} \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \begin{bmatrix} L_1 & M_1 & L_2 - r_1 \end{bmatrix} & E & \frac{1}{n}\begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} D \frac{1}{T} \end{bmatrix} & \begin{bmatrix} Q_1 \end{bmatrix} & \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} P_3^T & P_2 \end{bmatrix} \begin{bmatrix} L_1 & M_1 & L_2 - r_1 \end{bmatrix} & E & \frac{1}{n}\begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} \bar{P}_2 \bar{L}_2 \end{bmatrix} & \begin{bmatrix} Q_3 \end{bmatrix} & \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \end{bmatrix} \end{bmatrix}}$$

As the first $r_1$ rows of $P_2^T H$ are zeros, there exists $\bar{P}_2$ a permutation matrix of $L_2$, a lower triangular matrix, such that $P_2^T P_2 L_2 = \begin{bmatrix} 0_{r_1 \times \frac{n}{r}} & \bar{P}_2 \bar{L}_2 \end{bmatrix}$. Similarly, there exist $\bar{Q}_3$, a permutation matrix and $\bar{U}_1$, an upper triangular matrix, such that $U_1 Q_1 Q_3^T = \begin{bmatrix} 0_{r_1 \times \frac{n}{r}} & \bar{U}_1 \bar{Q}_3 \end{bmatrix}$. Hence

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} A_3 \end{bmatrix}^* \begin{bmatrix} \begin{bmatrix} P_1 & P_3 \end{bmatrix} \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \begin{bmatrix} L_1 & M_1 & L_2 - r_1 \end{bmatrix} & E & \frac{1}{n}\begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} D \frac{1}{T} \end{bmatrix} & \begin{bmatrix} Q_1 \end{bmatrix} & \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} P_3^T & P_2 \end{bmatrix} \begin{bmatrix} L_1 & M_1 & L_2 - r_1 \end{bmatrix} & E & \frac{1}{n}\begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} \bar{P}_2 \bar{L}_2 \end{bmatrix} & \begin{bmatrix} Q_3 \end{bmatrix} & \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \end{bmatrix} \end{bmatrix}}$$

Setting $N_1 = \bar{P}_2^T M_1$ and $W_1 = V_1 \bar{Q}_3^T$, we have

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} A_3 \end{bmatrix}^* = \begin{bmatrix} \begin{bmatrix} P_1 & P_3 \end{bmatrix} \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \begin{bmatrix} L_1 & M_1 & L_2 - r_1 \end{bmatrix} & E & \frac{1}{n}\begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} D \frac{1}{T} \end{bmatrix} & \begin{bmatrix} Q_1 \end{bmatrix} & \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} P_3^T & P_2 \end{bmatrix} \begin{bmatrix} L_1 & M_1 & L_2 - r_1 \end{bmatrix} & E & \frac{1}{n}\begin{bmatrix} 1 \end{bmatrix} \\ \begin{bmatrix} \bar{P}_2 \bar{L}_2 \end{bmatrix} & \begin{bmatrix} Q_3 \end{bmatrix} & \begin{bmatrix} 1 \frac{1}{T} \end{bmatrix} \end{bmatrix} \end{bmatrix}.$$

A PLUQ of $\begin{bmatrix} A_1 & A_2 \end{bmatrix}$ revealing its rank profile matrix is then obtained from this decomposition by a row block cyclic-shift on the second factor and a column block cyclic shift on the third factor as in $[5, Algorithm 1]$.

Finally,

$$P \begin{bmatrix} U_1 & V_1 & Q_1 & P_1 \end{bmatrix}^T \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} P_1 & 0 & 0 & P_3 \begin{bmatrix} U_1 & V_1 \end{bmatrix}^T Q_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}. \begin{bmatrix} P_2 & U_2 & Q_2 \end{bmatrix} \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} P_1 & 0 & 0 & P_3 \begin{bmatrix} U_1 & V_1 \end{bmatrix}^T Q_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}. \begin{bmatrix} P_2 & U_2 & Q_2 \end{bmatrix} \begin{bmatrix} D \end{bmatrix}.$$
consecutive block rows of a flat matrix lie in the same block column. Consequently these terms can be decomposed as a sum of two block diagonal matrices aligned on an $s \times s$ grid. Multiplying two such matrices costs $O(s^{\omega-1}n)$ which is consequently also the cost of computing the product $E_A C_B$. After left and right multiplication by the permutations $R_A$ and $R_B$, this $r_A \times r_B$ dense matrix is multiplied to the left by $C_A$. This costs $O(nr_B s^{\omega-2})$. Lastly, the right multiplication by $E_B$ of the resulting $n \times r_A$ matrix costs $O(n^2 s^{\omega-2})$ which dominates the overall cost.

5.4 Multiplying two quasiseparable matrices

Decomposing each multiplicand into its upper, lower and diagonal terms, a product of two quasiseparable matrices writes $A \times B = (L_A + D_A + U_A)(L_B + D_B + U_B)$. Beside the scaling by diagonal matrices, all other operations involve a product between any combination of lower an upper triangular matrices, which in turn translates into products of left triangular matrices and $J_n$ as shows in Table 1. The complexity of section 5.3 directly applies for the computation of $Upper \times Lower$ and $Lower \times Upper$ products. For the other products, a $J_n$ factor has to be added between the $E_A$ and $C_B$ factors in the innermost product of (9). As reversion the row order of $C_B$ does not impact the cost of computing this product, the same complexity applies here too.

<table>
<thead>
<tr>
<th>x</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower</td>
<td>$J_n \times Left \times J_n \times Left$</td>
<td>$J_n \times Left \times Left \times J_n$</td>
</tr>
<tr>
<td>Upper</td>
<td>$Left \times J_n \times Left \times Left$</td>
<td>$Left \times J_n \times Left \times J_n$</td>
</tr>
</tbody>
</table>

Table 1: Reducing products of lower and upper to products of left triangular matrices.

Theorem 4. Multiplying two quasiseparable matrices of order respectively $(l_A, u_A)$ and $(l_B, u_B)$ costs $O(n^2 s^{\omega-2})$ field operations where $s = \max(l_A, u_A, l_B, u_B)$, using either one of the binary tree or the compact Bruhat generator.

6. PERSPECTIVES

The algorithms proposed for multiplying two quasiseparable matrices output a dense $n \times n$ matrix in time $O(n^2 s^{\omega-2})$ for $s = \max(l_A, u_A, l_B, u_B)$. However, the product is also a quasiseparable matrix, of order $(l_A + l_B, u_A + u_B)$ [8, Theorem 4.1], which can be represented by a Bruhat generator with only $O(n(l_A + l_B + u_A + u_B))$ coefficients. A first natural question is thus to find an algorithm computing this representation from the generators of $A$ and $B$ in time $O(ns^{\omega-1})$.

Second, a probabilistic algorithm [7, §7] reduces the complexity of computing the rank profile matrix to $O^*(n^2 + r^2)$. It is not clear whether it can be applied to compute a compact Bruhat generator in time $O^*(n^2 + \max(l_A, u_A)^2)$.

Note (added Sept. 16, 2016.)

Equation (9) for the multiplication of two Bruhat generators is missing the Left operators, and is therefore incorrect. The target complexities can still be obtained by slight modification of the algorithm: computing the inner-most product $E_A \times C_B$ as an unevaluated sum of blocks products. This will be detailed in a follow-up paper.

Acknowledgment

We thank Paola Boito for introducing us to the field of quasiseparable matrices and two anonymous referees for pointing us to the HSS and the Givens weight representations. We acknowledge the financial support from the HPAC project (ANR 11 BS02 013) and from the OpenDreamKit Horizon 2020 European Research Infrastructures project (#676541).

7. REFERENCES


