



HAL
open science

Computing with quasiseparable matrices

Clément Pernet

► **To cite this version:**

| Clément Pernet. Computing with quasiseparable matrices. 2016. hal-01264131v1

HAL Id: hal-01264131

<https://hal.science/hal-01264131v1>

Preprint submitted on 28 Jan 2016 (v1), last revised 19 Sep 2016 (v2)

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 maximal 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 and quadratic with the quasiseparable order. We show, in this paper, the connection between the notion of quasiseparability and the rank profile matrix invariant, presented in [Dumas & al. ISSAC'15]. This allows us to propose an algorithm computing the quasiseparable orders (r_L, r_U) in time $O(n^2 s^{\omega-2})$ where $s = \max(r_L, r_U)$ and ω the exponent of matrix multiplication. We then present 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 algorithms computing these representations in time $O(n^2 s^{\omega-2})$. These representations allow 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 $O(n^2 s^{\omega-2})$.

1. INTRODUCTION

The inverse of a tridiagonal matrix, when it exists, is a dense matrix with the property that all maximal sub-matrices entirely below or above its diagonal have rank 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 [9] and [10] for a broad bibliographic overview on the topic. In this paper, we will focus on the class of quasiseparable matrices, introduced in [5]:

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 $1 \leq k \leq n - 1$,

$$\text{rank}(M_{k+1..n, 1..k}) \leq r_L, \quad (1)$$

$$\text{rank}(M_{1..k, k+1..n}) \leq r_U. \quad (2)$$

The integers r_L and r_U are called 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 [5, 9, 10, 6, 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

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

where $a_{ij}^> = a(i - 1) \dots a(j + 1)$ for $j > i + 1$, $a_{j+1,j} = 1$, and $b_{ij}^< = b(i + 1) \dots b(i - 1)$ for $i > j + 1$, $b_{i,i+1} = 1$. 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 $O(n \max(r_L, r_U)^3)$ [5].

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 [3]. 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).

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 $O(n^2 s^{\omega-2})$ where $s = \max(r_L, r_U)$ and ω 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 $O(n^2 s^{\omega-2})$. 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 $O(n^2 s^{\omega-2})$.

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 $\text{Left}(A)$ 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 $\text{Left}(A) = \text{Left}(\text{Left}(B)U)$.*

PROOF. Let $\bar{A} = \text{Left}(A)$, $\bar{B} = \text{Left}(B)$. For $j \leq n - i$, we have

$$\bar{A}_{i,j} = \sum_{k=1}^n B_{i,k} \cdot U_{k,j} = \sum_{k=1}^j B_{i,k} \cdot U_{k,j}$$

as U is upper triangular. Now for $k \leq j \leq n - i$, $B_{i,k} = \bar{B}_{i,k}$, which proves that the left triangular part of A is that of $\text{Left}(B)U$. \square

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 $\text{Left}(A) = \text{Left}(L\text{Left}(B))$.*

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 [3, Theorem 1], that summarizes the information on the ranks of any leading sub-matrices of a given input matrix.

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

This invariant can be computed in just one Gaussian elimination of the matrix A , at the cost of $O(mnr^{\omega-2})$ field operations [3], provided some conditions on the pivoting strategy being used. It is obtained from the corresponding PLUQ decomposition as the product

$$\mathcal{R}_A = P \begin{bmatrix} I_r & \\ & 0_{(m-r) \times (n-r)} \end{bmatrix} Q.$$

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

THEOREM 1 ([4, THEOREM 24], [2, THEOREM 1]). *Let $A = \text{PLUQ}$ be a PLUQ decomposition revealing the rank profile matrix of A . Then, $P \begin{bmatrix} L & 0_{m \times (m-r)} \end{bmatrix} P^T$ 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*

1. *left multiplication with an invertible lower triangular matrix,*
2. *right multiplication with an invertible upper triangular matrix.*

PROOF. Let $B = LA$ for an invertible lower triangular matrix L . Then $\text{rank}(B_{1..i,1..j}) = \text{rank}(L_{1..i,1..i} A_{1..i,1..j}) = \text{rank}(A_{1..i,1..j})$ for any $i \leq m, j \leq n$. Hence $\mathcal{R}_B = \mathcal{R}_A$. \square

3. COMPUTING THE QUASISEPARABLE ORDERS

Let M be an $n \times n$ 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_n be the unit anti-diagonal matrix. Multiplying on the left by J_n reverts the row order while multiplying on the right by J_n reverts the column order. Hence both $J_n L$ and $U J_n$ are left triangular matrices. Remark that the conditions (1) and (2) state that all leading $k \times (n - k)$ sub-matrices of $J_n L$ and $U J_n$ 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 $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ is $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. 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 $J_n L$ and $U J_n$, 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 $([(i_1, j_1), \dots, (i_r, j_r)])$.

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 [3]. In particular, the PLUQ decomposition algorithm of [2] computes the rank profile matrix of A in $O(n^2 r^{\omega-2})$

Algorithm 1 QS-order

Input: A , an $n \times n$ matrix**Output:** $\max\{\text{rank}(A_{1..k,1..n-k}) : 1 \leq k \leq n-1\}$
 $R \leftarrow \text{LT-RPM}(A)$ ▷ The left triangular part of the rank profile matrix of A
rows $\leftarrow (\text{False}, \dots, \text{False})$
cols $\leftarrow (\text{False}, \dots, \text{False})$
for all (i, j) such that $R_{i,j} = 1$ **do**
 rows $[i] \leftarrow \text{True}$
 cols $[j] \leftarrow \text{True}$
end for
 $s, r \leftarrow 0$
for $i = 1 \dots n$ **do**
 if rows $[i]$ **then**
 $r \leftarrow r + 1$
 end if
 if cols $[n - i + 1]$ **then**
 $r \leftarrow r - 1$
 end if
 if cols $[i]$ **then**
 $s \leftarrow \max(s, r)$
 end for
Return s

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 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 $s = r$.
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 [2], 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

Algorithm 2 LT-RPM: Left Triangular part of the Rank Profile Matrix

Input: A : an $n \times n$ matrix**Output:** \mathcal{R} : the left triangular part of the rank profile matrix of A

```
1: if  $n = 1$  then
2:   Return  $[0]$ 
3: end if
4: Split  $A = \begin{bmatrix} A_1 & A_2 \\ & A_3 \end{bmatrix}$  where  $A_3$  is  $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$ 
5: Decompose  $A_1 = P_1 \begin{bmatrix} L_1 \\ M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1$ 
6:  $\mathcal{R}_1 \leftarrow P_1 \begin{bmatrix} I_{r_1} & \\ & 0 \end{bmatrix} Q_1$  where  $r_1 = \text{rank}(A_1)$ .
7:  $\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \leftarrow P_1^T A_2$ 
8:  $\begin{bmatrix} C_1 & C_2 \end{bmatrix} \leftarrow A_3 Q_1^T$ 
9: Here  $A = \left[ \begin{array}{cc|c} L_1 \setminus U_1 & V_1 & B_1 \\ M_1 & 0 & B_2 \\ \hline C_1 & C_2 & \end{array} \right]$ .
10:  $D \leftarrow L_1^{-1} B_1$ 
11:  $E \leftarrow C_1 U_1^{-1}$ 
12:  $F \leftarrow B_2 - M_1 D$ 
13:  $G \leftarrow C_2 - E V_1$ 
14: Here  $A = \left[ \begin{array}{cc|c} L_1 \setminus U_1 & V_1 & D \\ M_1 & 0 & F \\ \hline E & G & \end{array} \right]$ .
15:  $H \leftarrow P_1 \begin{bmatrix} 0_{r_1 \times \frac{n}{2}} \\ F \end{bmatrix}$ 
16:  $I \leftarrow \begin{bmatrix} 0_{r_1 \times \frac{n}{2}} & G \end{bmatrix} Q_1$ 
17:  $\mathcal{R}_2 \leftarrow \text{LT-RPM}(H)$ 
18:  $\mathcal{R}_3 \leftarrow \text{LT-RPM}(I)$ 
19: Return  $\mathcal{R} \leftarrow \begin{bmatrix} \mathcal{R}_1 & \mathcal{R}_2 \\ & \mathcal{R}_3 \end{bmatrix}$ 
```

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 $O(n^2 s^{\omega-2})$.*

PROOF. First remark that

$$P_1 \begin{bmatrix} D \\ F \end{bmatrix} = P_1 \underbrace{\begin{bmatrix} L_1^{-1} & & \\ -M_1 L_1^{-1} & I_{n-r_1} & \end{bmatrix}}_L P_1^T P_1 \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = L A_2.$$

Hence

$$L \begin{bmatrix} A_1 & A_2 \end{bmatrix} = P_1 \left[\begin{array}{cc|c} [U_1 & V_1] Q_1 & D \\ & 0 & F \end{array} \right].$$

From **Theorem 1**, the matrix L is lower triangular and by **Lemma 3** the rank profile matrix of $\begin{bmatrix} A_1 & A_2 \end{bmatrix}$ equals that of $P_1 \left[\begin{array}{cc|c} [U_1 & V_1] Q_1 & D \\ & 0 & F \end{array} \right]$. As U_1 is upper triangular and non-

singular, this rank profile matrix is that of $P_1 \left[\begin{array}{cc|c} [U_1 & V_1] Q_1 & 0 \\ & 0 & F \end{array} \right]$

and its left triangular part is $\begin{bmatrix} \mathcal{R}_1 & \mathcal{R}_2 \end{bmatrix}$.

By a similar reasoning, $\begin{bmatrix} \mathcal{R}_1 & \mathcal{R}_3 \end{bmatrix}^T$ is the left triangular part of the rank profile matrix of $\begin{bmatrix} A_1 & A_3 \end{bmatrix}^T$, 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

$$T(n, s) = \alpha r_1^{\omega-2} n^2 + T_{\text{LT-RPM}}(n/2, s_1) + T_{\text{LT-RPM}}(n/2, s_2)$$

for a positive constant α . We will prove by induction that $T(n, s) \leq 2\alpha s^{\omega-2} n^2$.

Again, since L is lower triangular, the rank profile matrix of LA_2 is that of A_2 and the quasiseparable orders of the two matrices are the same. Now H is the matrix LA_2 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 $T(n, s) \leq \alpha s^{\omega-2} n^2 + 4\alpha s^{\omega-2} (n/2)^2 = 2\alpha s^{\omega-2} n^2$. \square

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 [6, 1].

The usual generator, as described in [5, 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 coefficients¹. We propose, in this section, two alternative generators, based on a PLUQ decomposition, with a space complexity linear in the quasiseparable order.

First, remark that storing a PLUQ decomposition of rank r and dimension $n \times n$ uses $2rn - r^2$ coefficients: each of the L and U factor has dimension $n \times r$ or $r \times n$; the negative r^2 term comes from the lower and upper triangular shapes of L and U . Here again, the rank r can be larger than the quasiseparable order s thus storing directly a PLUQ decomposition is too expensive. But as in Remark 1, the setting where $r \gg s$ is precisely when the pivots are near the anti-diagonal, and therefore the L and U factors have an additional structure, with numerous zeros. The two proposed generators, rely on this fact.

4.1 A binary tree of PLUQ decompositions

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

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

coefficients.

For a left triangular matrix $A = \begin{bmatrix} A_1 & A_2 \\ & A_3 \end{bmatrix}$, the sub-matrix A_1 is represented by its PLUQ decomposition (P_1, L_1, U_1, Q_1) , which requires $2r_1 \frac{n}{2} \leq sn$ field coefficients for L_1 and U_1 and $2n$ 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 recurrence relation for the size of the representation holds:

$$\begin{cases} S(n, s) = sn + 2S(n/2, s) & \text{for } s < n/2 \\ S(n, s) = \frac{n^2}{2} + 2S(n/2, n/4) & \text{for } s \geq n/2 \end{cases}$$

For $s \geq n/2$, it solves in $S(n, s) = n^2$. Then for $s < n/2$, $S(n, s) = sn + 2sn/2 + \dots + 2^k S(n/2^k, s)$, for k such that $\frac{n}{2^k} \leq s < \frac{n}{2^{k-1}}$, which is $k = \lceil \log_2 \frac{n}{s} \rceil$. Hence $S(n, s) = sn \log_2 \frac{n}{s} + sn = O(sn \log \frac{n}{s})$. The estimate (3) is obtained by applying this generator to the upper and lower triangular parts of the (r_L, r_U) -quasiseparable matrix.

This first generator does not take fully advantage of the rank structure of the matrix: the representation of each anti-diagonal block is independent from the pivots found in the

¹Note that the statement of $O(n(r_L + r_U))$ for the same generator in [6] is erroneous.

block A_1 . The second generator, that will be presented in the next section addresses this issue, in order to remove the logarithmic factors in the estimate (3).

4.2 The Bruhat generator

We propose an alternative generator inspired by the generalized Bruhat decomposition [8, 7, 4]. 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

$$\mathcal{L} = \text{Left}(P \begin{bmatrix} L & 0 \end{bmatrix} Q), \quad (4)$$

$$\mathcal{E} = \text{Left}(E), \quad (5)$$

$$\mathcal{U} = \text{Left}(P \begin{bmatrix} U \\ 0 \end{bmatrix} Q). \quad (6)$$

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

LEMMA 4. $A = \text{Left}(\mathcal{L}\mathcal{E}^T\mathcal{U})$

PROOF.

$$A = P \begin{bmatrix} L & 0_{m \times (n-r)} \end{bmatrix} Q Q^T \begin{bmatrix} U \\ 0_{(n-r) \times n} \end{bmatrix} Q.$$

From Theorem 1, the matrix $Q^T \begin{bmatrix} U \\ 0 \end{bmatrix} Q$ is upper triangular and the matrix $P \begin{bmatrix} L & 0 \end{bmatrix} P^T$ is lower triangular. Applying Lemma 1 yields

$$A = \text{Left}(A) = \text{Left}(\mathcal{L}Q^T \begin{bmatrix} U \\ 0 \end{bmatrix} Q) = \text{Left}(\mathcal{L}E^T P \begin{bmatrix} U \\ 0 \end{bmatrix} Q),$$

where $E = P \begin{bmatrix} L & 0 \\ & 0 \end{bmatrix} Q$. Then, as $\mathcal{L}E^T$ is the matrix $P \begin{bmatrix} L & 0 \end{bmatrix} P^T$ with some coefficients zeroed out, it is lower triangular, hence applying again Lemma 2 yields

$$A = \text{Left}(\mathcal{L}E^T\mathcal{U}). \quad (7)$$

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

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 $(\mathcal{L}, \mathcal{E}, \mathcal{U})$ uses $2s(n-s)$ field coefficients and $O(n)$ additional indices.

PROOF. The leading column elements of \mathcal{L} are located at the pivot positions of the left triangular rank profile matrix \mathcal{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 \mathcal{U} . \square

Figure 1 illustrates this generator on a left triangular matrix of quasiseparable order 5. As the support of \mathcal{L} and \mathcal{U}

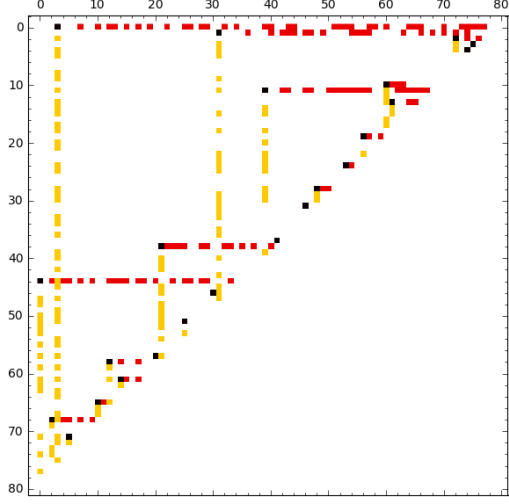


Figure 1: Support of the \mathcal{L} (yellow), \mathcal{E} (black) and \mathcal{U} (red) matrices of the Bruhat generator for a 80×80 left triangular matrix of quasiseparable order 5.

is disjoint, the two matrices can be shown on the same left triangular matrix. The pivots of \mathcal{E} (black) are the leading coefficients of every non-zero row of \mathcal{U} and non-zero column of \mathcal{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 \mathcal{L} factor of the Bruhat generator.

First remark that there exists a permutation matrix \mathcal{Q} which moves the non-zero columns of \mathcal{L} in the first r column positions, sorted by increasing leading row index, i.e. such that $\mathcal{L}\mathcal{Q}$ is in column echelon form. The matrix $\mathcal{L}\mathcal{Q}$ is now denser, but still has $r = \text{rank}(A)$ columns, which may exceed s and therefore prevent to compute within complexities bounded by a function 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 :

$$\begin{bmatrix} D_1 & & & & & \\ S_2 & D_2 & & & & \\ & S_3 & D_3 & & & \\ & & & \ddots & \ddots & \\ & & & & & S_t & D_t \end{bmatrix}.$$

Algorithm 3 Compressing the Bruhat generator

Input: \mathcal{L} : the first matrix of the Bruhat generator

Output: D, S, T, \mathcal{Q} : the compression of \mathcal{L}

1: $\mathcal{Q} \leftarrow$ a permutation s.t. $\mathcal{L}\mathcal{Q}$ is in column echelon form

2: $C \leftarrow \mathcal{L}\mathcal{Q} \begin{bmatrix} I_r \\ 0 \end{bmatrix}$ where $r = \text{rank}(\mathcal{L})$

3: Split C in column slices of width s .

4:
$$\triangleright C = \begin{bmatrix} C_{11} & & & & \\ C_{21} & C_{22} & & & \\ \vdots & \vdots & \ddots & & \\ C_{t1} & C_{t2} & \dots & C_{tt} \end{bmatrix}$$
 where C_{ii} is $k_i \times s$.

5: $D \leftarrow \text{Diag}(C_{11}, \dots, C_{tt})$

6: $C \leftarrow C - D = \begin{bmatrix} 0 & & & & \\ C_{21} & & & & \\ \vdots & \ddots & \ddots & & \\ C_{t1} & \dots & C_{t,t-1} & 0 \end{bmatrix}$

7: $T \leftarrow I_n$

8: **for** $i = 3 \dots t$ **do**

9: **for** each non zero column j of $\begin{bmatrix} C_{i,i-2} \\ \vdots \\ C_{t,i-2} \end{bmatrix}$ **do**

10: Let k be a zero column of $\begin{bmatrix} C_{i,i-1} \\ \vdots \\ C_{t,i-1} \end{bmatrix}$.

11: Move col. j in $\begin{bmatrix} C_{i,i-2} \\ \vdots \\ C_{t,i-2} \end{bmatrix}$ to col. k in $\begin{bmatrix} C_{i,i-1} \\ \vdots \\ C_{t,i-1} \end{bmatrix}$.

12: $T \leftarrow (I_n + \Delta^{(k,j)} - \Delta^{(k,k)}) \times T$

13: **end for**

14: **end for**

15: $S \leftarrow C = \begin{bmatrix} 0 & & & & \\ C_{21} & & & & \\ \vdots & \ddots & \ddots & & \\ C_{t1} & \dots & C_{t,t-1} & 0 \end{bmatrix}$

16: **Return** (D, S, T, \mathcal{Q})

LEMMA 6. Algorithm 3 computes a tuple (D, S, T, \mathcal{Q}) where \mathcal{Q} is a permutation matrix putting \mathcal{L} in column echelon form, T is an $r \times r$ $\{0, 1\}$ -matrix, $D = \text{Diag}(D_1, \dots, D_t)$, $S = \begin{bmatrix} 0 \\ S_2 & 0 \\ & \ddots & \ddots \\ & & & S_t \end{bmatrix}$ 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 \mathcal{L} and satisfies $\mathcal{L} = [D + ST \quad 0_{n \times (n-r)}] \mathcal{Q}^T$.

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 \mathcal{L} are column segments below a pivot and above the anti-diagonal. From Lemma 5, these segments have the property that each row of \mathcal{L} is intersected by no more than s of them. This property is preserved by

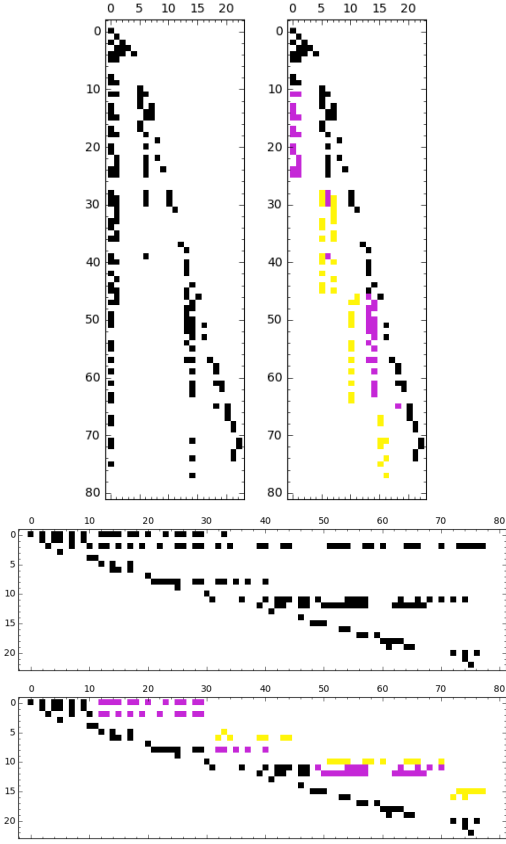


Figure 2: Support of the matrices \mathcal{LQ} (left), \mathcal{PU} (center) and of the corresponding compact Bruhat generator (right and bottom) for the matrix of Figure 1. In the compact Bruhat generator: D is in black and S in magenta and yellow; those rows and columns moved at step 11 of Algorithm 3 are in yellow.

column permutation, and still holds on the matrix C . In the first row of $[C_{i1} \dots C_{ii}]$, there is a pivot located in the block C_{ii} . Hence there is at most $s - 1$ such segments intersecting $[C_{i1} \dots C_{i,i-1}]$. These $s - 1$ columns can all be gathered in the block $C_{i,i-1}$ of column dimension s .

There only remains to show that ST is the matrix C of step 6. For every pair of indices (j, k) selected in loop 8, right multiplication by $(I_n + \Delta^{(k,j)} - \Delta^{(k,k)})$ adds up column k to column j and zeroes out column k . On matrix S , this has the effect of reverting each operations done at step 11 in the reverse order of the loop 8. \square

A compact representation of U is obtained in Lemma 7 by running Algorithm 3 on U^T and transposing its output.

LEMMA 7. *There exist a tuple (D, S, T, Q) where Q is a permutation matrix putting U in row echelon form, T is an $r \times r$ $\{0, 1\}$ -matrix, $D = \text{Diag}(D_1, \dots, D_t)$, $S = \begin{bmatrix} 0 & S_2 & & \\ & \ddots & \ddots & \\ & & 0 & S_t \end{bmatrix}$ where each D_i and S_i is $s \times k_i$ for $k_i \geq s$ and $\sum_{i=1}^t k_i = n$. This tuple is the compact Bruhat generator for U . These matrices satisfy $U = \mathcal{P}^T \begin{bmatrix} D + TS \\ 0_{(n-r) \times n} \end{bmatrix}$.*

According to (7), the reconstruction of the initial matrix A , from the compact Bruhat generators, writes

$$A = (D_{\mathcal{L}} + S_{\mathcal{L}}T_{\mathcal{L}})R(D_{\mathcal{U}} + T_{\mathcal{U}}S_{\mathcal{U}}) \quad (8)$$

where R is the leading $r \times r$ sub-matrix of $Q^T \mathcal{E}^T \mathcal{P}^T$. As it has full rank, it is a permutation matrix.

This factorization is a compact version of the generalized Bruhat decomposition [8, 4]: the left factor is a column echelon form, the right factor a row echelon form.

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 $T_1(n, s) = K_{\omega} s^{\omega-2} \left(\frac{n}{2}\right)^2 + 2T_1(n/2, s)$, which solves in

$$T(n, s) = \frac{K_{\omega}}{2} s^{\omega-2} n^2 \quad \text{with} \quad K_{\omega} = \frac{2^{\omega-2}}{(2^{\omega} - 2)(2^{\omega-2} - 1)} C_{\omega}$$

where C_{ω} is the leading constant of the complexity of matrix multiplication [2].

5.1.2 The Bruhat generator

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

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{E} 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 17 and 18, that is

$$\begin{aligned} \mathcal{U}_2 &= \text{Left}(P_2 \begin{bmatrix} U_2 \\ 0 \end{bmatrix} Q_2), & \mathcal{U}_3 &= \text{Left}(P_3 \begin{bmatrix} U_3 \\ 0 \end{bmatrix} Q_3), \\ \mathcal{L}_2 &= \text{Left}(P_2 \begin{bmatrix} L_2 & 0 \end{bmatrix} Q_2), & \mathcal{L}_3 &= \text{Left}(P_3 \begin{bmatrix} L_3 & 0 \end{bmatrix} Q_3). \end{aligned}$$

At step 9, we have

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & * \end{bmatrix} = \begin{bmatrix} P_1 & & \\ & I_{\frac{n}{2}} & \\ & & \end{bmatrix} \left[\begin{array}{c|c} L_1 & \\ \hline M_1 & I_{\frac{n}{2}-r_1} \\ \hline E & 0 \end{array} \middle| \begin{array}{c} \\ \\ \hline I_{\frac{n}{2}} \end{array} \right] \times \begin{bmatrix} U_1 & V_1 & D \\ & 0 & F \\ \hline & & G \end{bmatrix} \begin{bmatrix} Q_1 & \\ & I_{\frac{n}{2}} \end{bmatrix}$$

As the first r_1 rows of $P_1^T H$ are zeros, there exists \bar{P}_2 a permutation matrix and \bar{L}_2 , a lower triangular matrix, such that $P_1^T P_2 L_2 = \begin{bmatrix} 0_{r_1 \times \frac{n}{2}} \\ \bar{P}_2 \bar{L}_2 \end{bmatrix}$. Similarly, there exist \bar{Q}_3 , a permutation matrix and \bar{U}_3 , an upper triangular matrix,

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**
 - 2: **Return** $([0], [0], [0])$
 - 3: **end if**
 - 4: Split $A = \begin{bmatrix} A_1 & A_2 \\ A_3 & * \end{bmatrix}$ where A_3 is $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$
 - 5: Decompose $A_1 = P_1 \begin{bmatrix} L_1 \\ M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1$ \triangleright PLUQ(A_1)
 - 6: $R_1 \leftarrow P_1 \begin{bmatrix} I_{r_1} \\ 0 \end{bmatrix} Q_1$ where $r_1 = \text{rank}(A_1)$.
 - 7: $\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \leftarrow P_1^T A_2$ \triangleright PermR(A_2, P_1^T)
 - 8: $\begin{bmatrix} C_1 & C_2 \end{bmatrix} \leftarrow A_3 Q_1^T$ \triangleright PermC(A_3, Q_1^T)
 - 9: Here $A = \left[\begin{array}{cc|c} L_1 \setminus U_1 & V_1 & B_1 \\ M_1 & 0 & B_2 \\ \hline C_1 & C_2 & \end{array} \right]$.
 - 10: $D \leftarrow L_1^{-1} B_1$ \triangleright TRSM(L_1, B_1)
 - 11: $E \leftarrow C_1 U_1^{-1}$ \triangleright TRSM(C_1, U_1)
 - 12: $F \leftarrow B_2 - M_1 D$ \triangleright MM(B_2, M_1, D)
 - 13: $G \leftarrow C_2 - E V_1$ \triangleright MM(C_2, E, V_1)
 - 14: Here $A = \left[\begin{array}{cc|c} L_1 \setminus U_1 & V_1 & D \\ M_1 & 0 & F \\ \hline E & G & \end{array} \right]$.
 - 15: $H \leftarrow P_1 \begin{bmatrix} 0_{r_1 \times \frac{n}{2}} \\ F \end{bmatrix}$
 - 16: $I \leftarrow \begin{bmatrix} 0_{r_1 \times \frac{n}{2}} & G \end{bmatrix} Q_1$
 - 17: $(\mathcal{L}_2, \mathcal{E}_2, \mathcal{U}_2) \leftarrow \text{LT-Bruhat}(H)$
 - 18: $(\mathcal{L}_3, \mathcal{E}_3, \mathcal{U}_3) \leftarrow \text{LT-Bruhat}(I)$
 - 19: $\mathcal{L} \leftarrow \text{Left} \left(\begin{bmatrix} P_1 & & \\ & I_{\frac{n}{2}} & \\ & & \begin{bmatrix} L_1 & \\ M_1 & 0 \end{bmatrix} \begin{bmatrix} Q_1 & \\ & I_{\frac{n}{2}} \end{bmatrix} \end{bmatrix} \right) + \begin{bmatrix} 0 & \mathcal{L}_2 \\ \mathcal{L}_3 & \end{bmatrix}$
 - 20: $\mathcal{U} \leftarrow \begin{bmatrix} P_1 \begin{bmatrix} U_1 & V_1 \\ 0 & 0 \end{bmatrix} Q_1 & \text{Left}(P_1 \begin{bmatrix} D \\ 0 \end{bmatrix}) \\ & 0 \end{bmatrix} + \begin{bmatrix} 0 & \mathcal{U}_2 \\ \mathcal{U}_3 & \end{bmatrix}$
 - 21: $\mathcal{E} \leftarrow \begin{bmatrix} \mathcal{E}_1 & \mathcal{E}_2 \\ \mathcal{E}_3 & \end{bmatrix}$
 - 22: **Return** $(\mathcal{L}, \mathcal{E}, \mathcal{U})$
-

such that $U_3 Q_3 Q_1^T = [0_{\frac{n}{2} \times r_1} \quad \bar{U}_3 \bar{Q}_3]$. Hence

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & * \end{bmatrix} = \begin{bmatrix} P_1 & \\ & P_3 \end{bmatrix} \left[\begin{array}{cc|c} L_1 & & \\ M_1 & \bar{P}_2 \bar{L}_2 & \\ \hline \bar{P}_3^T E & 0 & L_3 \end{array} \right] \times \begin{bmatrix} U_1 & V_1 & D Q_2^T \\ & 0 & U_2 \\ \hline \bar{U}_3 \bar{Q}_3 & & \end{bmatrix} \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix}$$

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

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & * \end{bmatrix} = \begin{bmatrix} P_1 \begin{bmatrix} I_{r_1} & \\ & \bar{P}_2 \end{bmatrix} & \\ & P_3 \end{bmatrix} \left[\begin{array}{cc|c} L_1 & & \\ N_1 & \bar{L}_2 & \\ \hline E & 0 & L_3 \end{array} \right] \times \begin{bmatrix} U_1 & W_1 & D Q_2^T \\ & 0 & U_2 \\ \hline \bar{U}_3 & & \end{bmatrix} \begin{bmatrix} \begin{bmatrix} I_{r_1} & \\ & \bar{Q}_3 \end{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix}.$$

A PLUQ of $\begin{bmatrix} A_1 & A_2 \\ A_3 & * \end{bmatrix}$ 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 [2, Algorithm 1].

Finally,

$$P \begin{bmatrix} U \\ 0 \end{bmatrix} Q = \begin{bmatrix} P_1 & \\ & I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} U_1 & V_1 & D \\ & 0 & \bar{P}_2 U_2 Q_2 \\ 0 & P_3 \bar{U}_3 \bar{Q}_3 & 0 \end{bmatrix} \begin{bmatrix} Q_1 & \\ & I_{\frac{n}{2}} \end{bmatrix} \\ = \begin{bmatrix} P_1 \begin{bmatrix} U_1 & V_1 \\ 0 & 0 \end{bmatrix} Q_1 & P_1 \begin{bmatrix} D \\ 0 \end{bmatrix} \\ & 0 \end{bmatrix} + \begin{bmatrix} & P_2 \begin{bmatrix} U_2 \\ 0 \end{bmatrix} Q_2 \\ P_3 \begin{bmatrix} U_3 \\ 0 \end{bmatrix} Q_3 \end{bmatrix}.$$

Hence

$$\text{Left}(PUQ) = \begin{bmatrix} P_1 \begin{bmatrix} U_1 & V_1 \\ 0 & 0 \end{bmatrix} Q_1 & \text{Left}(P_1 \begin{bmatrix} D \\ 0 \end{bmatrix}) \\ & 0 \end{bmatrix} + \begin{bmatrix} & \mathcal{U}_2 \\ \mathcal{U}_3 & \end{bmatrix}.$$

The complexity analysis is exactly that of Theorem 2. \square

The computation of a compact Bruhat generator is obtained by combining Algorithm 4 with Algorithm 3.

5.2 Applying a vector

For the three generators proposed earlier, the application of a vector to the corresponding left triangular matrix takes the same amount of field operations as the number of coefficients used for its representation. This yields a cost of $O(n(r_L \log \frac{n}{r_L} + r_U \log \frac{n}{r_U}))$ field operations for multiplying a vector to an (r_L, r_U) -quasiseparable matrix using the binary tree PLUQ generator and $O(n(r_L + r_U))$ using either one of the Bruhat generator or its compact variant.

5.3 Multiplying two left-triangular matrices

5.3.1 The binary tree PLUQ generator

Let $T_{RL}(n, s)$ denote the cost of multiplying a dense $s \times n$ matrix by a left triangular quasiseparable matrix of order s . The natural divide and conquer algorithm yields the recurrence formula:

$$T_{RL}(n, s) = 2T_{RL}(n/2, s) + O(ns^{\omega-1}) = O(ns^{\omega-1} \log \frac{n}{s}).$$

Let $T_{PL}(n, s)$ denote the cost of multiplying a PLUQ decomposition of dimension n and rank $s \leq n/2$ with a left triangular quasiseparable matrix of order s . The product can be done in

$$T_{PL}(n, s) = T_{RL}(n, s) + O(n^2 s^{\omega-2}) = O(n^2 s^{\omega-2}).$$

Lastly, let $T_{LL}(n, s)$ denote the cost of multiplying two left-triangular matrices of quasiseparability order s . Again the natural recursive algorithm yields:

$$T_{LL}(n, s) = 2T_{LL}(n/2, s) + 2T_{PL}(n/2, s) + O(n^2 s^{\omega-2}) \\ = O(n^2 s^{\omega-2})$$

5.3.2 The Bruhat generator

Using the decomposition (8), the product of two left triangular matrices writes $A \times B = C_A R_A E_A \times C_B R_B E_B$ where $C_X = D_{\mathcal{L}_X} + S_{\mathcal{L}_X} T_{\mathcal{L}_X}$ and $E_X = D_{\mathcal{U}_X} + T_{\mathcal{U}_X} S_{\mathcal{U}_X}$ for $X \in \{A, B\}$. We will compute this product using the following parenthesizing:

$$A \times B = C_A (R_A (E_A \times C_B) R_B) E_B. \quad (9)$$

The product $E_A \times C_B = (D_{\mathcal{U}_A} + T_{\mathcal{U}_A} S_{\mathcal{U}_A})(D_{\mathcal{L}_B} + S_{\mathcal{L}_B} T_{\mathcal{L}_B})$ only consists in multiplying together block diagonal or sub-diagonal matrices $n \times r_B$ or $r_A \times n$. We will describe the

product of two block diagonal matrices (flat times tall); the other cases with sub-diagonal matrices work similarly.

Each term to be multiplied is decomposed in a grid of $s \times s$ tiles (except at the last row and column positions). In this grid, the non-zero blocks are non longer in a block-diagonal layout: in a flat matrix, the leading block of a block row may lie at the same block column position as the trailing block of its preceding block row, as shown in Figure 3. However, since $k_i \geq s$ for all i , no more than two

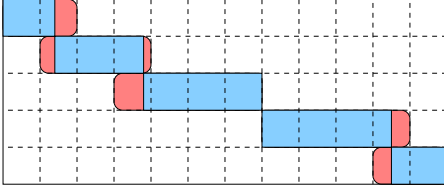


Figure 3: Aligning a block diagonal matrix (blue) on an $s \times s$ grid. Each block row of the aligned structure (red) may overlap with the previous and next block rows on at most one $s \times s$ tile on each side.

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 and upper triangular matrices, which in turn translate into products of left triangular matrices and J_n as shows in Table 1.

\times	Lower	Upper
Lower	$J_n \times \text{Left} \times J_n \times \text{Left}$	$J_n \times \text{Left} \times \text{Left} \times J_n$
Upper	$\text{Left} \times J_n \times J_n \times \text{Left}$	$\text{Left} \times J_n \times \text{Left} \times J_n$

Table 1: Converting products with lower and upper triangular matrices into products with left triangular matrices.

The complexity of section 5.3 directly applies for the computation of Upper \times Lower and Lower \times Upper products. For the remaining products, a J_n factor has to be added between two left triangular matrices, hence between the E_A and C_B factors in the innermost product of (9). Reverting the row order of C_B does not impact the complexity of computing this product, hence the same complexity applies here too.

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 proposed algorithms for multiplication of two quasiseparable matrices output a dense $n \times n$ matrix in time $O(n^2(l_A^{\omega-2} + u_A^{\omega-2} + l_B^{\omega-2} + u_B^{\omega-2}))$. However, the product is also a quasiseparable matrix, of order $(l_A + l_B, u_A + u_B)$ [5, 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 natural question is therefore to find an algorithm computing this representation from the generators of A and B in time $O(n(l_A^{\omega-1} + u_A^{\omega-1} + l_B^{\omega-1} + u_B^{\omega-1}))$.

Acknowledgment

We are thankful to Paola Boito for having introduced us to the notion of rank structured matrices and quasiseparability.

References

- [1] P. Boito, Y. Eidelman, and L. Gemignani. “Implicit QR for companion-like pencils”. In: *Mathematics of Computation* (2015). DOI: [10.1090/mcom/3020](https://doi.org/10.1090/mcom/3020).
- [2] J.-G. Dumas, C. Pernet, and Z. Sultan. “Simultaneous computation of the row and column rank profiles”. In: *Proc. ISSAC’13*. Ed. by M. Kauers. ACM Press, 2013, pp. 181–188. DOI: [10.1145/2465506.2465517](https://doi.org/10.1145/2465506.2465517).
- [3] J.-G. Dumas, C. Pernet, and Z. Sultan. “Computing the Rank Profile Matrix”. In: *Proc. ISSAC’15*. Bath, United Kingdom: ACM, 2015, pp. 149–156. DOI: [10.1145/2755996.2756682](https://doi.org/10.1145/2755996.2756682).
- [4] J.-G. Dumas, C. Pernet, and Z. Sultan. *Fast Computation of the Rank Profile Matrix and the Generalized Bruhat Decomposition*. Tech. rep. in submission. [arXiv:1601.01798 \[cs.SC\]](https://arxiv.org/abs/1601.01798). 2015.
- [5] Y. Eidelman and I. Gohberg. “On a new class of structured matrices”. In: *Integral Equations and Operator Theory* 34.3 (Sept. 1999), pp. 293–324. DOI: [10.1007/BF01300581](https://doi.org/10.1007/BF01300581).
- [6] Y. Eidelman, I. Gohberg, and V. Olshevsky. “The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order”. In: *Linear Algebra and its Applications* 404 (2005), pp. 305–324. DOI: [10.1016/j.laa.2005.02.037](https://doi.org/10.1016/j.laa.2005.02.037).
- [7] G. I. Malaschonok. “Fast generalized Bruhat decomposition”. In: *CASC’10*. Vol. 6244. LNCS. Tsakhkadzor, Armenia: Springer-Verlag, 2010, pp. 194–202. DOI: [10.1007/978-3-642-15274-0_16](https://doi.org/10.1007/978-3-642-15274-0_16).
- [8] W. Manthey and U. Helmke. “Bruhat canonical form for linear systems”. In: *Linear Algebra and its Applications* 425.2–3 (2007). Special Issue in honor of Paul Fuhrmann, pp. 261–282. DOI: [10.1016/j.laa.2007.01.022](https://doi.org/10.1016/j.laa.2007.01.022).
- [9] R. Vandebril, M. V. Barel, G. Golub, and N. Mastronardi. “A bibliography on semiseparable matrices*”. In: *CALCOLO* 42.3 (2005), pp. 249–270. DOI: [10.1007/s10092-005-0107-z](https://doi.org/10.1007/s10092-005-0107-z).
- [10] R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix computations and semiseparable matrices: linear systems*. Vol. 1. The Johns Hopkins University Press, 2007.