

Fast Computation of the Rank Profile Matrix and the Generalized Bruhat Decomposition

Jean-Guillaume Dumas, Clement Pernet, Ziad Sultan

► **To cite this version:**

Jean-Guillaume Dumas, Clement Pernet, Ziad Sultan. Fast Computation of the Rank Profile Matrix and the Generalized Bruhat Decomposition. Journal of Symbolic Computation, Elsevier, 2017, Special issue on ISSAC'15, 83, pp.187-210. 10.1016/j.jsc.2016.11.011 . hal-01251223v2

HAL Id: hal-01251223

<https://hal.archives-ouvertes.fr/hal-01251223v2>

Submitted on 14 May 2018

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.

Fast Computation of the Rank Profile Matrix and the Generalized Bruhat Decomposition

Jean-Guillaume Dumas

Université Grenoble Alpes, Laboratoire LJK, umr CNRS, BP53X, 51, av. des Mathématiques, F38041 Grenoble, France

Clément Pernet

Université Grenoble Alpes, Laboratoire de l'Informatique du Parallélisme, Université de Lyon, France.

Ziad Sultan

Université Grenoble Alpes, Laboratoire LJK and LIG, Inria, CNRS, Inovallée, 655, av. de l'Europe, F38334 St Ismier Cedex, France

Abstract

The row (resp. column) rank profile of a matrix describes the stair-case shape of its row (resp. column) echelon form. We describe a new matrix invariant, the rank profile matrix, summarizing all information on the row and column rank profiles of all the leading sub-matrices. We show that this normal form exists and is unique over a field but also over any principal ideal domain and finite chain ring. We then explore the conditions for a Gaussian elimination algorithm to compute all or part of this invariant, through the corresponding PLUQ decomposition. This enlarges the set of known elimination variants that compute row or column rank profiles. As a consequence a new Crout base case variant significantly improves the practical efficiency of previously known implementations over a finite field. With matrices of very small rank, we also generalize the techniques of Storjohann and Yang to the computation of the rank profile matrix, achieving an $(r^\omega + mn)^{1+o(1)}$ time complexity for an $m \times n$ matrix of rank r , where ω is the exponent of matrix multiplication. Finally, we give connections to the Bruhat decomposition, and several of its variants and generalizations. Consequently, the algorithmic improvements made for the PLUQ factorization, and their implementation, directly apply to these decompositions. In particular, we show how a PLUQ decomposition revealing the rank profile matrix also reveals both a row and a column echelon form of the input matrix or of any of its leading sub-matrices, by a simple post-processing made of row and column permutations.

Key words: Gaussian elimination, Rank profile, Echelon form, PLUQ decomposition, Bruhat decomposition, McCoy's rank, Finite Chain ring.

Contents

1	Introduction	2
2	The rank profile matrix	5
2.1	Definition over a field	5
2.2	Generalization over a commutative ring with unity	7
2.2.1	Over a principal ideal domain	8
2.2.2	Over a finite chain ring	8
3	When does a PLUQ algorithm reveal the rank profile matrix?	9
3.1	Ingredients of a PLUQ decomposition algorithm	9
3.1.1	Pivot search	10
3.1.2	Pivot permutation	10
3.2	How to reveal rank profiles	11
4	Algorithms for the rank profile matrix	14
4.1	Iterative algorithms	15
4.1.1	Row and Column order Search	15
4.1.2	Lexicographic order based pivot search	15
4.1.3	Product order based pivot search	16
4.2	Recursive algorithms	16
4.2.1	Slab recursive algorithms	16
4.2.2	Tile recursive algorithms	16
5	Improvements in practice	18
6	Relations with other triangularizations	19
6.1	The LEU decomposition	19
6.2	The Bruhat decomposition	21
6.3	Relation to LUP and PLU decompositions	22
6.4	Computing Echelon forms	23
6.5	The generalized Bruhat decomposition	24
7	Improvement for low rank matrices	25
7.1	Storjohann and Yang's algorithm	25
7.2	Online LU decomposition	27

1. Introduction

Triangular matrix decompositions are widely used in computational linear algebra. Besides solving linear systems of equations, they are also used to compute other objects more specific to exact arithmetic: computing the rank, sampling a vector from the null-space, computing echelon forms and rank profiles.

The *row rank profile* (resp. *column rank profile*) of an $m \times n$ matrix A with rank r , denoted by $\text{RowRP}(A)$ (resp. $\text{ColRP}(A)$), is the lexicographically smallest sequence of r indices of linearly independent rows (resp. columns) of A . An $m \times n$ matrix has generic row (resp. column) rank profile if its row (resp. column) rank profile is $(1, \dots, r)$. Lastly, an

* This research was partly supported by the HPAC project of the French Agence Nationale de la Recherche (ANR 11 BS02 013) and the OpenDreamKit Horizon 2020 European Research Infrastructures project (#676541) .

Email addresses: jean-guillaume.dumas@imag.fr (Jean-Guillaume Dumas),
clement.pernet@imag.fr (Clément Pernet), <mailto:ziad.sultan@imag.fr> (Ziad Sultan).

URLs: <http://www-ljk.imag.fr/membres/Jean-Guillaume.Dumas/> (Jean-Guillaume Dumas),
<http://lig-membres.imag.fr/pernet/> (Clément Pernet),
<http://moais.imag.fr/membres/ziad.sultan> (Ziad Sultan).

$m \times n$ matrix has generic rank profile if its r first leading principal minors are nonzero. Note that if a matrix has generic rank profile, then its row and column rank profiles are generic, but the converse is false: the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ does not have generic rank profile even if its row and column rank profiles are generic. The row support (resp. column support) of a matrix A , denoted by $\text{RowSupp}(A)$ (resp. $\text{ColSupp}(A)$), is the subset of indices of its nonzero rows (resp. columns).

We recall that the row echelon form of an $m \times n$ matrix A is an upper triangular matrix $E = TA$, for a nonsingular matrix T , with the zero rows of E at the bottom and the nonzero rows in stair-case shape: $\min\{j : a_{i,j} \neq 0\} < \min\{j : a_{i+1,j} \neq 0\}$. As T is nonsingular, the column rank profile of A is that of E , and therefore corresponds to the column indices of the leading elements in the staircase. Similarly the row rank profile of A is composed of the row indices of the leading elements in the staircase of the column echelon form of A .

Rank profiles and triangular matrix decompositions. The rank profiles of a matrix and the triangular matrix decompositions obtained by Gaussian elimination are strongly related. The elimination of matrices with arbitrary rank profiles gives rise to several matrix factorizations and many algorithmic variants. In numerical linear algebra one often uses the PLUQ decomposition, with P and Q permutation matrices, L a lower unit triangular matrix and U an upper triangular matrix. The LSP and LQUP variants of [Ibarra et al. \(1982\)](#) have been introduced to reduce the complexity of rank deficient Gaussian elimination to that of matrix multiplication. Many other algorithmic decompositions exist allowing fraction free computations [Jeffrey \(2010\)](#), in-place computations [Dumas et al. \(2008\)](#); [Jeannerod et al. \(2013\)](#) or sub-cubic rank-sensitive time complexity [Storjohann \(2000\)](#); [Jeannerod et al. \(2013\)](#). The reader may refer to [Jeannerod et al. \(2013\)](#) for a detailed comparison between these matrix factorizations, and further details on the CUP (resp. PLE) variants, revealing the row (resp. column) rank profiles. All these algorithms, together with the schoolbook Gaussian elimination algorithm share the property that, for a row rank profile computation, the pivot search processes rows in order, and searches a pivot in all possible column position before declaring the row linearly dependent with the previous ones. As a consequence, blocking is limited to only one dimension (in this case the row dimension) leading to slab algorithms [Klimkowski and van de Geijn \(1995\)](#) operating on rectangular blocks of unbalanced dimensions. This reduces the data locality of the algorithm, and therefore penalizes the efficiency of implementations in practice. In parallel, this blocking also puts more constraints on the dependencies between tasks [Dumas et al. \(2015a\)](#).

Contribution with respect to the state of the art. In [Dumas et al. \(2013\)](#) we proposed a first Gaussian elimination algorithm, with a recursive splitting of both row and column dimensions, which simultaneously computes the row and column rank profile while preserving the sub-cubic rank-sensitive time complexity and keeping the computation in-place. It showed that slab blocking is not a necessary condition for a Gaussian elimination to reveal rank profiles. Consequently, we have further analyzed the conditions on the pivoting that reveal the rank profiles in [Dumas et al. \(2015b\)](#), where we introduced a new matrix invariant, the rank profile matrix. This normal form contains the row and column rank profile information of the matrix and that of all its leading sub-matrices.

This normal form is closely related to a permutation matrix appearing in the Bruhat decomposition [Bruhat \(1956\)](#) and in related variants [Della Dora \(1973\)](#); [Grigoriev \(1981\)](#);

Bourbaki (2008); Malaschonok (2010); Manthey and Helmke (2007). Still, none of these did connect it to the notion of rank profile. In another setting, the construction of matrix Schubert varieties in (Miller and Sturmfels, 2005, Ch. 15) defines a similar invariant, but presents neither a matrix decomposition nor any computational aspects.

More precisely, the present paper gathers the key contributions of Dumas et al. (2013) and Dumas et al. (2015b):

- (1) we define a new matrix invariant over a field, the rank profile matrix, summarizing all information on the row and column rank profiles of all the leading sub-matrices;
- (2) we study the conditions for a Gaussian elimination algorithm to compute all or part of this invariant, through the corresponding PLUQ decomposition;
- (3) as a consequence, we show that the classical iterative CUP decomposition algorithm can actually be adapted to compute the rank profile matrix. Used, in a Crout variant, as a base-case to our recursive implementation over a finite field, it delivers a significant improvement in efficiency;
- (4) we also show that both the row and the column echelon forms of a matrix can be recovered from some PLUQ decompositions thanks to an elementary post-processing algorithm.

Further, we develop three novel aspects:

- (1) we study how the notion of rank profile matrix can be generalized over an arbitrary ring. We show that none of the common definition of rank over an arbitrary commutative ring allow to define a rank profile matrix in general. However, over a principal ideal domain and over a finite chain ring, we can produce such a definition.
- (2) we make further connections with existing matrix decompositions, in particular the Bruhat and the generalized Bruhat decompositions, exhibiting both the row and the column echelon in a single decomposition;
- (3) lastly, we extend the recent algorithmic improvements of Cheung et al. (2013); Storjohann and Yang (2014, 2015) for low rank matrices: indeed, we show here that the algorithm in Storjohann and Yang (2014) computes the rank profile matrix; propose an algorithmic variant thereof, reducing the leading constant by a factor of three; and lastly show an algorithmic reduction computing the rank profile matrix in time bounded by $(r^\omega + mn)^{1+o(1)}$ with a Las-Vegas probabilistic algorithm.

Organization of the article. We first introduce in Section 2 the rank profile matrix \mathcal{R}_A , and study in Section 2.2 its generalization over an arbitrary ring. We then study in Section 3 under which condition a PLUQ decomposition algorithms reveals the rank profile structure of a matrix. For this, we investigate existing and new pivoting strategies, based on all combination of elementary search and permutation operations, showing for each of them what part of the rank profile information is being computed. In particular we propose three new pivoting strategies that compute the rank profile matrix. As an illustration, we show in Section 4 how these pivoting strategies instantiate in iterative or recursive algorithms, using slab or tile blocking. Connections are made to the most common elimination algorithms and we state in full details the recent tile recursive algorithm of Dumas et al. (2013), implementing one of the new pivoting strategy. Section 5 shows how this better understanding on the pivoting strategies has resulted in the design of an iterative Crout CUP decomposition with rotations, to be used as a base case for the tile recursive algorithm, speeding up the computation efficiency, while still recovering the rank profile matrix invariant. We then show in Section 6 how a PLUQ decomposition

revealing the rank profile matrix relates with other triangular matrix decompositions, such as the LEU and the classical, modified or generalized Bruhat decompositions, or the computation of both row and column echelon forms, from a single PLUQ decomposition. Lastly, we extend in Section 7 the recent algorithms of [Storjohann and Yang \(2014, 2015\)](#) for the row or column rank profile of matrices with low rank to computed the rank profile matrix within the same complexities.

Notations. In the following, $0_{m \times n}$ denotes the $m \times n$ zero matrix. For two list of indices \mathcal{P} and \mathcal{Q} , $A_{\mathcal{P}, \mathcal{Q}}$ denotes the sub-matrix of A formed by the rows of index in \mathcal{P} and the columns of index in \mathcal{Q} . In particular, $A_{i..j, k..l}$ denotes the contiguous block of coefficients in A of rows position between i and j and columns position between k and l . We may denote by $*$ the sequence of all possible row or column indices: e.g. $A_{i, *}$ denotes the i -th row of A . To a permutation $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ we define the associated permutation matrix $P(\sigma)$, permuting rows by left multiplication: the rows of $P(\sigma)A$ are that of A permuted by σ . Reciprocally, for a permutation matrix P , we denote by $\sigma(P)$ the associated permutation.

2. The rank profile matrix

We propose in Theorem 3 the definition of the rank profile matrix, an invariant summarizing all information on the rank profiles of a matrix. As will be discussed in this section and in Section 6, this invariant is closely related to the Bruhat decomposition [Bruhat \(1956\)](#) and its generalizations [Grigoriev \(1981\)](#); [Tyrtysnikov \(1997\)](#); [Miller and Sturmfels \(2005\)](#).

2.1. Definition over a field

We first consider matrices over an arbitrary commutative field K .

Definition 1. An r -sub-permutation matrix is a matrix of rank r with only r non-zero entries equal to one.

Lemma 2. An $m \times n$ r -sub-permutation matrix has at most one non-zero entry per row and per column, and can be written $P \begin{bmatrix} I_r & \\ & 0_{(m-r) \times (n-r)} \end{bmatrix} Q$ where P and Q are permutation matrices.

Theorem 3. Let $A \in K^{m \times n}$ of rank r . There exists a unique $m \times n$ r -sub-permutation matrix \mathcal{R}_A of which every leading sub-matrix has the same rank as the corresponding leading sub-matrix of A . This sub-permutation matrix is called the rank profile matrix of A .

Proof. We prove existence by induction on the row dimension. of the leading submatrices.

If $A_{1,1..n} = 0_{1 \times n}$, setting $\mathcal{R}^{(1)} = 0_{1 \times n}$ satisfies the defining condition. Otherwise, let j be the index of the first invertible element in $A_{1,1..n}$ and set $\mathcal{R}^{(1)} = e_j^T$ the j -th n -dimensional canonical row vector, which satisfies the defining condition.

Now for a given $i \in \{1, \dots, m\}$, suppose that there is a unique $i \times n$ rank profile matrix $\mathcal{R}^{(i)}$ such that $\text{rank}(A_{1..\ell, 1..j}) = \text{rank}(\mathcal{R}_{1..\ell, 1..j})$ for every $j \in \{1..n\}$ and $\ell \in \{1..i\}$.

If $\text{rank}(A_{1..i+1,1..n}) = \text{rank}(A_{1..i,1..n})$, then $\mathcal{R}^{(i+1)} = \begin{bmatrix} \mathcal{R}^{(i)} \\ 0_{1 \times n} \end{bmatrix}$. Otherwise, consider k , the smallest column index such that $\text{rank}(A_{1..i+1,1..k}) = \text{rank}(A_{1..i,1..k}) + 1$ and set $\mathcal{R}^{(i+1)} = \begin{bmatrix} \mathcal{R}^{(i)} \\ e_k^T \end{bmatrix}$. Define $\begin{bmatrix} B & u \\ v^T & x \end{bmatrix} = A_{1..i+1,1..k}$, where u, v are vectors and x is a scalar. By definition of k , we have $\text{rank}(B) = \text{rank}(\begin{bmatrix} B \\ v^T \end{bmatrix})$.

First we show that $\mathcal{R}^{(i+1)}$ is an r_{i+1} -sub-permutation matrix. If not, then the k -th column of $\mathcal{R}^{(i)}$ would contain a 1 which, by induction hypothesis, would imply that $\text{rank}(\begin{bmatrix} B & u \\ v^T & x \end{bmatrix}) = \text{rank}(B) + 1$. Hence we would have $\text{rank}(\begin{bmatrix} B & u \\ v^T & x \end{bmatrix}) = \text{rank}(\begin{bmatrix} B & u \\ v^T & x \end{bmatrix}) + 1 = \text{rank}(B) + 2 = \text{rank}(\begin{bmatrix} B \\ v^T \end{bmatrix}) + 2$, a contradiction.

Then we show that any leading sub-matrix $A_{1..s,1..t}$ of A with $s \leq i+1, t \leq n$ has the same rank as the corresponding leading sub-matrix of $\mathcal{R}^{(i+1)}$. The case $s \leq i$ is covered by the induction; second, since $\text{rank}(B) = \text{rank}(\begin{bmatrix} B \\ v^T \end{bmatrix})$, any leading sub-matrix of $\begin{bmatrix} B \\ v^T \end{bmatrix}$ has the same rank as the corresponding sub-matrix of $\mathcal{R}^{(i+1)}$, which covers the case $t < k$. Lastly, for $s = i+1$ and $t \geq k$, the definition of k implies $\text{rank}(A_{1..s,1..t}) = \text{rank}(A_{1..i,1..t}) + 1 = \text{rank}(\mathcal{R}_{1..i,1..t}^{(i)}) + 1 = \text{rank}(\mathcal{R}_{1..i,1..t}^{(i+1)})$.

To prove uniqueness, suppose there exist two distinct rank profile matrices $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$ for a given matrix A and let (i, j) be the lexicographically minimal coordinates where $\mathcal{R}_{i,j}^{(1)} \neq \mathcal{R}_{i,j}^{(2)}$. The rank of the (i, j) -leading submatrices of $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$ differ but should both be equal to $\text{rank}(A_{1..i,1..j})$, a contradiction. \square

Example 4. $A = \begin{bmatrix} 2 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 2 & 0 & 1 \end{bmatrix}$ has $\mathcal{R}_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ for rank profile matrix over \mathbb{Q} .

Remark 5. The permutation matrix introduced in the *modified Bruhat decomposition* of [Tyrtysnikov \(1997\)](#), and defined there only for invertible matrices, is also the matrix E introduced in Malaschonok's LEU decomposition ([Malaschonok, 2010](#), Theorem 1). In the latter paper, an algorithm for this decomposition was only shown over a field for $m = n = 2^k$, and no connection was made to the relation with ranks and rank profiles. We have shown in ([Dumas et al., 2013](#), Corollary 1) that E is in fact the rank profile matrix and made the connection to the PLUQ decomposition explicit, as recalled in Section 6. We here generalize the existence to arbitrary rank r and dimensions m and n and after proving its uniqueness, we propose this definition as a new matrix normal form.

The rank profile matrix has the following properties:

Lemma 6. *Let A be a matrix.*

- (1) \mathcal{R}_A is diagonal if and only if A has generic rank profile.
- (2) \mathcal{R}_A is a permutation matrix if and only if A is invertible
- (3) $\text{RowRP}(A) = \text{RowSupp}(\mathcal{R}_A)$; $\text{ColRP}(A) = \text{ColSupp}(\mathcal{R}_A)$.

Moreover, for all $1 \leq i \leq m$ and $1 \leq j \leq n$, we have:

- (4) $\text{RowRP}(A_{1..i,1..j}) = \text{RowSupp}((\mathcal{R}_A)_{1..i,1..j})$
- (5) $\text{ColRP}(A_{1..i,1..j}) = \text{ColSupp}((\mathcal{R}_A)_{1..i,1..j})$,

These properties show how to recover the row and column rank profiles of A and of any of its leading sub-matrix.

2.2. Generalization over a commutative ring with unity

We now explore if and under which condition, the notion of rank profile matrix can be generalized over an arbitrary commutative ring with unity. As the rank profile matrix over a field relies on the notion of rank, which generalization over an arbitrary ring leads to several definitions, we review the most common of these definitions and explore their capacity to define a rank profile matrix invariant.

Let R be a commutative ring with unity. Following (Brown, 1992, §4), we will denote by $I_t(A)$ the t -th determinantal ideal, namely the ideal in R generated by all $t \times t$ minors of A , and set by definition $I_0 = R$. We have $R = I_0 \supseteq I_1(A) \supseteq I_2(A) \supseteq \cdots \supseteq (0)$. We will also denote by $\text{Ann}(S) = \{x \in R : \forall y \in S, xy = 0\}$ the annihilator ideal of a set S .

Definition 7. For an $m \times n$ matrix A over a commutative ring R , the rank of A can be defined as:

- (1) $r_{\text{Span}} = \min\{t : A = BC \text{ where } B \text{ is } m \times t \text{ and } C \text{ is } t \times n\}$, called spanning rank by Brown (1998) or Schein rank by (Bhaskara Rao, 2002, §2.4),
- (2) $r_{\text{McCoy}} = \max\{t : \text{Ann}(I_t(A)) = (0)\}$ introduced in (McCoy, 1948, Theorem 51),
- (3) $r_{\text{Det}} = \max\{t : A \text{ has a nonzero } t \times t \text{ minor}\}$ (Bhaskara Rao, 2002, §2.4) (this is also $\max\{t : I_t(A) \neq (0)\}$ defined in (Brown, 1992, §4, Ex. 11)),
- (4) $r_{\text{UnitMinor}} = \max\{t : A \text{ has a unit } t \times t \text{ minor}\}$ used in (Norton and Salagean, 2000, Def. 2.5).

We illustrate the value of these ranks in Table 1 for three 2×2 matrices (columns 3, 6, 9) and their leading 1×2 and 2×1 sub-matrices (columns 2, 5, 8). These ranks will be used to argue whether a rank profile matrix can be defined in each case (columns 4, 7 and 10 if it exists).

	over $\mathbb{Z}/4\mathbb{Z}$			over $\mathbb{Z}/6\mathbb{Z}$					
	$[0 \ 2]$, $\begin{bmatrix} 0 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}$	$\mathcal{R}_{\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}}$	$[2 \ 3]$, $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 & 3 \\ 3 & 0 \end{bmatrix}$	$\mathcal{R}_{\begin{bmatrix} 2 & 3 \\ 3 & 0 \end{bmatrix}}$	$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $[2 \ 3]$	$\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$	$\mathcal{R}_{\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}}$
r_{Span}	1	1	none	1	2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	1	2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
r_{McCoy}	0	1	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	1	1	none	1	2	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
r_{Det}	1	1	none	1	2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	1	2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$r_{\text{UnitMinor}}$	0	0	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	0	0	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	1,0	2	none

Table 1. Ranks of matrices according to the various definitions. These three 2×2 matrices are counterexamples showing that none of the four rank definitions of Definition 7 can be used to define a rank profile matrix in general.

For instance $r_{\text{Span}}(\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}) = 1$ since $\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} [2 \ 1]$ over $\mathbb{Z}/4\mathbb{Z}$. This matrix shows that r_{Span} can not be used to define a rank profile matrix. Indeed, if it would exist, its first row would be $[0 \ 1]$ (as the 1×1 leading submatrix has rank 0 and the 1×2 leading submatrix has rank 1). Similarly the first column of this rank profile matrix would be $[0 \ 1]^T$. But the rank of the permutation matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ would be $2 \neq r_{\text{Span}}(\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix})$. We will use the same reasoning pattern, to show that r_{Det} can not define a rank profile matrix for $\begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix}$, and neither does r_{McCoy} for $\begin{bmatrix} 2 & 3 \\ 3 & 0 \end{bmatrix}$ over $\mathbb{Z}/6\mathbb{Z}$: in these two cases, the rank profile matrix should be $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ to satisfy the rank condition on the 2×1 and 1×2 sub-matrices,

but the 2×2 rank is only 1. Lastly, $r_{\text{UnitMinor}}([2 \ 3]) = 0$ and $r_{\text{UnitMinor}}(\begin{bmatrix} 2 \\ 1 \end{bmatrix}) = 1$, therefore the rank profile matrix of $\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$ should be of the form $\begin{bmatrix} 0 & 0 \\ 1 & * \end{bmatrix}$, but it can then never be of rank 2, as the matrix $\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$.

Remark also that the rank profile matrix invariant is strongly connected with elimination as will be presented in the next sections. It therefore needs to be based on a notion of rank that is stable with multiplication by invertible matrices. This is the case with McCoy's rank (Brown, 1992, 4.11.(c)), but not with $r_{\text{UnitMinor}}$. Indeed the rank of $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$ is 1 whereas the rank of $\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$ is 0.

Consequently there is no notion of rank over an arbitrary commutative ring with unity supporting the definition of a rank profile matrix. We will now show that with additional assumptions on the ring, some of the above definitions for the rank coincide and make the existence of the rank profile matrix possible.

2.2.1. Over a principal ideal domain

Over a principal ideal domain, the existence of an underlying field of fractions guarantees that the McCoy rank, the spanning rank and the rank over the field of fractions coincide.

Corollary 8. *Let D be a principal ideal domain (PID) and let $A \in D^{m \times n}$ with McCoy's rank r . There exists a unique $m \times n$ r -sub-permutation matrix \mathcal{R}_A of which every leading sub-matrix has the same rank as the corresponding leading sub-matrix of A . This sub-permutation matrix is called the rank profile matrix of A .*

Proof. From (Brown, 1998, Proposition 1.6), over a PID with field of fractions K , $s_r = r_{\mathcal{M}} = \text{rank}_K$. Thus \mathcal{R}_A over K satisfies the requirements over D and is the unique such matrix. \square

2.2.2. Over a finite chain ring

A finite chain ring is a finite commutative ring with identity which ideals are ordered by inclusion (Clark and Liang, 1973). Equivalently, it is a finite local ring which maximal ideal is principal. These rings include $\mathbb{Z}/p^k\mathbb{Z}$ and $\text{GF}(p^k)$ for p prime, as well as all Galois rings of characteristic p^k .

Over a finite chain ring, McCoy rank and the UnitMinor rank coincide, and therefore allow to define a rank profile matrix, as shown next.

Lemma 9. *Over a finite chain ring, McCoy's rank is the largest positive integer $r \leq \{m, n\}$ such that there exist a unit $r \times r$ minor of A .*

Proof. Krull's theorem (Krull, 1938, Theorem 2) (see also (Hungerford, 1968, Proposition 4)) states that in a local ring, an element is a unit if and only if it is not in the maximal ideal. As the maximal ideal of the ring is principal, let g be a generator thereof. Then g is nilpotent. Indeed the ring is finite so there exists indices $i < j$ such that $g^i = g^j$. Therefore $g^i(1 - g^{j-i}) = 0$. But as g is not a unit, g^{j-i} is not a unit either, but then $1 - g^{j-i}$ must be a unit (otherwise the maximal ideal would contain 1 and would not be proper). Therefore $g^i = 0$. The nilpotency index of g is then the smallest positive integer ν such that $g^\nu = 0$.

Let $r = r_{\text{McCoy}}(A)$ and suppose that all $r \times r$ minors are non-unit. They belong to the maximal ideal and $I_r(A) = (g^i)$ for some $1 \leq i < \nu$. Therefore, $\text{Ann}(I_r(A)) = (g^{\nu-i})$, a contradiction. \square

Corollary 10 ((Norton and Salagean, 2000, Corollary 2.7)). *Over a finite chain ring R , with maximal ideal (g) , McCoy's rank is the rank over the field R/gR .*

Corollary 11. *Let R be a finite chain ring and let $A \in R^{m \times n}$ with McCoy's rank r . There exists a unique $m \times n$ r -sub-permutation matrix \mathcal{R}_A of which every leading sub-matrix has the same McCoy's rank as the corresponding leading sub-matrix of A . This sub-permutation matrix is called the rank profile matrix of A .*

Proof. From Corollary 10, we consider the residue field $K = R/gR$. Then \mathcal{R}_A over K satisfies the requirements over R and is the unique such matrix. \square

3. When does a PLUQ algorithm reveal the rank profile matrix?

From now on, for the sake of simplicity, we consider algorithms over a field.

3.1. Ingredients of a PLUQ decomposition algorithm

Over a field, the LU decomposition generalizes to matrices with arbitrary rank profiles, using row and column permutations (in some cases such as the CUP, or LSP decompositions, the row permutation is embedded in the structure of the C or S matrices). However such PLUQ decompositions are not unique and not all of them will necessarily reveal rank profiles and echelon forms. We will characterize the conditions for a PLUQ decomposition algorithm to reveal the row or column rank profile or the rank profile matrix.

We consider the four types of operation of a Gaussian elimination algorithm in the processing of the k -th pivot:

Pivot search: finding an element to be used as a pivot,

Pivot permutation: moving the pivot in diagonal position (k, k) by column and/or row permutations,

Update: applying the elimination at position (i, j) : $a_{i,j} \leftarrow a_{i,j} - a_{i,k}a_{k,k}^{-1}a_{k,j}$,

Normalization: dividing the k -th row (resp. column) by the pivot.

Choosing how each of these operation is done, and when they are scheduled results in an elimination algorithm. Conversely, any Gaussian elimination algorithm computing a PLUQ decomposition can be viewed as a set of specializations of each of these operations together with a scheduling.

The choice of doing the normalization on rows or columns only determines which of U or L will be unit triangular. The scheduling of the updates vary depending on the type of algorithm used: iterative, recursive, slab or tiled block splitting, with right-looking, left-looking or Crout variants (see Dongarra et al. (1998)). Neither the normalization nor the update impact the capacity to reveal rank profiles and we will thus now focus on the pivot search and permutation.

Choosing a search and a permutation strategy sets the matrices P and Q of the PLUQ decomposition obtained and, as we will see, determines the ability to recover information on the rank profiles. Once these matrices are fixed, the L and the U factors are unique. We therefore introduce the pivoting matrix.

Definition 12. The pivoting matrix of a PLUQ decomposition $A = PLUQ$ of rank r is the r -sub-permutation matrix

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

The r nonzero elements of $\Pi_{P,Q}$ are located at the initial positions of the pivots in the matrix A . Thus $\Pi_{P,Q}$ summarizes the choices made in the search and permutation operations.

3.1.1. Pivot search

The search operation vastly differs depending on the field of application. In numerical dense linear algebra, numerical stability is the main criterion for the selection of the pivot. In sparse linear algebra, the pivot is chosen so as to reduce the fill-in produced by the update operation. In order to reveal some information on the rank profiles, a notion of precedence has to be used: a usual way to compute the row rank profile is to search in a given row for a pivot and only move to the next row if the current row was found to be all zeros. This guarantees that each pivot will be on the first linearly independent row, and therefore the row support of $\Pi_{P,Q}$ will be the row rank profile. The precedence here is that the pivot's coordinates must minimize the order for the first coordinate (the row index). As a generalization, we consider the most common preorders of the cartesian product $\{1, \dots, m\} \times \{1, \dots, n\}$ inherited from the natural orders of each of its components and describe the corresponding search strategies, minimizing this preorder:

Row order: $(i_1, j_1) \preceq_{\text{row}} (i_2, j_2)$ iff $i_1 \leq i_2$: search for any invertible element in the first nonzero row.

Column order: $(i_1, j_1) \preceq_{\text{col}} (i_2, j_2)$ iff $j_1 \leq j_2$. search for any invertible element in the first nonzero column.

Lexicographic order: $(i_1, j_1) \preceq_{\text{lex}} (i_2, j_2)$ iff $i_1 < i_2$ or $i_1 = i_2$ and $j_1 \leq j_2$: search for the leftmost nonzero element of the first nonzero row.

Reverse lexicographic order: $(i_1, j_1) \preceq_{\text{revlex}} (i_2, j_2)$ iff $j_1 < j_2$ or $j_1 = j_2$ and $i_1 \leq i_2$: search for the topmost nonzero element of the first nonzero column.

Product order: $(i_1, j_1) \preceq_{\text{prod}} (i_2, j_2)$ iff $i_1 \leq i_2$ and $j_1 \leq j_2$: search for any nonzero element at position (i, j) being the only nonzero of the leading (i, j) sub-matrix.

Example 13. Consider the matrix $\begin{bmatrix} 0 & 0 & 0 & a & b \\ 0 & c & d & e & f \\ g & h & i & j & k \\ l & m & n & o & p \end{bmatrix}$, where each literal is a nonzero element.

The nonzero elements minimizing each preorder are the following:

Row order	a, b	Column order	g, l
Lexicographic order	a	Reverse lexic. order	g
Product order	a, c, g		

3.1.2. Pivot permutation

The pivot permutation moves a pivot from its initial position to the leading diagonal. Besides this constraint all possible choices are left for the remaining values of the permutation. Most often, it is done by row or column transpositions, as it clearly involves a small amount of data movement. However, these transpositions can break the precedence relations in the set of rows or columns, and can therefore prevent the recovery of the rank profile information. A pivot permutation that leaves the precedence relations unchanged will be called k -monotonically increasing.

Definition 14. A permutation of $\sigma \in \mathcal{S}_n$ is called k -monotonically increasing if its last $n - k$ values form a monotonically increasing sequence.

In particular, the last $n - k$ rows of the associated row-permutation matrix P_σ are in row echelon form. For example, the cyclic shift between indices k and i , with $k < i$ defined as $R_{k,i} = (1, \dots, k-1, i, k, k+1, \dots, i-1, i+1, \dots, n)$, that we will call a (k, i) -rotation, is an elementary k -monotonically increasing permutation.

Example 15. The $(1, 4)$ -rotation $R_{1,4} = (4, 1, 2, 3)$ is a 1-monotonically increasing permutation. Its row permutation matrix is $\begin{bmatrix} 0 & & & 1 \\ 1 & & & \\ & 1 & & \\ & & 1 & 0 \end{bmatrix}$.

Monotonically increasing permutations can be composed as stated in Lemma 16.

Lemma 16. *If $\sigma_1 \in \mathcal{S}_n$ is a k_1 -monotonically increasing permutation and $\sigma_2 \in \mathcal{S}_{k_1} \times \mathcal{S}_{n-k_1}$ a k_2 -monotonically increasing permutation with $k_1 < k_2$ then the permutation $\sigma_2 \circ \sigma_1$ is a k_2 -monotonically increasing permutation.*

Proof. The last $n - k_2$ values of $\sigma_2 \circ \sigma_1$ are the image of a sub-sequence of $n - k_2$ values from the last $n - k_1$ values of σ_1 through the monotonically increasing function σ_2 . \square

Therefore an iterative algorithm, using rotations as elementary pivot permutations, maintains the property that the permutation matrices P and Q at any step k are k -monotonically increasing. A similar property also applies with recursive algorithms.

3.2. How to reveal rank profiles

A PLUQ decomposition reveals the row (resp. column) rank profile if it can be read from the first r values of the permutation matrix P (resp. Q). Equivalently, by Lemma 6, this means that the row (resp. column) support of the pivoting matrix $\Pi_{P,Q}$ equals that of the rank profile matrix.

Definition 17. The decomposition $A = PLUQ$ reveals:

- (1) the row rank profile if $\text{RowSupp}(\Pi_{P,Q}) = \text{RowSupp}(\mathcal{R}_A)$,
- (2) the column rank profile if $\text{ColSupp}(\Pi_{P,Q}) = \text{ColSupp}(\mathcal{R}_A)$,
- (3) the rank profile matrix if $\Pi_{P,Q} = \mathcal{R}_A$.

Example 18. $A = \begin{bmatrix} 2 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 2 & 0 & 1 \end{bmatrix}$ has $\mathcal{R}_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ for rank profile matrix over \mathbb{Q} . Now the pivoting matrix obtained from a PLUQ decomposition with a pivot search operation following the row order (any column, first nonzero row) could be the matrix $\Pi_{P,Q} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. As these matrices share the same row support, the matrix $\Pi_{P,Q}$ reveals the row rank profile of A .

Remark 19. Example 18 suggests that a pivot search strategy minimizing row and column indices could be a sufficient condition to recover both row and column rank profiles at the same time, regardless the pivot permutation. However, this is unfortunately not the case. Consider for example a search based on the lexicographic order (first nonzero column of the first nonzero row) with transposition permutations, run on the matrix: $A = \begin{bmatrix} 0 & 0 & 1 \\ 2 & 3 & 0 \end{bmatrix}$. Its rank profile matrix is $\mathcal{R}_A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ whereas the pivoting matrix would be $\Pi_{P,Q} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, which does not reveal the column rank profile. This is due to the fact

that the column transposition performed for the first pivot changes the order in which the columns will be inspected in the search for the second pivot.

We will show that if the pivot permutations preserve the order in which the still unprocessed columns or rows appear, then the pivoting matrix will equal the rank profile matrix. This is achieved by the monotonically increasing permutations. Theorem 20 shows how the ability of a PLUQ decomposition algorithm to recover the rank profile information relates to the use of monotonically increasing permutations. More precisely, it considers an arbitrary step in a PLUQ decomposition where k pivots have been found in the elimination of an $\ell \times p$ leading sub-matrix A_1 of the input matrix A .

Theorem 20. *Consider a partial PLUQ decomposition of an $m \times n$ matrix A :*

$$A = P_1 \begin{bmatrix} L_1 & \\ & M_1 \ I_{m-k} \end{bmatrix} \begin{bmatrix} U_1 & V_1 \\ & H \end{bmatrix} Q_1$$

where $\begin{bmatrix} L_1 \\ M_1 \end{bmatrix}$ is $m \times k$ lower triangular and $\begin{bmatrix} U_1 & V_1 \end{bmatrix}$ is $k \times n$ upper triangular, and let A_1 be some $\ell \times p$ leading sub-matrix of A , for $\ell, p \geq k$. Let $H = P_2 L_2 U_2 Q_2$ be a PLUQ decomposition of H . Consider the PLUQ decomposition

$$A = P_1 \underbrace{\begin{bmatrix} I_k & \\ & P_2 \end{bmatrix}}_P \underbrace{\begin{bmatrix} L_1 & \\ & P_2^T M_1 \ L_2 \end{bmatrix}}_L \underbrace{\begin{bmatrix} U_1 & V_1 Q_2^T \\ & U_2 \end{bmatrix}}_U \underbrace{\begin{bmatrix} I_k & \\ & Q_2 \end{bmatrix}}_Q Q_1.$$

Consider the following clauses:

- (i) $\text{RowRP}(A_1) = \text{RowSupp}(\Pi_{P_1, Q_1})$
- (ii) $\text{ColRP}(A_1) = \text{ColSupp}(\Pi_{P_1, Q_1})$
- (iii) $\mathcal{R}_{A_1} = \Pi_{P_1, Q_1}$
- (iv) $\text{RowRP}(H) = \text{RowSupp}(\Pi_{P_2, Q_2})$
- (v) $\text{ColRP}(H) = \text{ColSupp}(\Pi_{P_2, Q_2})$
- (vi) $\mathcal{R}_H = \Pi_{P_2, Q_2}$
- (vii) P_1^T is k -monotonically increasing or $(P_1^T$ is ℓ -monotonically increasing and $p = n$)

(viii) Q_1^T is k -monotonically increasing or $(Q_1^T$ is p -monotonically increasing and $\ell = m$)
Then,

- (a) if (i) or (ii) or (iii) then $H = \begin{bmatrix} 0^{(\ell-k) \times (p-k)} & * \\ & * & * \end{bmatrix}$
- (b) if (vii) then ((i) and (iv)) $\Rightarrow \text{RowRP}(A) = \text{RowSupp}(\Pi_{P, Q})$;
- (c) if (viii) then ((ii) and (v)) $\Rightarrow \text{ColRP}(A) = \text{ColSupp}(\Pi_{P, Q})$;
- (d) if (vii) and (viii) then (iii) and (vi) $\Rightarrow \mathcal{R}_A = \Pi_{P, Q}$.

Proof. Let $P_1 = \begin{bmatrix} P_{11} & E_1 \end{bmatrix}$ and $Q_1 = \begin{bmatrix} Q_{11} \\ F_1 \end{bmatrix}$ where E_1 is $m \times (m-k)$ and F_1 is $(n-k) \times n$.

On one hand we have

$$A = \underbrace{\begin{bmatrix} P_{11} & E_1 \\ L_1 \\ M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} Q_{11} \\ F_1 \end{bmatrix}}_B + E_1 H F_1. \quad (1)$$

On the other hand,

$$\begin{aligned} \Pi_{P,Q} &= P_1 \begin{bmatrix} I_k \\ P_2 \end{bmatrix} \begin{bmatrix} I_r & \\ & 0_{(m-r) \times (n-r)} \end{bmatrix} \begin{bmatrix} I_k \\ Q_2 \end{bmatrix} Q_1 \\ &= P_1 \begin{bmatrix} I_k \\ \Pi_{P_2, Q_2} \end{bmatrix} Q_1 = \Pi_{P_1, Q_1} + E_1 \Pi_{P_2, Q_2} F_1. \end{aligned}$$

Let $\bar{A}_1 = \begin{bmatrix} A_1 & 0 \\ 0 & 0_{(m-\ell) \times (n-p)} \end{bmatrix}$ and denote by B_1 the $\ell \times p$ leading sub-matrix of B .

- (a) The clause (i) or (ii) or (iii) implies that all k pivots of the partial elimination were found within the $\ell \times p$ sub-matrix A_1 . Hence $\text{rank}(A_1) = k$ and we can write

$$P_1 = \begin{bmatrix} P_{11} & E_1 \\ 0_{(m-\ell) \times k} \end{bmatrix} \text{ and } Q_1 = \begin{bmatrix} Q_{11} & 0_{k \times (n-p)} \\ F_1 \end{bmatrix}, \text{ and the matrix } A_1 \text{ writes}$$

$$A_1 = \begin{bmatrix} I_\ell & 0 \end{bmatrix} A \begin{bmatrix} I_p \\ 0 \end{bmatrix} = B_1 + \begin{bmatrix} I_\ell & 0 \end{bmatrix} E_1 H F_1 \begin{bmatrix} I_p \\ 0 \end{bmatrix}. \quad (2)$$

Now $\text{rank}(B_1) = k$ as a sub-matrix of B of rank k and since

$$\begin{aligned} B_1 &= \begin{bmatrix} P_{11} & [I_\ell \ 0] \cdot E_1 \end{bmatrix} \begin{bmatrix} L_1 \\ M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} Q_{11} \\ F_1 \cdot \begin{bmatrix} I_p \\ 0 \end{bmatrix} \end{bmatrix} \\ &= P_{11} L_1 U_1 Q_{11} + \begin{bmatrix} I_\ell & 0 \end{bmatrix} E_1 M_1 \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1 \begin{bmatrix} I_p \\ 0 \end{bmatrix} \end{aligned}$$

where the first term, $P_{11} L_1 U_1 Q_{11}$, has rank k and the second term has a disjoint row support.

Finally, consider the term $\begin{bmatrix} I_\ell & 0 \end{bmatrix} E_1 H F_1 \begin{bmatrix} I_p \\ 0 \end{bmatrix}$ of equation (2). As its row support is disjoint with that of the pivot rows of B_1 , it has to be composed of rows linearly dependent with the pivot rows of B_1 to ensure that $\text{rank}(A_1) = k$. As its column support is disjoint with that of the pivot columns of B_1 , we conclude that it must be the zero matrix. Therefore the leading $(\ell - k) \times (p - k)$ sub-matrix of $E_1 H F_1$ is zero.

- (b) From (a) we know that $A_1 = B_1$. Thus $\text{RowRP}(B) = \text{RowRP}(A_1)$. Recall that $A = B + E_1 H F_1$. No pivot row of B can be made linearly dependent by adding rows of $E_1 H F_1$, as the column position of the pivot is always zero in the latter matrix. For the same reason, no pivot row of $E_1 H F_1$ can be made linearly dependent by adding rows of B . From (i), the set of pivot rows of B is $\text{RowRP}(A_1)$, which shows that

$$\text{RowRP}(A) = \text{RowRP}(A_1) \cup \text{RowRP}(E_1 H F_1). \quad (3)$$

Let $\sigma_{E_1} : \{1..m-k\} \rightarrow \{1..m\}$ be the map representing the sub-permutation E_1 (i.e. such that $E_1[\sigma_{E_1}(i), i] = 1 \forall i$). If P_1^T is k -monotonically increasing, the matrix E_1 has full column rank and is in column echelon form, which implies that

$$\begin{aligned} \text{RowRP}(E_1 H F_1) &= \sigma_{E_1}(\text{RowRP}(H F_1)) \\ &= \sigma_{E_1}(\text{RowRP}(H)), \end{aligned} \quad (4)$$

since F_1 has full row rank. If P_1^T is ℓ monotonically increasing, we can write $E_1 = \begin{bmatrix} E_{11} & E_{12} \end{bmatrix}$, where the $m \times (m - \ell)$ matrix E_{12} is in column echelon form. If $p = n$,

the matrix H writes $H = \begin{bmatrix} 0_{(\ell-k) \times (n-k)} \\ H_2 \end{bmatrix}$. Hence we have $E_1 H F_1 = E_{12} H_2 F_1$ which

also implies

$$\text{RowRP}(E_1 H F_1) = \sigma_{E_1}(\text{RowRP}(H)).$$

From equation (2), the row support of $\Pi_{P,Q}$ is that of $\Pi_{P_1, Q_1} + E_1 \Pi_{P_2, Q_2} F_1$, which is the union of the row support of these two terms as they are disjoint. Under the conditions of point (b), this row support is the union of $\text{RowRP}(A_1)$ and $\sigma_{E_1}(\text{RowRP}(H))$, which is, from (4) and (3), $\text{RowRP}(A)$.

(c) Similarly as for point (b).

(d) From (a) we have still $A_1 = B_1$. Now since $\text{rank}(B) = \text{rank}(B_1) = \text{rank}(A_1) = k$, there is no other nonzero element in \mathcal{R}_B than those in $\mathcal{R}_{\overline{A_1}}$ and $\mathcal{R}_B = \mathcal{R}_{\overline{A_1}}$. The row and column support of \mathcal{R}_B and that of $E_1 H F_1$ are disjoint. Hence

$$\mathcal{R}_A = \mathcal{R}_{\overline{A_1}} + \mathcal{R}_{E_1 H F_1}. \quad (5)$$

If both P_1^T and Q_1^T are k -monotonically increasing, the matrix E_1 is in column echelon form and the matrix F_1 in row echelon form. Consequently, the matrix $E_1 H F_1$ is a copy of the matrix H with k zero-rows and k zero-columns interleaved, which does not impact the linear dependency relations between the nonzero rows and columns. As a consequence

$$\mathcal{R}_{E_1 H F_1} = E_1 \mathcal{R}_H F_1. \quad (6)$$

Now if Q_1^T is k -monotonically increasing, P_1^T is ℓ -monotonically increasing and $p = n$, then, using notations of point (b), $E_1 H F_1 = E_{12} H_2 F_1$ where E_{12} is in column echelon form. Thus $\mathcal{R}_{E_1 H F_1} = E_1 \mathcal{R}_H F_1$ for the same reason. The symmetric case where Q_1^T is p -monotonically increasing and $\ell = m$ works similarly. Combining equations (2), (5) and (6) gives $\mathcal{R}_A = \Pi_{P,Q}$.

□

4. Algorithms for the rank profile matrix

Using Theorem 20, we deduce what rank profile information is revealed by a PLUQ algorithm by the way the search and the permutation operations are done. Table 2 summarizes these results, and points to instances known in the literature, implementing the corresponding type of elimination. More precisely, we first distinguish in this table the ability to compute the row or column rank profile or the rank profile matrix, but we also indicate whether the resulting PLUQ decomposition preserves the monotonicity of

the rows or columns. Indeed some algorithm may compute the rank profile matrix, but break the precedence relation between the linearly dependent rows or columns, making it unusable as a base case for a block algorithm of higher level.

Search	Row Perm.	Col. Perm.	Reveals	Monotonicity	Instance
Row order	Transposition	Transposition	RowRP		Ibarra et al. (1982); Jeannerod et al. (2013)
Col. order	Transposition	Transposition	ColRP		Keller-Gehrig (1985); Jeannerod et al. (2013)
Lexico.	Transposition	Transposition	RowRP		Storjohann (2000)
	Transposition	Rotation	RowRP, ColRP, \mathcal{R}	Col.	here
	Rotation	Rotation	RowRP, ColRP, \mathcal{R}	Row, Col.	here
Rev. lexico.	Transposition	Transposition	ColRP		Storjohann (2000)
	Rotation	Transposition	RowRP, ColRP, \mathcal{R}	Row	here
	Rotation	Rotation	RowRP, ColRP, \mathcal{R}	Row, Col.	here
Product	Rotation	Transposition	RowRP	Row	here
	Transposition	Rotation	ColRP	Col	here
	Rotation	Rotation	RowRP, ColRP, \mathcal{R}	Row, Col.	Dumas et al. (2013)

Table 2. Pivoting Strategies revealing rank profiles

4.1. Iterative algorithms

We start with iterative algorithms, where each iteration handles one pivot at a time. Here Theorem 20 is applied with $k = 1$, and the partial elimination represents how one pivot is being treated. The elimination of H is done by induction.

4.1.1. Row and Column order Search

The row order pivot search operation is of the form: *any nonzero element in the first nonzero row*. Each row is inspected in order, and a new row is considered only when the previous row is all zeros. With the notations of Theorem 20, this means that A_1 is the leading $\ell \times n$ sub-matrix of A , where ℓ is the index of the first nonzero row of A . When permutations P_1 and Q_1 , moving the pivot from position (ℓ, j) to (k, k) are transpositions, the matrix Π_{P_1, Q_1} is the element $E_{\ell, j}$ of the canonical basis. Its row rank profile is (ℓ) which is that of the $\ell \times n$ leading sub-matrix A_1 . Finally, the permutation P_1 is ℓ -monotonically increasing, and Theorem 20 case (b) can be applied to prove by induction that any such algorithm will reveal the row rank profile: $\text{RowRP}(A) = \text{RowSupp}(\Pi_{P, Q})$. The case of the column order search is similar.

4.1.2. Lexicographic order based pivot search

In this case the pivot search operation is of the form: *first nonzero element in the first nonzero row*. The lexicographic order being compatible with the row order, the above results hold when transpositions are used and the row rank profile is revealed. If in addition column rotations are used, $Q_1 = R_{1, j}$ which is 1-monotonically increasing. Now $\Pi_{P_1, Q_1} = E_{\ell, j}$ which is the rank profile matrix of the $\ell \times n$ leading sub-matrix A_1 of A . Theorem 20 case (d) can be applied to prove by induction that any such algorithm will reveal the rank profile matrix: $\mathcal{R}_A = \Pi_{P, Q}$. Lastly, the use of row rotations, ensures that the order of the linearly dependent rows will be preserved as well. Algorithm 2 is an instance of Gaussian elimination with a lexicographic order search and rotations for row and column permutations.

The case of the reverse lexicographic order search is similar. As an example, the algorithm in (Storjohann, 2000, Algorithm 2.14) is based on a reverse lexicographic order search but with transpositions for the row permutations. Hence it only reveals the column rank profile.

4.1.3. Product order based pivot search

The search here consists in finding any nonzero element $A_{\ell,p}$ such that the $\ell \times p$ leading sub-matrix A_1 of A is all zeros except this coefficient. If the row and column permutations are the rotations $R_{1,\ell}$ and $R_{1,p}$, we have $\Pi_{P_1,Q_1} = E_{\ell,p} = \mathcal{R}_{A_1}$. Theorem 20 case (d) can be applied to prove by induction that any such algorithm will reveal the rank profile matrix: $\mathcal{R}_A = \Pi_{P,Q}$. An instance of such an algorithm is given in (Dumas et al., 2013, Algorithm 2). If P_1 (resp. Q_1) is a transposition, then Theorem 20 case (c) (resp. case (b)) applies to show by induction that the columns (resp. row) rank profile is revealed.

4.2. Recursive algorithms

A recursive Gaussian elimination algorithm can either split one of the row or column dimension, cutting the matrix in wide or tall rectangular slabs, or split both dimensions, leading to a decomposition into tiles.

4.2.1. Slab recursive algorithms

Most algorithms computing rank profiles are slab recursive Ibarra et al. (1982); Keller-Gehrig (1985); Storjohann (2000); Jeannerod et al. (2013). When the row dimension is split, this means that the search space for pivots is the whole set of columns, and Theorem 20 applies with $p = n$. This corresponds to either a row or a lexicographic order. From case (b), one shows that, with transpositions, the algorithm recovers the row rank profile, provided that the base case does. If in addition, the elementary column permutations are rotations, then case (d) applies and the rank profile matrix is recovered. Finally, if rows are also permuted by monotonically increasing permutations, then the PLUQ decomposition also respects the monotonicity of the linearly dependent rows and columns. The same reasoning holds when splitting the column dimension.

4.2.2. Tile recursive algorithms

Tile recursive Gaussian elimination algorithms Dumas et al. (2013); Malaschonok (2010); Dumas and Roch (2002) are more involved, especially when dealing with rank deficiencies. Algorithm 1 recalls the tile recursive algorithm that the authors proposed in Dumas et al. (2013).

Here, the search area A_1 has arbitrary dimensions $\ell \times p$, often specialized as $m/2 \times n/2$. As a consequence, the pivot search can not satisfy either a row, a column, a lexicographic or a reverse lexicographic order. Now, if the pivots selected in the elimination of A_1 minimize the product order, then they necessarily also respect this order as pivots of the whole matrix A . Now, from (a), the remaining matrix H writes $H = \begin{bmatrix} 0_{(\ell-k) \times (p-k)} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$ and its elimination is done by two independent eliminations on the blocks H_{12} and H_{21} , followed by some update of H_{22} and a last elimination on it. Here again, pivots minimizing the row order on H_{21} and H_{12} are also pivots minimizing this order for H , and so are those of the fourth elimination. Now the block row and column permutations used in (Dumas et al., 2013, Algorithm 1) to form the PLUQ decomposition are r -monotonically increasing. Hence, from case (d), the algorithm computes the rank profile

Algorithm 1 PLUQ

Input: $A = (a_{ij})$ a $m \times n$ matrix over a field

Output: P, Q : $m \times m$ and $n \times n$ permutation matrices, r , the rank of A

Output: $A \leftarrow \begin{bmatrix} L \setminus U & V \\ M & 0 \end{bmatrix}$ where L and U are $r \times r$ resp. unit lower and upper triangular, and $A = P \begin{bmatrix} L \\ M \end{bmatrix} \begin{bmatrix} U & V \end{bmatrix} Q$.

if $\min(m, n) \leq \text{Threshold}$ **then**

 Call a base case implementation

end if

Split $A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$ where A_1 is $\lfloor \frac{m}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$.

Decompose $A_1 = P_1 \begin{bmatrix} L_1 \\ M_1 \end{bmatrix} \begin{bmatrix} U_1 & V_1 \end{bmatrix} Q_1$

▷ PLUQ(A_1)

$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \leftarrow P_1^T A_2$

▷ PermR(A_2, P_1^T)

$\begin{bmatrix} C_1 & C_2 \end{bmatrix} \leftarrow A_3 Q_1^T$

▷ PermC(A_3, Q_1^T)

Here $A = \begin{bmatrix} L_1 \setminus U_1 & V_1 & B_1 \\ M_1 & 0 & B_2 \\ C_1 & C_2 & A_4 \end{bmatrix}$.

$D \leftarrow L_1^{-1} B_1$

▷ TRSM(L_1, B_1)

$E \leftarrow C_1 U_1^{-1}$

▷ TRSM(C_1, U_1)

$F \leftarrow B_2 - M_1 D$

▷ MM(B_2, M_1, D)

$G \leftarrow C_2 - E V_1$

▷ MM(C_2, E, V_1)

$H \leftarrow A_4 - E D$

▷ MM(A_4, E, D)

Here $A = \begin{bmatrix} L_1 \setminus U_1 & V_1 & D \\ M_1 & 0 & F \\ E & G & H \end{bmatrix}$.

Decompose $F = P_2 \begin{bmatrix} L_2 \\ M_2 \end{bmatrix} \begin{bmatrix} U_2 & V_2 \end{bmatrix} Q_2$

▷ PLUQ(F)

Decompose $G = P_3 \begin{bmatrix} L_3 \\ M_3 \end{bmatrix} \begin{bmatrix} U_3 & V_3 \end{bmatrix} Q_3$

▷ PLUQ(G)

$\begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \leftarrow P_3^T H Q_2^T$

▷ PermR(H, P_3^T); PermC(H, Q_2^T)

$\begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \leftarrow P_3^T E$

▷ PermR(E, P_3^T)

$\begin{bmatrix} M_{11} \\ M_{12} \end{bmatrix} \leftarrow P_2^T M_1$

▷ PermR(M_1, P_2^T)

$\begin{bmatrix} D_1 & D_2 \end{bmatrix} \leftarrow D Q_2^T$

▷ PermR(D, Q_2^T)

$\begin{bmatrix} V_{11} & V_{12} \end{bmatrix} \leftarrow V_1 Q_3^T$

▷ PermR(V_1, Q_3^T)

Here $A = \begin{bmatrix} L_1 \setminus U_1 & V_{11} & V_{12} & D_1 & D_2 \\ M_{11} & 0 & 0 & L_2 \setminus U_2 & V_2 \\ M_{12} & 0 & 0 & M_2 & 0 \\ E_1 & L_3 \setminus U_3 & V_3 & H_1 & H_2 \\ E_2 & M_3 & 0 & H_3 & H_4 \end{bmatrix}$.

$I \leftarrow H_1 U_2^{-1}$

▷ TRSM(H_1, U_2)

$J \leftarrow L_3^{-1} I$

▷ TRSM(L_3, I)

$K \leftarrow H_3 U_2^{-1}$

▷ TRSM(H_3, U_2)

$N \leftarrow L_3^{-1} H_2$

▷ TRSM(L_3, H_2)

$O \leftarrow N - J V_2$

▷ MM(N, J, V_2)

$R \leftarrow H_4 - K V_2 - M_3 O$

▷ MM(H_4, K, V_2); MM(H_4, M_3, O)

Decompose $R = P_4 \begin{bmatrix} L_4 \\ M_4 \end{bmatrix} \begin{bmatrix} U_4 & V_4 \end{bmatrix} Q_4$

▷ PLUQ(R)

$\begin{bmatrix} E_{21} & M_{31} & 0 & K_1 \\ E_{22} & M_{32} & 0 & K_2 \end{bmatrix} \leftarrow P_4^T \begin{bmatrix} E_2 & M_3 & 0 & K \end{bmatrix}$

▷ PermR

$\begin{bmatrix} D_{21} & D_{22} \\ V_{21} & V_{22} \\ 0 & 0 \\ O_1 & O_2 \end{bmatrix} \leftarrow \begin{bmatrix} D_2 \\ V_2 \\ 0 \\ O \end{bmatrix} Q_4^T$

▷ PermC

Algorithm 2 Crout variant of PLUQ with lexicographic search and column rotations

```
1:  $k \leftarrow 1$ 
2: for  $i = 1 \dots m$  do
3:    $A_{i,k..n} \leftarrow A_{i,k..n} - A_{i,1..k-1} \times A_{1..k-1,k..n}$ 
4:   if  $A_{i,k..n} = 0$  then
5:     Loop to next iteration
6:   end if
7:   Let  $A_{i,s}$  be the left-most nonzero element of row  $i$ .
8:    $A_{i+1..m,s} \leftarrow A_{i+1..m,s} - A_{i+1..m,1..k-1} \times A_{1..k-1,s}$ 
9:    $A_{i+1..m,s} \leftarrow A_{i+1..m,s} / A_{i,s}$ 
10:  Bring  $A_{*,s}$  to  $A_{*,k}$  by column rotation
11:  Bring  $A_{i,*}$  to  $A_{k,*}$  by row rotation
12:   $k \leftarrow k + 1$ 
13: end for
```

r -sub-permutation matrix whose nonzero elements positions are chosen uniformly at random. The effective speed is obtained by dividing an estimate of the arithmetic cost ($2mnr + 2/3r^3 - r^2(m+n)$) by the computation time.

Figure 1 shows the computation speed of Algorithm 2 (3), compared to that of the pure recursive algorithm (6), and to our previous base case algorithm in [Dumas et al. \(2013\)](#), using a product order search, and either a left-looking (4) or a right-looking (5) schedule. At $n = 200$, the left-looking variant (4) improves over the right looking variant (5) by a factor of about 3.18 as it performs fewer modular reductions. Then, the Crout variant (3) again improves variant (4) by a factor of about 9.29. Lastly we also show the speed of the final implementation, formed by the tile recursive algorithm cascading to either the Crout base case (1) or the left-looking one (2). The threshold where the cascading to the base case occurs is experimentally set to its optimum value, i.e. 256 for variant (1) and 40 for variant (2). This illustrates that the gain on the base case efficiency leads to a higher threshold, and improves the efficiency of the cascade implementation (by an additive gain of about 1 effective Gfops in the range of dimensions considered).

Note that the experiments reported here differ from that of [Dumas et al. \(2015b\)](#). We found a mistake in the code generating the random matrices, making their row rank profile generic, which led to a reduced amount of row permutations and therefore a much higher computation speed. After fixing this issue, we noticed a slow-down of most base case implementations, but all variants still compare in the same way. The spikes in curves (1) and (2) showing drops in computation speed in a few points are reproducible, and seem to correspond to pathological dimensions, for which the blocking generates worst case cache misses scenari.

6. Relations with other triangularizations

6.1. The LEU decomposition

The LEU decomposition of [Malaschonok \(2010\)](#) involves a lower triangular matrix L , an upper triangular matrix U and a r -sub-permutation matrix E . It is also called the *modified Bruhat decomposition* in [Tyrtshnikov \(1997\)](#). Theorem 21 shows how to recover an LEU decomposition from a PLUQ decomposition revealing the rank profile matrix.

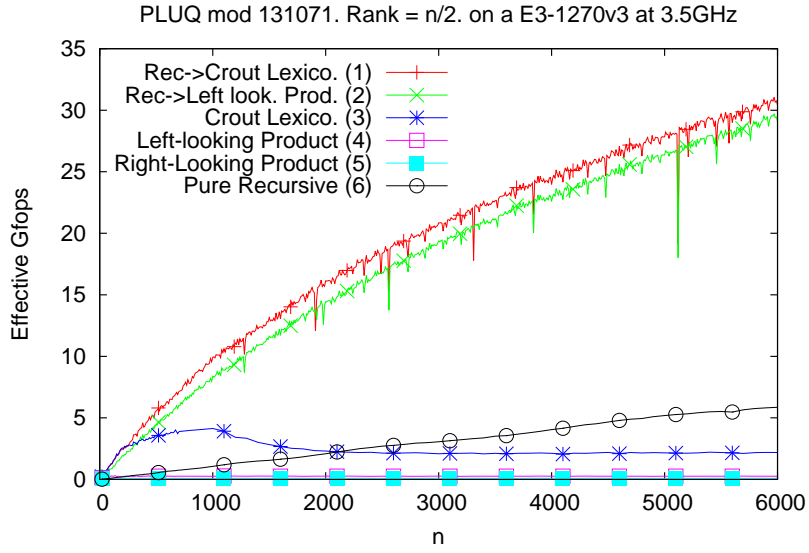
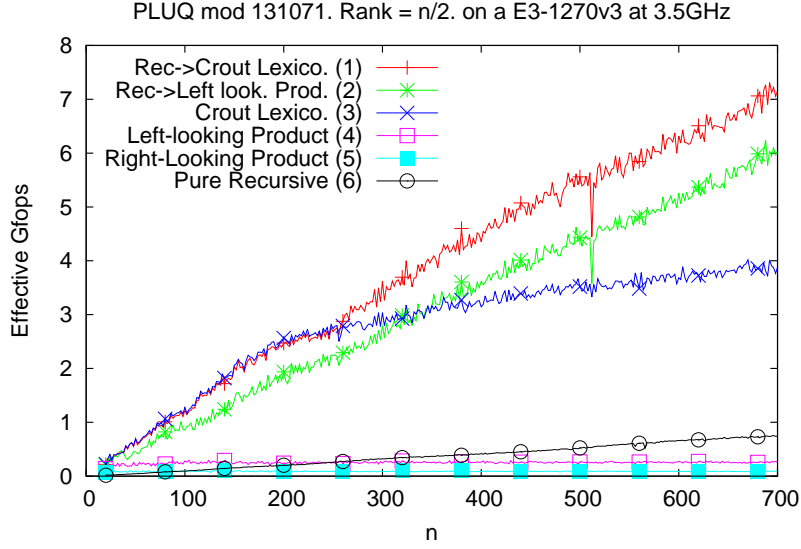


Fig. 1. Computation speed of PLUQ decomposition base cases.

Theorem 21. Let $A = PLUQ$ be a PLUQ decomposition revealing the rank profile matrix ($\Pi_{P,Q} = \mathcal{R}_A$). Then an LEU decomposition of A with $E = \mathcal{R}_A$ is obtained as follows (only using row and column permutations):

$$A = P \underbrace{\begin{bmatrix} L & 0_{m \times (m-r)} \end{bmatrix}}_{\bar{L}} P^T P \underbrace{\begin{bmatrix} I_r \\ 0 \end{bmatrix}}_E Q Q^T \underbrace{\begin{bmatrix} U \\ 0_{(n-r) \times n} \end{bmatrix}}_{\bar{U}} Q \quad (7)$$

Proof. First $E = P \begin{bmatrix} I_r & \\ & 0 \end{bmatrix} Q = \Pi_{P,Q} = \mathcal{R}_A$. Then there only needs to show that \bar{L} is lower triangular and \bar{U} is upper triangular. Suppose that \bar{L} is not lower triangular, let i be the first row index such that $\bar{L}_{i,j} \neq 0$ for some $i < j$. First $j \in \text{RowRP}(A)$ since the nonzero columns in \bar{L} are placed according to the first r values of P . Remarking that $A = P \begin{bmatrix} L & 0_{m \times (n-r)} \\ & I_{n-r} \end{bmatrix} Q$, and since right multiplication by a nonsingular matrix does not change row rank profiles, we deduce that $\text{RowRP}(\Pi_{P,Q}) = \text{RowRP}(A) = \text{RowRP}(\bar{L})$. If $i \notin \text{RowRP}(A)$, then the i -th row of \bar{L} is linearly dependent with the previous rows, but none of them has a nonzero element in column $j > i$. Hence $i \in \text{RowRP}(A)$.

Let (a, b) be the position of the coefficient $\bar{L}_{i,j}$ in L , that is $a = \sigma_P^{-1}(i)$, $b = \sigma_P^{-1}(j)$. Let also $s = \sigma_Q(a)$ and $t = \sigma_Q(b)$ so that the pivots at diagonal position a and b in L respectively correspond to ones in \mathcal{R}_A at positions (i, s) and (j, t) . Consider the $\ell \times p$ leading sub-matrices A_1 of A where $\ell = \max_{x=1..a-1}(\sigma_P(x))$ and $p = \max_{x=1..a-1}(\sigma_Q(x))$. On one hand (j, t) is an index position in A_1 but not (i, s) , since otherwise $\text{rank}(A_1) = b$. Therefore, $(i, s) \not\prec_{\text{prod}}(j, t)$, and $s > t$ as $i < j$. As coefficients (j, t) and (i, s) are pivots in \mathcal{R}_A and $i < j$ and $t < s$, there can not be a nonzero element above (j, t) at row i when it is chosen as a pivot. Hence $\bar{L}_{i,j} = 0$ and \bar{L} is lower triangular. The same reasoning applies to show that \bar{U} is upper triangular. \square

Remark 22. Note that the LEU decomposition with $E = \mathcal{R}_A$ is not unique, even for invertible matrices. As a counter-example, the following decomposition holds for any $a \in \mathbb{K}$:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix} \quad (8)$$

However, uniqueness of the full rank LEU decomposition can be achieved by imposing an additional constraint on the L matrix: there is a unique LEU decomposition where L is such that $E^T L E$ is also lower triangular (Manthey and Helmke, 2007, Th. 2.1). We remark that for the LEU decomposition proposed in (7), this condition becomes:

$$\text{the matrix } E^T \bar{L} E = Q_{*,1..r}^T L_{1..r,1..r} Q_{*,1..r} \text{ must be lower triangular.} \quad (9)$$

Whenever $Q_{*,1..r}$ is in column echelon form, which means that the sequence of the pivot's column positions is monotonically increasing, then the above matrix is lower triangular. A sufficient condition to ensure it, is thus that the pivoting strategy of the corresponding PLUQ decomposition minimizes the reverse lexicographic order.

Corollary 23. *Let $A = PLUQ$ be a full-rank PLUQ decomposition computed with a pivoting strategy minimizing the reverse lexicographic order and performing row rotations. Then the LEU decomposition of equation (7) is the unique modified Bruhat decomposition as defined in (Manthey and Helmke, 2007, Theorem 2.1).*

6.2. The Bruhat decomposition

The Bruhat decomposition, from which Malaschonok's LEU decomposition was inspired in Malaschonok (2010), is another decomposition with a central permutation matrix, see Bruhat (1956); Della Dora (1973); Bourbaki (2008); Grigoriev (1981). Bruhat's theorem is stated in terms of Weyl groups but reduces to the following:

Theorem 24 (Bourbaki (2008)). Any invertible matrix A can be written as $A = VPU$ for V and U upper triangular invertible matrices and P a permutation matrix. The latter decomposition is called the Bruhat decomposition of A .

It was then naturally extended to singular square matrices in Grigoriev (1981) and an algorithm over principal ideals domains given in Malaschonok (2013). Corollary 25 generalizes it to matrices with arbitrary dimensions, and relates it to the PLUQ decomposition.

Corollary 25. Any $m \times n$ matrix of rank r has a VP decomposition, where V and U are upper triangular matrices, and P is a r -sub-permutation matrix.

Proof. Let J_n be the unit anti-diagonal matrix. From the LEU decomposition of $J_n A$, we have $A = \underbrace{J_n L J_n}_V \underbrace{J_n E}_P U$ where V is upper triangular. \square

Remark 26. Manthey and Helmke (2007) gives also a unique *generalized Bruhat decomposition*, $A = XPY$, but with non-square and non-triangular matrices X, Y . There, P is called the *Bruhat permutation* but contains only the r nonzero rows and columns of the rank profile matrix. We show in Section 6.5, that X and Y are actual row and column echelon forms of the matrix.

6.3. Relation to LUP and PLU decompositions

The LUP decomposition $A = LUP$ only exists for matrices with generic row rank profile (including matrices with full row rank). Corollary 27 shows upon which condition the permutation matrix P equals the rank profile matrix \mathcal{R}_A . Note that although the row rank profile of A is trivial in such cases, the matrix \mathcal{R}_A still carries non-trivial information not only on the column rank profile but also on the row and column rank profiles of any leading sub-matrix of A .

Corollary 27. Let A be an $m \times n$ matrix.

If A has generic column rank profile, then any PLU decomposition $A = PLU$ computed using reverse lexicographic order search and row rotations is such that $\mathcal{R}_A = P \begin{bmatrix} I_r & \\ & 0 \end{bmatrix}$. In particular, $P = \mathcal{R}_A$ if $r = m = n$.

If A has generic row rank profile, then any LUP decomposition $A = LUP$ computed using lexicographic order search and column rotations is such that $\mathcal{R}_A = \begin{bmatrix} I_r & \\ & 0 \end{bmatrix} P$. In particular, $P = \mathcal{R}_A$ if $r = m = n$.

Proof. Consider A has generic column rank profile. From table 2, any PLUQ decomposition algorithm with a reverse lexicographic order based search and rotation based row permutation is such that $\Pi_{P,Q} = P \begin{bmatrix} I_r & \\ & \end{bmatrix} Q = \mathcal{R}_A$. Since the search follows the reverse lexicographic order and the matrix has generic column rank profile, no column will be permuted in this elimination, and therefore $Q = I_n$. The same reasoning hold for when A has generic row rank profile. \square

Note that the L and U factors in a PLU decomposition are uniquely determined by the permutation P . Hence, when the matrix has full row rank, $P = \mathcal{R}_A$ and the

decomposition $A = \mathcal{R}_A LU$ is unique. Similarly the decomposition $A = LU\mathcal{R}_A$ is unique when the matrix has full column rank. Now when the matrix is rank deficient with generic row rank profile, there is no longer a unique PLU decomposition revealing the rank profile matrix: any permutation applied to the last $m - r$ columns of P and the last $m - r$ rows of L yields a PLU decomposition where $\mathcal{R}_A = P \begin{bmatrix} I_r \\ \end{bmatrix}$.

Lastly, we remark that the only situation where the rank profile matrix \mathcal{R}_A can be read directly as a sub-matrix of P or Q is as in corollary 27, when the matrix A has generic row or column rank profile. Consider a PLUQ decomposition $A = PLUQ$ revealing the rank profile matrix ($\mathcal{R}_A = P \begin{bmatrix} I_r \\ \end{bmatrix} Q$) such that \mathcal{R}_A is a sub-matrix of P . This means that $P = \mathcal{R}_A + S$ where S has disjoint row and column support with \mathcal{R}_A . We have $\mathcal{R}_A = (\mathcal{R}_A + S) \begin{bmatrix} I_r \\ \end{bmatrix} Q = (\mathcal{R}_A + S) \begin{bmatrix} Q_1 \\ 0_{(n-r) \times n} \end{bmatrix}$. Hence $\mathcal{R}_A (I_n - \begin{bmatrix} Q_1 \\ 0_{(n-r) \times n} \end{bmatrix}) = S \begin{bmatrix} Q_1 \\ 0_{(n-r) \times n} \end{bmatrix}$ but the row support of these matrices are disjoint, hence $\mathcal{R}_A \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} = 0$ which implies that A has generic column rank profile. Similarly, one shows that \mathcal{R}_A can be a sub-matrix of Q only if A has a generic row rank profile.

6.4. Computing Echelon forms

Usual algorithms computing an echelon form by [Ibarra et al. \(1982\)](#); [Keller-Gehrig \(1985\)](#); [Storjohann \(2000\)](#); [Jeannerod et al. \(2013\)](#) use a slab block decomposition (with row or lexicographic order search), which implies that pivots appear in the order of the echelon form. The column echelon form is simply obtained as $C = PL$ from the PLUQ decomposition.

We extend here this approach to any PLUQ decomposition that reveals the rank profile matrix ($\mathcal{R}_A = \Pi_{P,Q}$) and show how both the row and column echelon forms can be recovered by only permutations on the L and U matrices. Besides the ability to compute both echelon forms from one Gaussian elimination, this enables one to choose from a broader range of algorithmic variants for this Gaussian elimination.

Consider a PLUQ decomposition $A = PLUQ$ revealing the rank profile matrix and let $P_C = P_{*,1..r}$ be the first r columns of P . The ones in P_C indicate the row position of the pivots in the column echelon form of A , but they may not appear in the same column order. There exist an $r \times r$ permutation matrix S sorting the columns of P_C , i.e. such that $P_S = P_C S$ is in column echelon.

Lemma 28. *The matrix $\begin{bmatrix} PLS & 0_{m \times n-r} \end{bmatrix}$ is a column echelon form of A .*

Proof. $PLS = P \begin{bmatrix} L & 0_{m \times (m-r)} \end{bmatrix} P^T P \begin{bmatrix} S \\ 0_{(m-r) \times r} \end{bmatrix} = \bar{L} P_S$. From Theorem 21, the matrix $\bar{L} = P \begin{bmatrix} L & 0_{m \times (m-r)} \end{bmatrix} P^T$ is lower triangular. Multiplying \bar{L} on the right by P_S simply removes the zero columns in \bar{L} putting it in column echelon form. Now the relation $A = \begin{bmatrix} PLS & 0_{m \times (n-r)} \end{bmatrix} \begin{bmatrix} S^T & \\ & I_{n-r} \end{bmatrix} \begin{bmatrix} U \\ 0_{(n-r) \times r} & I_{n-r} \end{bmatrix} Q$ shows that $\begin{bmatrix} PLS & 0_{m \times (n-r)} \end{bmatrix}$ is right-equivalent to A and is therefore a column echelon form for A . \square

Equivalently, the same reasoning applies for the recovery of a row echelon form of A . Algorithm 3 summarizes how a row and a column echelon form of A can be computed by sorting the first r values of the permutations σ_P and σ_Q .

Algorithm 3 Echelon forms from a PLUQ decomposition

Input: P, L, U, Q , a PLUQ decomposition of A with $\mathcal{R}_A = \Pi_{P,Q}$

Output: C : a column echelon form of A

Output: R : a row echelon form of A

- 1: $(p_1, \dots, p_r) = \text{Sort}(\sigma_P(1), \dots, \sigma_P(r))$
 - 2: $(q_1, \dots, q_r) = \text{Sort}(\sigma_Q(1), \dots, \sigma_Q(r))$
 - 3: $\tau = (\sigma_P^{-1}(p_1), \dots, \sigma_P^{-1}(p_r)) \in S_r$
 - 4: $\chi = (\sigma_Q^{-1}(q_1), \dots, \sigma_Q^{-1}(q_r)) \in S_r$
 - 5: $S_\tau = P(\tau)$
 - 6: $S_\chi = P(\chi)^T$
 - 7: **Return** $C \leftarrow P \begin{bmatrix} LS_\tau & 0_{m \times (n-r)} \end{bmatrix}$ and $R \leftarrow \begin{bmatrix} S_\chi U \\ 0_{(m-r) \times n} \end{bmatrix} Q$
-

Remark 29. A row and column echelon form of the $i \times j$ leading sub-matrix can be computed by removing rows of PL below index i and filtering out the pivots of column index greater than j . The latter is achieved by replacing lines 1 and 2 in Algorithm 3 by:

- 1': $(p_1, \dots, p_s) = \text{Sort}(\{\sigma_P(k) : \sigma_Q(k) \leq j \text{ and } 1 \leq k \leq r\})$
- 2': $(q_1, \dots, q_s) = \text{Sort}(\{\sigma_Q(k) : \sigma_P(k) \leq i \text{ and } 1 \leq k \leq r\})$.

6.5. The generalized Bruhat decomposition

The generalization of the Bruhat decomposition for rank deficient matrices of (Manthey and Helmke, 2007, Theorem 2.4) is of the form $A = XFY$ where X is in column echelon form, Y in row echelon form, and F is a $r \times r$ permutation matrix. Such a decomposition immediately follows from the echelon forms computed by Algorithm 3. Indeed, by Lemma 28, PLS_τ is in column echelon form and $S_\chi UQ$ is in row echelon form. The identity

$$A = \underbrace{(PLS_\tau)}_X \underbrace{S_\tau^T S_\chi^T}_F \underbrace{(S_\chi UQ)}_Y \quad (10)$$

then directly gives an $A = XFY$ decomposition.

In a similar way as in the full-rank case (see Section 6.1) (Manthey and Helmke, 2007, Theorem 2.4) requires an additional condition on the X matrix to ensure the uniqueness of the decomposition: $F^T X_{\text{RowRP}(X),*} F$ must be lower triangular, where $X_{I,*}$ denotes the lower triangular submatrix of X formed by the rows of its row rank profile. This condition is not satisfied in general by the X and Y matrices computed from Algorithm 3 and a PLUQ decomposition revealing the rank profile matrix. As a counter example, the matrix $\begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}$ has two PLUQ decompositions revealing the rank profile matrix $F = J_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$: $A = J_2 I_2 \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} I_2$ and $A = I_2 \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} I_2 J_2$. For the latter one, $F^T X F = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$ which is not lower triangular.

In a same way as in equation (9), we derive an equivalent condition for the uniqueness of the generalized Bruhat decomposition. We first need the following Lemma.

Lemma 30. *Let C be an $m \times r$ matrix in column echelon form, and P the $m \times r$ r -sub-permutation matrix with ones at the location of the pivots of C . Then*

$$C_{\text{RowRP}(X),*} = P^T C.$$

As $P_{*,1..r}S_\tau$ is the r -sub-permutation matrix with ones on the locations of the pivots of the column echelon form X , we deduce that

$$F^T X_{\text{RowRP}(X),*} F = S_\chi S_\tau S_\tau^T (P_{*,1..r})^T P L S_\tau S_\tau^T S_\chi^T = S_\chi L_{1..r,1..r} S_\chi^T.$$

Again, a sufficient condition for this matrix to be lower triangular is that $S_\chi = I_r$ which happens if the PLUQ decomposition has been computed using a pivoting respecting the reverse lexicographic order.

Corollary 31. *Let $A = PLUQ$ be a PLUQ decomposition computed with a pivoting strategy minimizing the reverse lexicographic order and performing row rotations. Then the XFY decomposition of equation (10) is the unique generalized Bruhat decomposition as defined in (Manthey and Helmke, 2007, Theorem 2.4).*

7. Improvement for low rank matrices

7.1. Storjohann and Yang's algorithm

An alternative way to compute row and column rank profiles has recently been proposed by Storjohann and Yang in Storjohann and Yang (2014, 2015), reducing the time complexity from a deterministic $O(mnr^{\omega-2})$ to a Monte Carlo probabilistic $2r^3 + O(r^2(\log m + \log n) + mn)$ in Storjohann and Yang (2014) first, and to $(r^\omega + mn)^{1+o(1)}$ in Storjohann and Yang (2015), under the condition that the base field contains at least $2 \min(m, n)(1 + \lceil \log_2 m \rceil + \lceil \log_2 n \rceil)$ elements. We show how these results can be extended to the computation of the rank profile matrix.

The technique of Storjohann and Yang originates from the oracle system solving technique of (Mulders and Storjohann, 2000, §2). There, a linear system is solved or proved to be inconsistent, by the incremental construction of a non-singular sub-matrix of the augmented matrix of the system, and the computation of its inverse by rank one updates.

Applied to a right-hand-side vector b sampled uniformly from the column space of a matrix, this technique is used to incrementally build a list of row indices $\mathcal{P} = [i_1, \dots, i_s]$ and of column indices $\mathcal{Q} = [j_1, \dots, j_s]$, for $s = 1, \dots, r$, forming an incrementally growing invertible sub-matrix $A_{\mathcal{P}, \mathcal{Q}}$. Each coordinate (i_s, j_s) is that of a pivot as in a standard Gaussian elimination algorithm, but the complexity improvement comes from the fact that the elimination is only performed on this $s \times s$ submatrix, and not on the linearly dependent rows and columns of the matrix A . To achieve this, the search for pivots is done by limiting the Schur complement computation to the last column of the augmented system $\left[A \middle| b \right]$:

$$b - A_{*, \mathcal{Q}} \cdot (A_{\mathcal{P}, \mathcal{Q}})^{-1} \cdot b_{\mathcal{P}}.$$

Under a randomized assumption, the first non-zero component of this vector indicates the value i_s of the next linearly independent row. A similar approach is used for determining the value of j_s .

As shown in (Storjohann and Yang, 2014, Theorem 6), the algorithm ends up computing the row rank profile in \mathcal{P} and the column rank profile in \mathcal{Q} . In fact, it computes all information of the rank profile matrix.

Theorem 32. *The r -permutation matrix whose ones are located at positions (i_s, j_s) for $1 \leq s \leq r$ is precisely the rank profile matrix of A .*

Proof. In this algorithm, the pivot search looks for the first non-zero element in the Schur complement of b which, with probability $1 - 1/\#\mathbb{K}$, will correspond to the first linearly independent row. The exact Schur complement can then be computed on this row only:

$$A_{i_s,*} - A_{i_s,\mathcal{Q}} \cdot (A_{\mathcal{P},\mathcal{Q}})^{-1} \cdot A_{\mathcal{P},*},$$

so as to find the column position j_s of the first non zero element in it. This search strategy minimizes the lexicographic order of the pivot coordinates. No permutation strategy is being used, instead the columns where pivots have already been found are zeroed out in the Schur complement computation, which has the same effect as a Column rotation. From the 4th line of Table 2, this pivoting strategy reveals the rank profile matrix. \square

The complexity of the direct Monte Carlo algorithm, in $O((m+n)r^2)$, hence directly applies for the computation of the rank profile matrix. This complexity is reduced to $2r^3 + O(r^2(\log n + \log m) + mn)$ by the introduction of linear independence oracles in a divide and conquer scheme. Upon success of the probabilistic assumption, the values for \mathcal{P} and \mathcal{Q} are unchanged. Consequently, (Storjohann and Yang, 2014, Theorem 19, Corollary 21) also hold for the computation of the rank profile matrix.

Corollary 33. *There exist a Monte Carlo algorithm computing the rank profile matrix that has running time bounded by $2r^3 + (r^2 + m + n + |A|)^{1+o(1)}$, where $|A|$ denotes the number of non-zero elements in A .*

The r^3 term in this complexity is from the iterative construction of the inverses of the non-singular sub-matrices of order s for $1 \leq s \leq r$, by rank one updates. To reduce this complexity to $O(r^\omega)$, Storjohann and Yang (2015) propose a relaxed computation of this online matrix inversion. In order to group arithmetic operations into matrix multiplications, their approach is to anticipate part of the updates on the columns that will be appended in the future. This however requires that one has an a priori knowledge on the position of r linearly independent columns of the initial matrix A and, even further, that the sub-matrix formed by these linearly independent columns has generic rank profile. The first condition is ensured by the Monte Carlo selection of linearly independent columns of (Cheung et al., 2013, Theorem 2.11) and the second by the use of a lower triangular Toeplitz preconditioner as in (Kaltofen and Saunders, 1991, Theorem 2). In the process, the row rank profile can be recovered, but all information on the column rank profile is lost, and the rank profile matrix can thus not be recovered.

Yet, this algorithm can be run twice (once on A and once on A^T), to recover the row and the column rank profiles, and extract the corresponding $r \times r$ invertible sub-matrix $A_{\mathcal{P},\mathcal{Q}}$ of A . Following Corollary 27, it then suffices to compute a PLU decomposition of this sub-matrix, using an appropriate pivoting strategy, to recover its rank profile matrix and therefore that of A as well.

Theorem 34. *Algorithm 4 is Monte Carlo probabilistic and computes the rank profile matrix of A in time $(r^\omega + m + n + |A|)^{1+o(1)}$ field operations in \mathbb{K} .*

Note that the last operation consists in the introduction of zero rows and columns to expand the $r \times r$ permutation matrix P into the $m \times n$ matrix \mathcal{R}_A . The position of these zero rows and columns are deduced from the row rank profile \mathcal{P} and the column rank profile \mathcal{Q} .

Algorithm 4 Low Rank Profile Matrix

Input: A , an $m \times n$ matrix of rank r over a field K .

Output: \mathcal{R}_A the rank profile matrix of A or FAILURE

- 1: Compute the Row rank profile $\mathcal{P} = [i_1, \dots, i_r]$ of A using [Storjohann and Yang \(2015\)](#)
 - 2: Compute the Column rank profile $\mathcal{Q} = [j_1, \dots, j_r]$ of A using [Storjohann and Yang \(2015\)](#) applied on A^T .
 - 3: Let $B = A_{\mathcal{P}, \mathcal{Q}}$
 - 4: Compute $B = LUP$ a LUP decomposition of B using a lexicographic pivot search and rotations for the column permutations.
 - 5: Return $\mathcal{R}_A = P_{\mathcal{P}} \begin{bmatrix} P & \\ & 0_{(m-r) \times (n-r)} \end{bmatrix} P_{\mathcal{Q}}^T$
-

7.2. Online LU decomposition

The rank profile algorithms of Storjohann and Yang rely on an online computation of a matrix inverse, either by rank one updates in [Storjohann and Yang \(2014\)](#) or by a relaxed variant [Storjohann and Yang \(2015\)](#), that corresponds to an online version of the classic divide and conquer algorithm [Strassen \(1969\)](#). We remark that these inversion algorithms can be advantageously replaced by LU decomposition algorithms. Following the argument in ([Jeannerod et al., 2013](#), Fig 2), the use of LU decomposition for linear system solving offers a speed-up factor of about 3, compared to the use of a matrix inverse. Indeed, solving a system from an LU decomposition is done by two back substitutions, which has the same costs of $2n^2 + O(n)$ field operations, as applying the inverse of the matrix to the right-hand-side vector. But the cost computing an LU decomposition is about three times as fast as computing a matrix inverse ($2/3n^3$ versus $2n^3$ when classic matrix arithmetic is used). We present now how to replace the online matrix inverse by an online LU decomposition in [Storjohann and Yang \(2014, 2015\)](#).

First consider the iterative construction of the LU decomposition of $A_{\mathcal{P}, \mathcal{Q}}$ by rank one updates. Suppose that $A_{s-1} = LU$ with L lower triangular with a unit diagonal, and U upper triangular. Then we have

$$\left[\begin{array}{c|c} A_{s-1} & u \\ \hline v & d \end{array} \right] = \left[\begin{array}{c|c} L & \\ \hline vU^{-1} & 1 \end{array} \right] \cdot \left[\begin{array}{c|c} U & L^{-1}u \\ \hline & w \end{array} \right], \quad (11)$$

where $w = d - vU^{-1}L^{-1}u$. This rank one update of the LU decomposition of the work matrix costs $2s^2 + O(s)$ to compute for an $(s-1) \times (s-1)$ matrix $A_{\mathcal{P}, \mathcal{Q}}$ (compared with $6s^2 + O(s)$ in ([Storjohann and Yang, 2014](#), Lemma 2)). The remaining of the paper can then be used, replacing every multiplication by the pre-computed matrix inverse $x \leftarrow (A_{\mathcal{P}, \mathcal{Q}})^{-1} \cdot b$ by two consecutive triangular system solving: $y \leftarrow L^{-1}b$; $x \leftarrow U^{-1}y$. This leads to a Monte Carlo algorithm computing the rank profile matrix in time $2/3r^3 + (r^2 + m + n + |A|)^{(1+o(1))}$.

Note that the decomposition in (11) uses two kinds of matrix vector products: vU^{-1} and $L^{-1}u$ contrary to the matrix inverse formula (12) used in [Storjohann and Yang \(2014\)](#) and [Storjohann and Yang \(2015\)](#)

$$\left[\begin{array}{c|c} A_{s-1} & u \\ \hline v & d \end{array} \right]^{-1} = \left[\begin{array}{c|c} I_{s-1} & -A_{s-1}^{-1}u_s(d - v_sA_{s-1}^{-1}u_s)^{-1} \\ \hline & (d - v_sA_{s-1}^{-1}u_s)^{-1} \end{array} \right] \cdot \left[\begin{array}{c|c} I_{s-1} & \\ \hline -v_s & 1 \end{array} \right] \cdot \left[\begin{array}{c|c} A_{s-1}^{-1} & \\ \hline & 1 \end{array} \right], \quad (12)$$

where only one matrix-vector product, $A_{s-1}^{-1}u_s$, appears. The relaxation is achieved there by anticipating part of these products into larger matrix-matrix multiplications. This requires that all of the column vectors that will be treated are known in advance: this is ensured by the selection of r linearly independent columns of Cheung et al. (2013). Since no matrix-vector product involves row vectors v_s , it is still possible to select and add them incrementally one after the other, thus allowing to compute the row rank profile.

Now, relaxing the LU decomposition of equation (11) in a similar way would require to also anticipate computations on the rows, namely vU^{-1} . This would be either too expensive, if no pre-selection of r linearly independent rows is performed, or such a pre-selection would lose the row rank profile information that has to be computed. Hence we can not propose a similar improvement of the leading constant in the relaxed case.

References

- K.P.S. Bhaskara Rao. *The Theory of Generalized Inverses Over Commutative Rings*, volume 17 of *Algebra Logic and Application series*. Taylor and Francis, London, 2002.
- N. Bourbaki. *Groupes et Algèbres de Lie*. Number Chapters 4–6 in *Elements of mathematics*. Springer, 2008.
- William C. Brown. *Matrices over Commutative Rings*. Chapman & Hall Pure and Applied Mathematics. CRC Press, 1992. ISBN 9780824787554.
- William C. Brown. Null ideals and spanning ranks of matrices. *Communications in Algebra*, 26(8):2401–2417, 1998. doi: 10.1080/00927879808826285.
- François Bruhat. Sur les représentations induites des groupes de Lie. *Bulletin de la Société Mathématique de France*, 84:97–205, 1956. URL <http://eudml.org/doc/86911>.
- Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast Matrix Rank Algorithms and Applications. *J. ACM*, 60(5):31:1–31:25, October 2013. ISSN 0004-5411. doi: 10.1145/2528404.
- W. Edwin Clark and Joseph J Liang. Enumeration of finite commutative chain rings. *Journal of Algebra*, 27(3):445–453, December 1973. ISSN 0021-8693. doi: 10.1016/0021-8693(73)90055-0.
- Jean Della Dora. *Sur quelques algorithmes de recherche de valeurs propres*. Theses, Université Joseph Fourier - Grenoble I, July 1973. URL <https://tel.archives-ouvertes.fr/tel-00010274>.
- Jack J. Dongarra, Lain S. Duff, Danny C. Sorensen, and Henk A. Vander Vorst. *Numerical Linear Algebra for High Performance Computers*. SIAM, 1998. ISBN 0898714281.
- Jean-Guillaume Dumas and Jean-Louis Roch. On parallel block algorithms for exact triangularizations. *Parallel Computing*, 28(11):1531–1548, November 2002. doi: 10.1016/S0167-8191(02)00161-8.
- Jean-Guillaume Dumas, Pascal Giorgi, and Clément Pernet. Dense linear algebra over prime fields. *ACM TOMS*, 35(3):1–42, November 2008. doi: 10.1145/1391989.1391992.
- Jean-Guillaume Dumas, Clément Pernet, and Ziad Sultan. Simultaneous computation of the row and column rank profiles. In Manuel Kauers, editor, *Proc. ISSAC'13*, pages 181–188. ACM Press, 2013. doi: 10.1145/2465506.2465517.
- Jean-Guillaume Dumas, Thierry Gautier, Clément Pernet, and Ziad Sultan. Parallel computation of echelon forms. In *Euro-Par 2014 Parallel Proc.*, LNCS (8632), pages 499–510. Springer, 2014. ISBN 978-3-319-09872-2. doi: 10.1007/978-3-319-09873-9_42.

- Jean-Guillaume Dumas, Clément Pernet, and Ziad Sultan. Recursion based parallelization of exact dense linear algebra routines for Gaussian elimination. *Parallel Computing*, 2015a. doi: [10.1016/j.parco.2015.10.003](https://doi.org/10.1016/j.parco.2015.10.003).
- Jean-Guillaume Dumas, Clément Pernet, and Ziad Sultan. Computing the rank profile matrix. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC '15*, pages 149–156, New York, NY, USA, 2015b. ACM. ISBN 978-1-4503-3435-8. doi: [10.1145/2755996.2756682](https://doi.org/10.1145/2755996.2756682).
- Dima Yu. Grigoriev. Analogy of Bruhat decomposition for the closure of a cone of Chevalley group of a classical serie. *Soviet Mathematics Doklady*, 23(2):393–397, 1981.
- Thomas W. Hungerford. On the structure of principal ideal rings. *Pacific Journal of Mathematics*, 25(3):543–547, 1968. URL <http://projecteuclid.org/euclid.pjm/1102986148>.
- Oscar H. Ibarra, Shlomo Moran, and Roger Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *J. of Algorithms*, 3(1):45–56, 1982. doi: [10.1016/0196-6774\(82\)90007-4](https://doi.org/10.1016/0196-6774(82)90007-4).
- Claude-Pierre Jeannerod, Clément Pernet, and Arne Storjohann. Rank-profile revealing Gaussian elimination and the CUP matrix decomposition. *J. Symbolic Comput.*, 56: 46–68, 2013. ISSN 0747-7171. doi: [10.1016/j.jsc.2013.04.004](https://doi.org/10.1016/j.jsc.2013.04.004).
- David John Jeffrey. LU factoring of non-invertible matrices. *ACM Comm. Comp. Algebra*, 44(1/2):1–8, July 2010. ISSN 1932-2240. doi: [10.1145/1838599.1838602](https://doi.org/10.1145/1838599.1838602).
- Erich Kaltofen and B. David Saunders. On Wiedemann’s method of solving sparse linear systems. In Harold F. Mattson, Teo Mora, and T. R. N. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, number 539 in Lecture Notes in Computer Science, pages 29–38. Springer Berlin Heidelberg, January 1991. ISBN 978-3-540-54522-4 978-3-540-38436-6. doi: [10.1007/3-540-54522-0_93](https://doi.org/10.1007/3-540-54522-0_93).
- Walter Keller-Gehrig. Fast algorithms for the characteristic polynomial. *Th. Comp. Science*, 36:309–317, 1985. doi: [10.1016/0304-3975\(85\)90049-0](https://doi.org/10.1016/0304-3975(85)90049-0).
- Kenneth Klimkowski and Robert A. van de Geijn. Anatomy of a parallel out-of-core dense linear solver. In *International Conference on Parallel Processing*, volume 3, pages 29–33. CRC Press, aug 1995.
- Wolfgang Krull. Dimensionstheorie in stellenringen. *Journal für die reine und angewandte Mathematik*, 179:204–226, 1938. URL <http://eudml.org/doc/150048>.
- Gennadi Malaschonok. Generalized Bruhat decomposition in commutative domains. In VladimirP. Gerdt, Wolfram Koepf, ErnstW. Mayr, and EvgeniiV. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 8136 of *Lecture Notes in Computer Science*, pages 231–242. Springer International Publishing, 2013. ISBN 978-3-319-02296-3. doi: [10.1007/978-3-319-02297-0_20](https://doi.org/10.1007/978-3-319-02297-0_20).
- Gennadi Ivanovich Malaschonok. Fast generalized Bruhat decomposition. In *CASC'10*, volume 6244 of *LNCS*, pages 194–202. Springer-Verlag, Berlin, Heidelberg, 2010. ISBN 3-642-15273-2, 978-3-642-15273-3. doi: [10.1007/978-3-642-15274-0_16](https://doi.org/10.1007/978-3-642-15274-0_16).
- Wilfried Manthey and Uwe Helmke. Bruhat canonical form for linear systems. *Linear Algebra and its Applications*, 425(2–3):261 – 282, 2007. ISSN 0024-3795. doi: [10.1016/j.laa.2007.01.022](https://doi.org/10.1016/j.laa.2007.01.022). Special Issue in honor of Paul Fuhrmann.
- Neal H. McCoy. *Rings and ideals*. Carus Monograph Series, no. 8. The Open Court Publishing Company, LaSalle, Ill., 1948.
- Ezra Miller and Bernd Sturmfels. *Combinatorial commutative algebra*, volume 227. Springer, 2005. doi: [10.1007/b138602](https://doi.org/10.1007/b138602).

- Thom Mulders and Arne Storjohann. Rational Solutions of Singular Linear Systems. In *Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation*, ISSAC '00, pages 242–249, New York, NY, USA, 2000. ACM. ISBN 1-58113-218-2. doi: [10.1145/345542.345644](https://doi.org/10.1145/345542.345644).
- Graham H. Norton and Ana Salagean. On the hamming distance of linear codes over a finite chain ring. *IEEE Trans. Information Theory*, 46(3):1060–1067, 2000. doi: [10.1109/18.841186](https://doi.org/10.1109/18.841186).
- Arne Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, ETH-Zentrum, Zürich, Switzerland, November 2000.
- Arne Storjohann and Shiyun Yang. Linear Independence Oracles and Applications to Rectangular and Low Rank Linear Systems. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 381–388, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2501-1. doi: [10.1145/2608628.2608673](https://doi.org/10.1145/2608628.2608673).
- Arne Storjohann and Shiyun Yang. A Relaxed Algorithm for Online Matrix Inversion. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '15, pages 339–346, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3435-8. doi: [10.1145/2755996.2756672](https://doi.org/10.1145/2755996.2756672).
- Prof Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, August 1969. ISSN 0029-599X, 0945-3245. doi: [10.1007/BF02165411](https://doi.org/10.1007/BF02165411).
- E. Tyrtyshnikov. Matrix Bruhat decompositions with a remark on the QR (GR) algorithm. *Linear Algebra and its Applications*, 250:61 – 68, 1997. ISSN 0024-3795. doi: [10.1016/0024-3795\(95\)00453-X](https://doi.org/10.1016/0024-3795(95)00453-X).