LSP matrix decomposition revisited

Claude-Pierre Jeannerod

Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON-UCBL n° 5668

LSP Matrix Decomposition Revisited

Claude-Pierre Jeannerod

September, 2006

Research Report N° 2006-28

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

LSP Matrix Decomposition Revisited

Claude-Pierre Jeannerod

September, 2006

Abstract

In this paper, we study the problem of computing an LSP-decomposition of a matrix over a field. This decomposition is an extension to arbitrary matrices of the well-known LUP-decomposition of full row-rank matrices. We present three different ways of computing an LSP-decomposition, that are both rank-sensitive and based on matrix multiplication. In each case, for an m by n input matrix of unknown rank r, the cost we obtain is in $O(mnr^{\omega-2})$ for $\omega > 2$. When r is small, this improves the $O(nm^{\omega-1})$ complexity bound of Ibarra, Moran and Hui.

Keywords: Matrix factorization, matrix multiplication, reduced echelon form, rank profile, algorithmic complexity

Résumé

Cet article considère le problème du calcul d'une décomposition LSP d'une matrice à coefficients dans un corps. Cette décomposition est une extension aux matrices de rang quelconque de la décomposition LUP, classique pour les matrices de rang plein en lignes. On présente trois façons de calculer une décomposition LSP en fonction du rang et via le produit de matrices. Dans chaque cas, le coût obtenu pour une matrice $m \times n$ de rang r (inconnu a priori) est en $O(mnr^{\omega-2})$ avec $\omega > 2$. Pour r petit, cela améliore la borne de complexité en $O(nm^{\omega-1})$ d'Ibarra, Moran et Hui.

Mots-clés: Factorisation de matrice, produit de matrices, forme échelonnée, profil de rang, complexité algorithmique

1 Introduction

Let A be an m by n matrix over a field k. It is well known that if A has full row rank then it has an LUP-decomposition: $A = LUP$ where $L \in k^{m\times m}$ is unit lower triangular, $U \in k^{m\times n}$ is upper triangular with nonzero elements on the main diagonal, and $P \in k^{n\times n}$ is a permutation matrix. Furthermore, Bunch and Hopcroft's algorithm [3, 1] computes an LUP-decomposition of A in $O(nm^{\omega-1})$ field operations (see for example the proof of [4, Theorem 16.5]). Here, a field operation is any of $\{+, -, \times, \text{divide by a nonzero}, \text{compare with zero}\}$ and ω is such that two n by n matrices over k can multiplied in $O(n^\omega)$ field operations.

When A has not full row rank, LUP-decomposition may not exist anymore and Ibarra, Moran and Hui [8] extended it to the so-called LSP-decomposition (assuming $m \le n$): $A = LSP$ where L and P are as before, but where S is only *semi-upper triangular*, that is, deleting the zero rows of S yields an upper triangular matrix whose entries on the main diagonal are nonzero.

LSP-decomposition is interesting because it reveals the rank r of A as the number of nonzero rows of S. Let $i_1 < \cdots < i_r$ be the indices of those rows. Then another interesting property is that $(i_1, \ldots, i_r)$ is the *row rank profile* of A, that is, the lexicographically smallest subsequence $(h_1, \ldots, h_r)$ of $(1, \ldots, m)$ such that rows $h_1, \ldots, h_r$ have full rank. (The column rank profile is defined similarly by considering columns instead of rows.)

Following Bunch and Hopcroft's approach, Ibarra, Moran and Hui also reduced the problem of computing an LSP-decomposition to that of matrix multiplication: the algorithm they give in [8, §2]—which we shall call the *IMH algorithm*—is for $m \le n$ and has cost $O(nm^{\omega-1})$.

By definition, S has r nonzero rows and thus its size (that is, the number of field elements required to represent it) is only $O(rn)$. For L, the definition requires no more than a unit lower triangular shape. However, since rows $j \notin \{i_1, \ldots, i_r\}$ of S are zero and since L multiplies S, the values of $L_{j+1,j}, \ldots, L_{m,j}$ can be anything; by choosing them to be zero, we get a factor L whose jth column is e_j, the jth unit vector of k^m. For example, if $A \in k^{4\times 6}$ has row rank profile $(1, 3)$ then

$$S = \begin{bmatrix} * & * & * & * & * & * \\ & & & & & \\ & * & * & * & * & * \\ & & & & & \end{bmatrix} \rightarrow L = \begin{bmatrix} 1 & & & \\ * & 1 & & \\ * & & 1 & \\ * & & * & 1 \end{bmatrix} = \begin{bmatrix} * & \\ * & \\ * & * \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \end{bmatrix} + \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}.$$

With no loss of generality, we can thus consider that L has such a shape, with (vertical) stripes corresponding to the (horizontal) stripes of S, and we shall say that L is *striped according to* S. Clearly, such an L can now be represented in a more compact way as

$$L = MF^T + I_m, \quad M \in k^{m\times r}, \quad F = [e_{i_1}, \ldots, e_{i_r}] \in k^{m\times r}. \tag{1}$$

Here M is dense but strictly lower triangular and even in echelon form: for $1 \le j \le r$, $M_{i,j} = 0$ if $i \le i_j$. To sum up, $O((m + n)r)$ field elements are enough for representing an LSP-decomposition L, S, P with L striped according to S. Since such a decomposition has a size that depends on $r = \text{rank}(A)$, how to compute it in a rank-sensitive manner?

Gaussian elimination with column pivoting would do that with $O(mnr)$ field operations, which is indeed rank-sensitive, but unlike IMH it does not reduce to matrix multiplication.

Following the approach of [11], we show in this paper that an LSP-decomposition with L striped according to S can be computed both in a rank-sensitive way and via matrix

multiplication. We give three different ways of doing this, each of them having cost $O(mnr^{\omega-2})$ if $w > 2$ and $O(mn \log_2 r)$ if $\omega = 2$.

A first way, which we present in Section 2, is to compute the so-called reduced column echelon form C of A with Storjohann's fast, rank-sensitive `GaussJordan` algorithm [11, §2.2]. In Section 2.1 we show that moving from C to L, S, P then essentially reduces to LUP-decomposition of a full row-rank r by n matrix plus some matrix multiplications, for an extra cost of $O((m+n)r^{\omega-1})$. Since $r \leq \min\{m, n\}$, note that $(m+n)r^{\omega-1} = O(mnr^{\omega-2})$ even if $\omega = 2$. Conversely, we remark in Section 2.2 that moving from L, S, P to C can be done also with $O((m+n)r^{\omega-1})$ field operations.

A second way, presented in Section 3, is to improve the IMH algorithm [8] itself. Our improvement is twofold: remove the assumption that $m \leq n$; modify the elimination step to have its cost decreased from $O(nm^{\omega-1})$ to $O(mnr^{\omega-2})$ for $\omega \geq 2$. The resulting algorithm `LSP` is described in Section 3.1 and, using the techniques of [11, §1.3], its complexity is analysed in Section 3.2.

The third approach is to directly compute an LQUP-decomposition of A as defined in [8]: Q is an m by m permutation matrix, $U = [V^T \,|\, 0]^T$ where $V \in k^{r \times r}$ is upper triangular and invertible, and $QU = S$. As before, we can assume with no loss of generality that L is striped according to S and of the form (1). Then, because of the structure of S and V,

$$Q = \left[\begin{array}{c|c} F & G \end{array}\right] \quad \text{for some } G \in k^{m \times (m-r)}. \tag{2}$$

Therefore, using $QQ^T = I_m$,

$$A = LSP = LQUP = \left([M \,|\, 0] + Q\right)\left[\begin{array}{c} V \\ \hline 0 \end{array}\right]P. \tag{3}$$

In Section 4 we give an algorithm for computing an LQUP-decomposition of A in the above compressed form M, Q, V, P; again, its cost is $O(mnr^{\omega-2})$ if $w > 2$ and $O(mn \log_2 r)$ if $\omega = 2$. The advantage of such a compressed form is that now M and V can be stored as the first r columns and rows, respectively, of a single m by n matrix. Since recovering L, S, P from M, Q, V, P is easy, the latter form has been used in [9, 6, 5] to obtain extremely efficient in-place implementations of LSP-decomposition for matrices over finite fields. However no rank-sensitive complexities appear in these works.

As shown in [8, 10], LSP-/LQUP-decomposition has several applications beyong the rank and rank profile. Among them are linear system solving, computing a nullspace basis and diagonalizing transforms and generalized inverses. We conclude in Section 5 by remarking how such application problems may benefit from our improved complexity results. This may be seen as an alternative to Storjohann's Gauss-Jordan canonical form approach [11, §2.2].

2 LSP-decomposition and reduced echelon forms

For every $A \in k^{m \times n}$ there exists an invertible $U \in k^{m \times m}$ such that $UA = R$ is the *reduced row echelon form* of A. That form R is unique and also known as the Gauss-Jordan canonical form of A [11]. As a row echelon form, R displays the rank r and the column rank profile $(j_1, \ldots, j_r)$ of A: only the first r rows of R are nonzero and, for $1 \leq i \leq r$, the first nonzero entry in row i is R_{i,j_i}. What makes R reduced is the additional property that column j_i is the ith unit vector.

For example, if $A \in k^{5\times 9}$ has rank 3 and column rank profile $(2,5,7)$ then its reduced row echelon form R has the following shape:

$$R = \left[\begin{array}{ccccccccc} & 1 & * & * & & * & & * & * \\ & & & & 1 & * & & * & * \\ & & & & & & 1 & * & * \\ \hline & & & & & & & & \end{array}\right].$$

The *reduced column echelon form* of A can be defined as the transpose of the reduced row echelon form of A^T. If we call it C then $AV = C$ for some invertible matrix V and

$$C = \left[\begin{array}{c|c} E + F & 0 \end{array}\right], \quad \text{with } F = [e_{i_1}, \ldots, e_{i_r}] \in k^{m\times r},$$

and with $(i_1, \ldots, i_r)$ the row rank profile of A. Transposing the above 5×9 matrix example, we see that the first r columns of C simply consist of F (which has the 1's) superimposed on E (which has the $*$'s). This way of writing C will be used in the next two subsections.

Given A, how fast can we compute R or C? Gauss-Jordan elimination would do that in time $O(mnr)$. Another way is to use Storjohann's `GaussJordan` algorithm [11, p. 42] whose cost is $O(mnr^{\omega-2})$ if $\omega > 2$ and $O(mn \log_2 r)$ if $\omega = 2$. This recursive algorithm will produce in particular the rank r as well as a transform U such that $UA = R$. Then, moving from U to R is cheap because U is in fact returned in the form

$$U = U_1 U_2, \qquad \text{where } U_1 = \left[\begin{array}{cc} U_{11} & \\ U_{21} & I_{m-r} \end{array}\right] \text{ and } U_2 \text{ is an } m \times m \text{ permutation matrix.}$$

To get the r nonzero rows of R, it suffices to multiply U_{11} by the first r rows of $U_2 A$. The cost of this rectangular matrix product is at most $\lceil n/r \rceil \mathsf{MM}(r)$, which for $r \leq m$ is $O(mnr^{\omega-2})$.

By transposing twice, we obtain the same upper bound on the cost of the reduced column echelon form.

2.1 From the reduced column echelon form to L, S, P factors

Given the reduced column echelon form C, we already have the row rank profile $(i_1, \ldots, i_r)$ of A. To find some L, S, P factors, let C' consist of rows $i_1, \ldots, i_r$ of C. Then $C' = [I_r \,|\, 0]$ and thus

$$C = (E + F)C'.$$

Since $AV = C$, we have in particular $A'V = C'$ where A' consists of the pivot rows $i_1, \ldots, i_r$ of A. Since V is invertible, multiplying on the right by V^{-1} leads further to $A = (E + F)A'$. For example, if $A \in k^{9\times 5}$ has rank 3 and row rank profile $(2,5,7)$ then

$$A = (E+F)A' = \left[\begin{array}{ccc} & & \\ 1 & & \\ * & & \\ * & & \\ & 1 & \\ * & * & \\ & & 1 \\ * & * & * \\ * & * & * \end{array}\right] \left[\begin{array}{ccccc} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array}\right].$$

Notice that such a factorization reveals an important property of the reduced column echelon form $C = [E + F \,|\, 0]$: every row i of A is equal to a unique linear combination of earlier pivot

rows (that is, of rows $i_1, i_2, \ldots$ of A such that $i_1, i_2, \ldots \leq i$) and, above all, the coefficients of this combination are exactly the entries of the ith row of $E + F$.

Since $A' \in k^{r\times n}$ has full row rank, it has an LUP-decomposition, say $A' = L'U'P$. Let $S = FU'$. Since $U' \in k^{r\times n}$ is upper triangular, $S \in k^{m\times n}$ is semi-upper triangular. Therefore, using the fact that $F^TF = I_r$ we obtain the following factorization:

$$A = L''SP, \qquad \text{where } L'' = (E + F)L'F^T.$$

The last step to modify L'' slightly in order to find a unit lower triangular factor L. Let $E' \in k^{m\times r}$ be defined by

$$E' + F = (E + F)L'.$$

Since L' is unit lower triangular, $E'+F$ has exactly the same echelon form as $E+F$. (However, notice that row i_j of $E + F$ is not a unit vector but contains instead the jth row of L'.) It follows that $L'' = E'F^T + FF^T =$ (strictly lower triangular) + (diagonal) is lower triangular. To get a unit lower triangular factor, simply replace L'' with

$$L = E'F^T + I_m. \tag{4}$$

The rows $i_1, \ldots, i_r$ of S being zero, we have indeed $A = LSP$, which is now an LSP-decomposition of A. Furthermore, L in (4) is striped according to S.

The above move from C to L, S, P requires LUP-decomposition of an $r \times n$ matrix as well as computing the product $(E + F)L'$ where $E + F$ is $m \times r$ and L' is $r \times r$. The latter costs at most $\lceil m/r\rceil \mathsf{MM}(r) \in O(mr^{\omega-1})$ and the former can be done in time $O(nr^{\omega-1})$ with the Bunch and Hopcroft algorithm [3]. Hence a total of $O((m+n)r^{\omega-1})$, which is in $O(mnr^{\omega-2})$ for $r \leq \min\{m, n\}$. We thus have shown the following result:

Theorem 2.1 *An LSP-decomposition of $A \in k^{m\times n}$ of (unknown) rank r can be computed in time $O(mnr^{\omega-2})$ by calling Storjohann's reduced row echelon form algorithm and then the Bunch and Hopcroft LUP-decomposition algorithm. If $\omega = 2$, the cost becomes $O(mn \log_2 r)$.*

2.2 From L, S, P factors to the reduced column echelon form

Conversely, what if some L, S, P factors of A are given? First, deducing from S the row rank profile $(i_1, \ldots, i_r)$ of A and taking $F = [e_{i_1}, \ldots, e_{i_r}] \in k^{m\times r}$, an $LQUP$-decomposition follows easily: augment F with $m - r$ columns into a permutation matrix $Q = [F \,|\, G] \in k^{m\times m}$ and let $V = F^TS$; then V is upper triangular with nonzero elements on the main diagonal, and

$$A = LQUP, \quad U = \left[\frac{V}{0}\right].$$

A second step is to move further to the reduced column echelon form of A; this can be done fast with as claimed below.

Theorem 2.2 *Given $A = LSP \in k^{m\times n}$, one can deduce the reduced column echelon form of A together with a transformation matrix in $O((m + n)r^{\omega-1})$ operations.*

Proof. With no loss of generality, assume that L is striped according to S. Then, by (3) we have $A = (M + F)VP$. Let $N \in k^{r\times r}$ consist of rows $i_1, \ldots, i_r$ of $M + F$. Then N is unit

lower triangular and, writing A' for the submatrix of A that consists of rows $i_1, \ldots, i_r$, we get $A' = NVP$ and thus

$$A = (M + F)N^{-1}A'.$$

Now let $D \in k^{m \times r}$ be defined by $D + F = (M + F)N^{-1}$. It is not hard to see that the m by n matrix $[D + F \,|\, 0]$ is in reduced column echelon form; if we show that $A = [D + F \,|\, 0]W$ for some invertible matrix W then, by uniqueness, $[D + F \,|\, 0]$ must be *the* reduced column echelon form of A. To show that, let

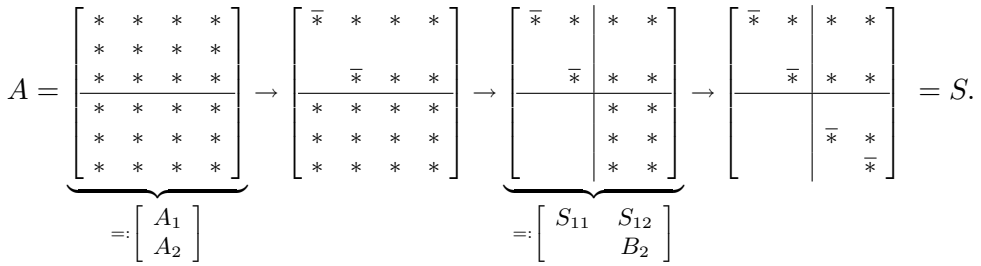
$$W = W_1 W_2, \quad W_1 = \begin{bmatrix} W_{11} & W_{12} \\ & I_{n-r} \end{bmatrix}, \quad \begin{bmatrix} W_{11} & W_{12} \end{bmatrix} = A'P^T, \quad W_2 = P^T.$$

Since $V = N^{-1}A'P^T$ has nonzero elements on the main diagonal, W_1 is invertible and so is W. On the other hand, using $(D + F)A' = A$, we obtain $A = [D + F \,|\, 0]W$. Therefore $[D + F \,|\, 0]$ is indeed the reduced column echelon form of A and a transformation matrix is W^{-1}.

Since $D+F = (M+F)N^{-1}$ is $m \times r$ and N is $r \times r$, one can obtain D with $O(mr^{\omega-1})$ field operations; since $[W_{11} \,|\, W_{12}]$ is $r \times n$ and W_{11} is $r \times r$, one can obtain W^{-1} with $O(nr^{\omega-1})$ field operations. ∎

3 A rank-sensitive version of the IMH algorithm

We now present another approach to computing $A = LSP$, which is a rank-sensitive extension of algorithm IMH (Ibarra, Mora and Hui [8]; see also [2, p. 103] for a description). The principle of IMH is as follows: cut A into horizontal slices A_1 and A_2 with roughly the same number of rows; compute $A_1 = L_1 S_1 P_1$ recursively; perform on $A_2 P_1^T$ an elimination step similar to that of block-Gaussian elimination; compute $B_2 = L_2 S_2 P_2$ recursively with B_2 the lower-right block produced by elimination. This principle is depicted below on a 6×4 matrix with row rank profile $(1, 3, 5, 6)$; there, $\overline{*}$ means a nonzero entry:



Factor S is obtained by stacking $[S_{11} \; S_{12}P_2^T]$ over $[0 \; S_2]$, with $[S_{11} \; S_{12}] = S_1$. Factors L and P record respectively the row operations and column interchanges needed for moving from A to S. For example, after the elimination step, we have an intermediate factorization of the form

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} L_1 & \\ L_{21} & I_{m_2} \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ & B_2 \end{bmatrix} P_1. \tag{5}$$

Now let r_1 be the rank of A_1. Then r_1 is also the number of nonzero rows of S_1, and S_{11} consists of the first r_1 columns of S_1.

Our algorithm improves on algorithm IMH in two ways. In [8] it is always assumed that $m \leq n$, which implies $r_1 < n$. Unlike algorithm IMH, the algorithm we describe in Section 3.1

works for arbitrary input dimensions $m \times n$ and the situation where $r_1 = n$ may occur. In that case, B_2 is an empty matrix and the second recursive call is not needed. Hence an explicit test to detect that $r_1 = n$ and then exit early. Second, algorithm IMH computes L_{21} and B_2 in $O(m^{\omega-1}n)$ field operations. We show in Section 3.2 that our algorithm computes them in $O(mnr^{\omega-2})$ with r the rank of A.

3.1 Algorithm description

We present here a rank-sensitive version of the IMH algorithm along with a correctness proof. The algorithm is written in Maple-like pseudo code. Note that $m \leq n$ is not assumed.

Algorithm `LSP`(A)
Input: $A \in k^{m \times n}$.
Output: an LSP-decomposition of A.
if $A = 0$ **then**
  $L, S, P := I_m, 0_{m \times n}, I_n$;
else if $m = 1$ **then**
  $j :=$ the index of the first nonzero entry of A;
  $P :=$ the permutation matrix that interchanges $A_{1,1}$ and $A_{1,j}$;
  $L, S := I_1, AP$;
else
  $m_1, m_2 := \lfloor m/2 \rfloor, \lceil m/2 \rceil$;
  $L_1, S_1, P_1 :=$ `LSP`(A_1) with A_1 the first m_1 rows of A;
  $r_1 :=$ the number of nonzero rows of S_1;
  if $r_1 = 0$ **then**
    $L_2, S_2, P_2 :=$ `LSP`(A_2) with A_2 the last m_2 rows of A;
    $L, S, P := \begin{bmatrix} I_{m_1} & \\ & L_2 \end{bmatrix}, \begin{bmatrix} \\ S_2 \end{bmatrix}, P_2$;
  else
    $S_{11} :=$ the first r_1 columns of S_1;
    $Q_1 :=$ the first r_1 rows of a permutation matrix Q such that $QS_{11} = [U^T\ 0]^T$
      with U upper triangular and nonsingular;
    $T_2 := A_2 P_1^T$ with A_2 the last m_2 rows of A;
    $L_{21} := (T_{21}U^{-1})Q_1$ with T_{21} the first r_1 columns of T_2;
    if $r_1 = n$ **then**
      $L, S, P := \begin{bmatrix} L_1 & \\ L_{21} & I_{m_2} \end{bmatrix}, \begin{bmatrix} S_1 \\ \end{bmatrix}, P_1$;
    else
      $B_2 := T_{22} - (L_{21}Q_1^T)(Q_1 S_{12})$ with S_{12}, T_{22} the last $n - r_1$ columns of S_1, T_2;
      $L_2, S_2, P_2 :=$ `LSP`(B_2);
      $L, S, P := \begin{bmatrix} L_1 & \\ L_{21} & L_2 \end{bmatrix}, \begin{bmatrix} S_{11} & S_{12}P_2^T \\ & S_2 \end{bmatrix}, \begin{bmatrix} I_{r_1} & \\ & P_2 \end{bmatrix} P_1$;
    fi;
  fi;
fi;
return L, S, P;

If $A = 0$ then correctness of Algorithm `LSP` is clear. If $A \neq 0$, we proceed by induction on m. If $m = 1$ then $P^2 = I_n$ and $A = (I_1)(AP)(P)$ is an LSP-decomposition of A. Assume now that $m > 1$. From $m_1 = \lfloor m/2 \rfloor$ and $m_2 = \lceil m/2 \rceil$, it follows that

$$m_1 \leq m_2 < m.$$

Hence by induction we have the LSP-decomposition $A_1 = L_1 S_1 P_1$; similarly, L_2, S_2, P_2 is an LSP-decomposition of A_2 when $r_1 = 0$ and of B_2 when $0 < r_1 < n$. Therefore, for each of the three cases "$r_1 = 0$", "$r_1 = n$" and "$0 < r_1 < n$", the matrices L, S, P computed by the algorithm have the desired shape (L is unit lower triangular, S is semi-upper triangular and P is a permutation matrix). It remains to show that the product LSP indeed equals A. To do that let us consider each case separately, using $r_1 = \text{rank}(A_1)$ and $U^{-1} Q_1 S_{11} = I_{r_1}$.

1. If $r_1 = 0$ then $A_1 = 0$ and $A_2 = L_2 S_2 P_2$, which leads to
$$A = \begin{bmatrix} \\ A_2 \end{bmatrix} = \begin{bmatrix} I_{m_1} & \\ & L_2 \end{bmatrix} \begin{bmatrix} \\ S_2 \end{bmatrix} P_2 = LSP.$$

2. If $r_1 = n$ then $S_{11} = S_1$ and $T_2 = T_{21}$. Hence $U^{-1} Q_1 S_1 = I_{r_1}$ and $A_2 = T_{21} P_1$. Since $L_{21} = T_{21} U^{-1} Q_1$, it follows that $L_{21} S_1 P_1 = A_2$. Using $A_1 = L_1 S_1 P_1$ we arrive at
$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} L_1 & \\ L_{21} & I_{m_2} \end{bmatrix} \begin{bmatrix} S_1 \\ \end{bmatrix} P_1 = LSP.$$

3. If $0 < r_1 < n$ then $L_{21} Q_1^T$ consists of the first r_1 columns of $L_{21} Q^T$ and $Q_1 S_{12}$ consists of the first r_1 rows of $Q S_{12}$. Since by definition of Q and S_1 all other rows of $Q S_{12}$ are zero, $(L_{21} Q_1^T)(Q_1 S_{12}) = (L_{21} Q^T)(Q S_{12}) = L_{21} S_{12}$. Therefore $T_{22} = L_{21} S_{12} + B_2$. In addition, $U^{-1} Q_1 S_{11} = I_{r_1}$ gives $T_{21} = L_{21} S_{11}$. Since $T_2 P_1 = [T_{21}\ T_{22}] P_1 = A_2$ and $L_1 [S_{11}\ S_{12}] P_1 = A_1$ we get the intermediate factorization (5). Then $B_2 = L_2 S_2 P_2$ gives
$$A = \begin{bmatrix} L_1 & \\ L_{21} & L_2 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} P_2^T \\ & S_2 \end{bmatrix} \begin{bmatrix} I_{r_1} & \\ & P_2 \end{bmatrix} P_1 = LSP.$$

To sum up, the output L, S, P satisfies $A = LSP$ for each case and correctness follows.

Notice that so far we have used only the fact that $m_1 + m_2 = m$ with $m_1, m_2 < m$. The values $m_1 = \lfloor m/2 \rfloor$ and $m_2 = \lceil m/2 \rceil$ will be used in the next subsection for bounding the complexity of the algorithm.

3.2 Complexity analysis

First let us verify that the unit lower triangular factor L computed by Algorithm `LSP` has the expected sparsity structure.

Property 3.1 *The factor L computed by Algorithm* `LSP` *is striped according to S.*

Proof. For $A = 0$ this is clear since $L - I_m$ is zero. For A nonzero, let us proceed by induction on m, the case $m = 1$ being clear too. Assume $m > 1$ and recall that $m_1 = \lfloor m/2 \rfloor \leq m_2 = \lceil m/2 \rceil < m$. If $r_1 = 0$ then $(i_1, \ldots, i_r) = (m_1 + i''_1, \ldots, m_1 + i''_r)$ with (i''_j) the row rank profile of A_2, and the assertion follows by induction. If $r_1 > 0$ then the row rank profile of A_1

is $(i_1, \ldots, i_{r_1})$; hence by induction only those columns of $L_1 - I_{m_1}$ can be nonzero. Now, the jth column of Q_1 is zero for all $j \notin \{i_1, \ldots, i_{r_1}\}$, for otherwise U would have a zero row (a contradiction). It follows that only columns $i_1, \ldots, i_{r_1}$ of L_{21} can be nonzero. If $r_1 = n$ then we are done. If $r_1 < n$ then let (i''_j) be the row rank profile of B_2. Since from (5) one has $r_2 := \mathrm{rank}(B_2) = r - r_1$ and $i_{r_1+j} = m_1 + i''_j$ for $1 \leq j \leq r_2$, the claim follows by induction. ∎

Therefore L requires only $O(mr)$ field elements compared to $m(m-1)/2$ for general $m \times m$ unit lower triangular matrices. Now let us bound the cost of the elimination step, that is, of computing L_{21} and B_2.

Lemma 3.2 *Computing L_{21} and B_2 can be done with $O(mnr^{\omega-2})$ field operations.*

Proof. For $L_{21} = (T_{21}U^{-1})Q_1$, the invertible matrix U is $r_1 \times r_1$ and T_{21} is $m_2 \times r_1$. In addition, $r_1 = \mathrm{rank}(A_1)$ implies $r_1 \leq m_2$. Hence $T_{21}U^{-1}$ can be computed in time $O(m_2 r_1^{\omega-1})$. For B_2, let $C_{21} = L_{21}Q_1^T$ and $R_{12} = Q_1 S_{12}$. Then C_{21} is $m_2 \times r_1$ and R_{12} is $r_1 \times (n - r_1)$. We have already seen that $r_1 \leq m_2$. Now, if $r_1 \leq n - r_1$ then one can compute in $C_{21}R_{12}$ in time $O(m_2(n-r_1)r_1^{\omega-2})$; else simply multiply C_{21} by R_{12} padded first with $2r_1 - n$ zero columns, and that in time $O(m_2 r_1^{\omega-1})$. Adding T_{22} has cost $O(m_2(n-r_1))$. Conclusion follows for both L_{21} and B_2 from $m_2 \leq m$ and $r_1 \leq r \leq n$. ∎

Using this lemma, let us now estimate $T(m, n, r)$, the number of field operations required by Algorithm `LSP` for input $A \in k^{m \times n}$ of (unknown) rank r. Following [11] we count a comparison with zero as a field operation. Then, when $r = 0$ (that is, $A = 0$) or $m = 1$, we have $T(m, n, r) = O(mn)$. Otherwise, recall that r_1 is the rank of A_1 and let r_2 be the rank of B_2; then, by Lemma 3.2

$$T(m,n,r) = \begin{cases} T(m_1, n, 0) + T(m_2, n, r) + O(mn) & \text{if } r_1 = 0, \\ T(m_1, n, r_1) + O(mnr^{\omega-2}) & \text{if } r_1 = n, \\ T(m_1, n, r_1) + T(m_2, n - r_1, r_2) + O(mnr^{\omega-2}) & \text{if } 0 < r_1 < n. \end{cases}$$

Hence $T(m, n, r)$ is upper bounded by a function $f_n(m, r)$ such that

$$f_n(m,r) = \begin{cases} O(mn) & \text{if } m = 1 \text{ or } r = 0, \\ f_n(\lfloor m/2 \rfloor, r_1) + f_n(\lceil m/2 \rceil, r_2) + O(mnr^{\omega-2}) & \text{otherwise,} \end{cases}$$

for some $r_1, r_2 \geq 0$ with $r = r_1 + r_2$. By [11, §1.3] one has $f_n(m, r) = O(mnr^{\omega-2})$ if $\omega > 2$ and $f_n(m, r) = O(mn \log_2 r)$ if $\omega = 2$. Thus we have shown the following:

Theorem 3.3 *Let $A \in k^{m \times n}$ of (unknown) rank r. Algorithm `LSP` computes an LSP-decomposition of A in time $O(mnr^{\omega-2})$ if $\omega > 2$. If $\omega = 2$, the cost becomes $O(mn \log_2 r)$.*

One could modify easily Algorithm `LSP` so that it returns not only some factors L, S, P but also the rank and the row rank profile of A. For applications (see Section 5), one might need L^{-1} rather than L; modifying Algorithm `LSP` accordingly and obtaining the same complexity bound as in Theorem 3.3 is not hard neither. Here it is interesting to note that if L is striped according to S then its inverse L^{-1} is too. To check this, simply observe that if $j \notin \{i_1, \ldots, i_r\}$ then e_j is the jth column of both L and I_m. Therefore the jth column of L^{-1} is $L^{-1}e_j = e_j$, which means that L^{-1} is striped according to S.

As already observed in [8], once we have an LSP-decomposition of A, an LQUP-decomposition can be deduced immediately. In the next section, we modify Algorithm `LSP` so as to compute directly an LQUP-decomposition.

4 A rank-sensitive LQUP-decomposition algorithm

Instead of computing L, S, P recursively, one can work directly with a corresponding LQUP-decomposition in compressed form. Using the same divide-and-conquer approach as in Section 3, the goal is now to compute matrices M, Q, V, P as in (1), (2), (3).

Recall that $M \in k^{m\times r}$ is in echelon form ($M_{i,j} = 0$ for $i \leq i_j$), that the first r columns of Q are F, and that $V \in k^{r\times r}$ is upper triangular and invertible. Furthermore, an LSP-decomposition of A with L striped according S follows from

$$A = \big([M \,|\, 0] + Q\big)\begin{bmatrix} V \\ \hline 0 \end{bmatrix} P$$

by taking

$$L = [M \,|\, 0]Q^T + I_m \quad \text{and} \quad S = Q\begin{bmatrix} V \\ \hline 0 \end{bmatrix}.$$

Theorem 4.1 *Algorithm* `LQUP` *is correct. Its cost is* $O(mnr^{\omega-2})$ *if* $\omega > 2$ *and* $O(mn \log_2 r)$ *if* $\omega = 2$.

Proof. The proof is similar to that of Theorem 3.3: for correctness, we can proceed by induction on m and, for $m > 1$, study separately three cases "$r_1 = 0$", "$r_1 = n$" and "$0 < r_1 < n$." (Some details about each case are provided in Appendix A.) As for the complexity bound, it is obtained in exactly the same way as in Section 3.2. ∎

In the algorithm below, by "matrix $m \times 0$" we mean a matrix with m empty rows.

Algorithm LQUP(A)
Input: $A \in k^{m\times n}$.
Output: an LQUP-decomposition of A in compressed form M, Q, V, P and its rank r.
if $A = 0$ **then**
  $M, Q, V, P, r :=$ matrix $m \times 0, I_m,$ matrix $0 \times n, I_n, 0$;
else if $m = 1$ **then**
  $j :=$ the index of the first nonzero entry of A;
  $P :=$ the permutation matrix that interchanges $A_{1,1}$ and $A_{1,j}$;
  $M, Q, V, r := 0_{1\times 1}, I_1, AP, 1$;
else
  $m_1, m_2 := \lfloor m/2 \rfloor, \lceil m/2 \rceil$;
  $M_1, Q_1, V_1, P_1, r_1 :=$ LQUP(A_1) with A_1 the first m_1 rows of A;
  if $r_1 = 0$ **then**
    $M_2, Q_2, V_2, P_2, r_2 :=$ LQUP(A_2) with A_2 the last m_2 rows of A;
    $M, Q, V, P, r := \begin{bmatrix} & \\ M_2 & \end{bmatrix}, \begin{bmatrix} & I_{m_1} \\ Q_2 & \end{bmatrix}, V_2, P_2, r_2$;
  else
    $V_{11} :=$ the first r_1 columns of V_1;
    $T_2 := A_2 P_1^T$ with A_2 the last m_2 rows of A;
    $M_{21} := T_{21} V_{11}^{-1}$ with T_{21} the first r_1 columns of T_2;
    if $r_1 = n$ **then**
      $M, Q, V, P, r := \begin{bmatrix} M_1 \\ M_{21} \end{bmatrix}, \begin{bmatrix} Q_1 & \\ & I_{m_2} \end{bmatrix}, V_{11}, P_1, r_1$;
    else
      $B_2 := T_{22} - M_{21} V_{12}$ with V_{12}, T_{22} the last $n - r_1$ columns of V_1, T_2;
      $M_2, Q_2, V_2, P_2, r_2 :=$ LQUP(B_2);
      $M, V, P, r := \begin{bmatrix} M_1 & \\ M_{21} & M_2 \end{bmatrix}, \begin{bmatrix} V_{11} & V_{12}P_2^T \\ & V_2 \end{bmatrix}, \begin{bmatrix} I_{r_1} & \\ & P_2 \end{bmatrix} P_1, r_1 + r_2$;
      $Q := \left[\begin{array}{c|c|c|c} Q_{11} & & Q_{12} & \\ \hline & Q_{21} & & Q_{22} \end{array}\right]$ with $Q_i = [Q_{i1} \,|\, Q_{i2}]$ and Q_{i1} having r_i columns;
    fi;
  fi;
fi;
return M, Q, V, P, r;

5 Some applications of LSP-/LQUP-decomposition

In [8, §4] and [10, §4], several applications of an LSP- or LQUP-decomposition are given: solve the *linear system* $Ax = b$ or detect that it has no solution; find a submatrix of A whose column and row ranks are both equal to r, and, more generally, reorder the rows and columns of A to obtain a matrix with *generic rank profile*; compute left and right *nullspace* bases of A; *diagonalize* A, that is, compute an m by m nonsingular matrix X and an n by n nonsingular matrix Y such that $XAY = \mathrm{diag}(I_r, 0)$; compute various *generalized inverses* (generalized, reflexive generalized [8, p.54] and, when k is the field of complex numbers, Moore-Penrose [7, p.257]).

A consequence of the algorithms of the previous sections is that the complexity of such application problems is now bounded by $O(nmr^{\omega-2})$ or $O(mn \log_2 r)$ instead of $O(nm^{\omega-1})$.

We give some details below.

Generic rank profile. Partition V in (3) as $V = [V_1 \,|\, V_2]$ with $V_1 \in k^{r\times r}$. Then

$$Q^T A P^T = \left[\frac{F^T M + I_r}{G^T M}\right]\left[\, V_1 \mid V_2 \,\right].$$

Since $F^T M + I_r$ is unit lower triangular and V_1 is upper triangular and nonsingular, the first r principal minors of the product $(F^T M + I_r)V_1$ are nonsingular. Hence $Q^T A P^T$ has generic rank profile.

Linear system solving. Here the classic process is as follows. Given $A \in k^{m\times n}$ and $b \in k^n$, first compute an LSP-decomposition of A; then compute the vector $c = L^{-1}b$; conclude that $Ax = b$ has no solution, otherwise solve $Sy = c$ for y and return $x = P^T y$.

Due to the semi-upper triangular shape of S, solving $Sy = c$ has cost $O(r^\omega)$. In order to bound the cost of solving $Lc = b$, observe that

$$L = [M \,|\, 0]Q^T + I_m$$

leads to

$$Q^T L Q = Q^T [M \,|\, 0] + I_m.$$

Now, $W := Q^T L Q$ has the shape

$$W = \begin{bmatrix} W_1 & \\ W_2 & I_{m-r} \end{bmatrix}, \quad \text{with } W_1 \in k^{r\times r} \text{ invertible.}$$

Therefore, by inverting W, one can compute $L^{-1}b = Q\big(W^{-1}(Q^T b)\big)$ with $O(mr^{\omega-1})$ field operations. In conclusion, when $A = LSP$ is given, one can solve $Ax = b$ or decide that no solution exists with $O(mr^{\omega-1})$ extra operations.

Left and right nullspace bases. With V_1, V_2, W_1, W_2 as above, it is not hard to verify that bases of the left and right nullspaces of A are given by, respectively,

$$\begin{bmatrix} -W_2 W_1^{-1} & I_{m-r} \end{bmatrix} Q^T \quad \text{and} \quad P^T \begin{bmatrix} -V_1^{-1} V_2 \\ I_{n-r} \end{bmatrix}.$$

Given $A = LQUP$, computing those two bases costs $O((m+n)r^{\omega-1})$ field operations.

Diagonalizing transforms. Some transforms $X \in k^{m\times m}$ and $Y \in k^{n\times n}$ such that $XAY = \mathrm{diag}(I_r, 0)$ are (see [8, p.53])

$$X = Q^{-1}L^{-1} \quad \text{and} \quad Y = P^T \begin{bmatrix} V_1^{-1} & -V_1^{-1}V_2 \\ & I_{n-r} \end{bmatrix}.$$

Hence, given $A = LQUP$, one can compute them at an extra cost of $O((m+n)r^{\omega-1})$.

Generalized inverses. From the proof of [8, Theorem 3.3], an arbitrary generalized inverse and an arbitrary reflexive inverse can both be deduced from the above X, Y, L, Q, V via some matrix multiplications having cost $O(mnr^{\omega-2})$. From the proof of [8, Theorem 3.4] one can see that, given $A = LQUP$, the cost of the Moore-Penrose inverse is dominated by the cost of multiplying an $n \times r$ matrix by an $r \times m$ matrix, which is $O(mnr^{\omega-2})$ too.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] D. Bini and V.Y. Pan. *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms.* Birkhäuser, Boston, 1994.

[3] J. Bunch and J. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

[4] B. Bürgisser, C. Clausen, and M.A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1997.

[5] J.-G. Dumas, T. Gautier, P. Giorgi, and C. Pernet. Dense linear algebra over finite fields: the FFLAS and FFPACK packages. Technical Report cs.SC/0601133, arXiv, 2006.

[6] J.-G. Dumas, P. Giorgi, and C. Pernet. FFPACK: Finite Field Linear Algebra Package. In J. Gutierrez, editor, *Proc. International Symposium on Symbolic and Algebraic Computation, Santander, Spain*, pages 119–126. ACM Press, 2004.

[7] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[8] O.H. Ibarra, S. Moran, and R. Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *Journal of Algorithms*, 3:45–56, 1982.

[9] C. Pernet. Calcul du polynôme caractéristique sur des corps finis. Master's thesis, Université Joseph Fourier, 2003.

[10] C. Pernet. *Algèbre linéaire exacte efficace: le calcul du polynôme caractéristique*. PhD thesis, Université Joseph Fourier, 2006.

[11] A. Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Swiss Federal Institute of Technology, ETH-Zurich, 2000.

A Some details for the correctness proof of Algorithm LQUP

A.1 Case where $r_1 = 0$

In this case $r = r_2$ and

$$A = \left[\begin{array}{c} 0 \\ \hline A_2 \end{array}\right],$$

which implies

$$AP_2^T = \left[\begin{array}{c} 0 \\ \hline L_2Q_2U_2 \end{array}\right] = \left[\begin{array}{c|c} I_{m_1} & \\ \hline & L_2Q_2 \end{array}\right]\left[\begin{array}{c} 0 \\ \hline U_2 \end{array}\right], \quad L_2Q_2 = [M_2 \,|\, 0] + Q_2, \quad U_2 = \left[\begin{array}{c} V_2 \\ \hline 0 \end{array}\right].$$

Hence

$$A = \left[\begin{array}{c|c} & I_{m_1} \\ \hline [M_2 \,|\, 0] + Q_2 & \end{array}\right]\left[\begin{array}{c} U_2 \\ \hline 0 \end{array}\right] = \left(\underbrace{\left[\begin{array}{c|c} 0 & 0_{m_1\times(m-r_2)} \\ \hline M_2 & 0 \end{array}\right]}_{=:[M\,|\,0]} + \underbrace{\left[\begin{array}{c|c} & I_{m_1} \\ \hline Q_2 & \end{array}\right]}_{=:Q}\right)\left[\begin{array}{c} V_2 \\ \hline 0 \end{array}\right]P_2,$$

and $V := V_2$ and $P := P_2$. The first r columns of Q are $\left[\begin{array}{c} 0 \\ \hline Q_2 \end{array}\right] = [e_{i'_1+m_1}, \ldots, e_{i'_r+m_1}]$ where, by induction, (i'_j) is the row rank profile of A_2. Conclusion follows from the fact that when $r_1 = 0$ one has $i_j = i'_j + m_1$ for $1 \le j \le r$. Similarly, M is of the desired shape because M_2 is, by induction.

A.2 Case where $r_1 > 0$

In this case

$$A = \left[\begin{array}{c} A_1 \\ \hline A_2 \end{array}\right] = \left[\begin{array}{c} L_1Q_1U_1 \\ \hline T_2 \end{array}\right]P_1, \quad L_1Q_1 = [M_1 \,|\, 0] + Q_1, \quad U_1 = \left[\begin{array}{c} V_1 \\ \hline 0 \end{array}\right].$$

Therefore, with $V_1 = [V_{11}\ V_{12}]$ and $T_2 = [T_{21}\ T_{22}]$,

$$\begin{aligned} AP_1^T &= \left[\begin{array}{c|c} [M_1 \,|\, 0] + Q_1 & \\ \hline & I_{m_2} \end{array}\right]\left[\begin{array}{cc} V_{11} & V_{12} \\ 0 & 0 \\ \hline T_{21} & T_{22} \end{array}\right] \\ &= \left[\begin{array}{c|c} [M_1 \,|\, 0] + Q_1 & \\ \hline & I_{m_2} \end{array}\right]\left[\begin{array}{cc|c} I_{r_1} & & \\ & I_{m_1-r_1} & \\ \hline M_{21} & & I_{m_2} \end{array}\right]\left[\begin{array}{cc} V_{11} & V_{12} \\ 0 & 0 \\ \hline 0 & B_2 \end{array}\right], \end{aligned}$$

where $M_{21} = T_{21}V_{11}^{-1}$ and $B_2 = T_{22} - M_{21}V_{12}$.

A.2.1 Case where $r_1 = n$

In this case $r = r_1 = n$ and thus

$$\begin{aligned} A &= \left[\begin{array}{c|c} [M_1 \,|\, 0] + Q_1 & \\ \hline & I_{m_2} \end{array}\right] \left[\begin{array}{cc|c} I_n & & \\ & I_{m_1-n} & \\ \hline M_{21} & & I_{m_2} \end{array}\right] \left[\begin{array}{c} V_{11} \\ 0 \\ \hline 0 \end{array}\right] P_1 \\ &= \left(\underbrace{\left[\begin{array}{c|c} M_1 & 0_{m_1\times(m-n)} \\ \hline M_{21} & 0 \end{array}\right]}_{=:[M\,|\,0]} + \underbrace{\left[\begin{array}{c|c} Q_1 & \\ \hline & I_{m_2} \end{array}\right]}_{=:Q} \right) \left[\begin{array}{c} V_{11} \\ \hline 0 \end{array}\right] P_1, \quad V := V_{11}, \quad P := P_1. \end{aligned}$$

In addition the first r columns of Q are the first r_1 columns of $\left[\begin{array}{c} Q_1 \\ \hline 0 \end{array}\right]$, that is, by induction, $[e_{i'_1}, \ldots, e_{i'_{r_1}}]$ where (i'_j) is the row rank profile of A_1. Since $r_1 = n$, (i'_j) is also the row rank profile (i_j) of A and thus the first r columns of Q are $[e_{i_1}, \ldots, e_{i_r}]$ as wanted. Also, since by induction M_1 has the correct echelon shape, the same holds for M.

A.2.2 Case where $0 < r_1 < n$

Since $B_2 = ([M_2 \,|\, 0] + Q_2)\left[\begin{array}{c} V_2 \\ \hline 0 \end{array}\right] P_2$ we have

$$AP_1^T = \left(\left[\begin{array}{cc|cc} M_1 & 0 & & \\ M_{21} & 0 & M_2 & 0 \end{array}\right] + \left[\begin{array}{cc|cc} Q_1 & & & \\ & & & Q_2 \end{array}\right] \right) \left[\begin{array}{cc} V_{11} & V_{12}P_2^T \\ 0 & 0 \\ \hline & V_2 \\ & 0 \end{array}\right] \left[\begin{array}{c|c} I_{r_1} & \\ \hline & P_2 \end{array}\right].$$

Hence

$$\begin{aligned} A &= \left(\left[\begin{array}{cc|cc} M_1 & 0 & & \\ M_{21} & 0 & M_2 & 0 \end{array}\right] + \left[\begin{array}{cc|cc} Q_{11} & Q_{12} & & \\ & & Q_{21} & Q_{22} \end{array}\right] \right) \left[\begin{array}{cc|c|c} I_{r_1} & & & \\ \hline & & I_{m_1-r_1} & \\ \hline & I_{r_2} & & \\ \hline & & & I_{m_2-r_2} \end{array}\right] \\ &\quad \times \underbrace{\left[\begin{array}{cc} V_{11} & V_{12}P_2^T \\ & V_2 \\ \hline 0 & 0 \\ & 0 \end{array}\right]}_{=:\left[\frac{V}{0}\right]} \underbrace{\left[\begin{array}{c|c} I_{r_1} & \\ \hline & P_2 \end{array}\right] P_1}_{=:P}. \\ &= \left(\underbrace{\left[\begin{array}{cc|c} M_1 & 0 & 0 \\ M_{21} & M_2 & 0 \end{array}\right]}_{=:[M\,|\,0]} + \underbrace{\left[\begin{array}{cc|c|c} Q_{11} & & Q_{12} & \\ & Q_{21} & & Q_{22} \end{array}\right]}_{=:Q} \right) \left[\begin{array}{c} V \\ \hline 0 \end{array}\right] P. \end{aligned}$$

Now, $r = r_1 + r_2$ and the first r columns of Q are $\left[\begin{array}{c|c} Q_{11} & \\ \hline & Q_{21} \end{array}\right]$, that is, by induction,

$$[e_{i'_1}, \ldots, e_{i'_{r_1}} | e_{i''_1+m_1}, \ldots, e_{i''_{r_2}+m_1}]$$

where (i'_j) is the row rank profile of A_1 and (i''_j) is the row rank profile of B_2. Since the row rank profile (i_j) of A satisfies $i_j = i'_j$ for $1 \leq j \leq r_1$ and $i_j = i''_j + m_1$ for $r_1 < j \leq r$, we conclude that he first r columns of Q are $[e_{i_1}, \ldots, e_{i_r}]$ as wanted. Also, $M = \begin{bmatrix} M_1 & \\ M_{21} & M_2 \end{bmatrix}$ has the desired shape, because that is true, by induction, for both M_1 and M_2 and because of the above relationship between (i_j), (i'_j) and (i''_j).