Leading constants of rank deficient Gaussian elimination
Clément Pernet, Hippolyte Signargout, Gilles Villard

To cite this version:
Clément Pernet, Hippolyte Signargout, Gilles Villard. Leading constants of rank deficient Gaussian elimination. 2023. hal-03976168

HAL Id: hal-03976168
https://cnrs.hal.science/hal-03976168
Preprint submitted on 6 Feb 2023

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.
Leading constants of rank deficient Gaussian elimination

Clément Pernet *  Hippolyte Signargout †
Grenoble INP – UGA, LJK  ENS de Lyon, LIP, LJK

Gilles Villard ‡
CNRS, ENS de Lyon, LIP

February 6, 2023

1 Introduction

Rank revealing factorizations play a key role in many applications, especially in producing compact generators for structured matrices [4, 1, 7, 8].

In the context of exact linear algebra, Gaussian elimination based factorizations prevail, since there is no numerical instability issue. We refer to [6, 2, 3, 9, 5] for further details on the various decompositions (LSP, LQUP, PLUQ, CUP, PLE, CRE, etc) and algorithms to compute them.

We focus here on algorithms reducing to fast matrix multiplication, with a rank sensitive cost. For instance, in [6, 2], cost analysis are given in the form $O(mnr^\omega - 2)$ where $\omega$ is a feasible exponent for matrix multiplication.

In order to compare algorithms more precisely, it is essential to estimate the leading constant in the cost of such algorithms. We will denote by $T_{XX}(n)$ the leading term in the cost of running algorithm $XX$ on dimension $n$, i.e. such that the cost is $T_{XX}(n) + o(T_{XX}(n))$. In the following we will assume that the product of two $n \times n$ matrices over a field can be computed in $T_{MM}(n) = C_\omega n^{\omega}$, where $2 < \omega \leq 3$.

In [6, Table 1] and [2], such constants are shown but only under strong genericity assumptions: namely, that the matrices are square with full rank ($m = n = r$) and must also sometimes have generic rank profile.

The purpose of this report is to state leading constants for the costs of the main algorithms when matrices are rank deficient and with no genericity assumption.

Unfortunately, the non-predictable rank distribution among the blocks being processed leads to over-estimate some costs, which prevents us to produce tight constants (i.e. matching the known ones in the generic case).

Following [6, 2, 3], the rank revealing factorizations CUP, PLE, PLUQ, CRE can all be computed using either a uni- [6] or bi-dimensional [2] divide and conquer algorithm.

Recall from [6, Table 1] that the cost for solving $n$ triangular systems of order $m$ is

$$T_{TRSM}(m, n) = \frac{C_\omega}{2^{\omega-1}} m^{\omega-1} n$$

2 Unidimensional recursive algorithms

Most divide and conquer Gaussian elimination follow split one of the two dimension of the work matrix in halves (either the row or column dimension). Refer to [6] for a survey on these algorithms and the related matrix factorizations being produced: CUP, PLE, LSP, LQUP, PLUQ, etc.
Lemma 1. \( T_{\text{uni}}(m, n, r) = \alpha mn r^\omega - 2 + \beta mr^{\omega - 1} \) with

\[
\alpha = \frac{C_\omega}{2 - 2^\omega}, \quad \beta = \max \left( 0, \frac{C_\omega (3 - 2^\omega - 1)}{2^\omega - 1 - 2} \right).
\]

Note that with \((\omega, C_\omega) = (3, 2)\), this becomes \( T_{\text{uni}}(m, n, r) = 2mn r^\omega - 2 - \frac{3}{2} mr^{\omega - 1} \). The constant overshoots by a factor of 2, the actual constant for \( m = n = r \) which is \( 2/3 \).

Proof.

\[
T_{\text{uni}}(m, n, r) = T_{\text{uni}}(m, n, r_1) + T_{\text{uni}}(m - 2, n - r_1, r_2) + \frac{C_\omega}{2^\omega - 1 - 2} \left( \frac{m}{2} - r_1 \right)^{\omega - 1} + C_\omega \frac{m - 1}{2} \left( m - 1 \right)^{r_1^{\omega - 2}}
\]

\[
\leq m \left( \alpha (r_1^{\omega - 2} + r_2^{\omega - 2}) + \frac{C_\omega}{2} \right) + m \left( \beta (r_1^{\omega - 1} + r_2^{\omega - 1}) + \frac{C_\omega (3 - 2^\omega - 1)}{2(2^\omega - 1 - 2)} \right)
\]

since \( a^{\omega - 2} + b^{\omega - 2} \leq 2^3 (a + b)^{\omega - 2} \). For any \( \omega \geq \log_2 3 + 1 \), we have \( 3 - 2^\omega \leq 0 \) and therefore setting \( \beta = 0 \) we get

\[
T_{\text{uni}}(m, n, r) \leq mn r^\omega - 2 \left( 2^\omega a + \frac{C_\omega}{2} \right)
\]

Lemma \( \square \) is satisfied as soon as \( \alpha \geq 2^\omega a + \frac{C_\omega}{2} \) which gives \( \alpha = \frac{C_\omega 2^\omega}{2} \).

When \( \omega \leq 1 + \log_2 3 \), we have

\[
T_{\text{uni}}(m, n, r) \leq mn r^\omega - 2 \left( 2^\omega a + \frac{C_\omega}{2} \right) + mr^{\omega - 1} \left( \beta + \frac{C_\omega (3 - 2^\omega - 1)}{2(2^\omega - 1 - 2)} \right)
\]

Similarly, Lemma \( \square \) is satisfied for \( \alpha = \frac{C_\omega (3 - 2^\omega - 1)}{2} \) and \( \beta = \frac{C_\omega (3 - 2^\omega - 1)}{2(2^\omega - 1 - 2)} \).

\[ \square \]

3 Bi-dimensional recursive algorithms

Alternatively, a bi-dimensional split can be applied, which generates in the general case four recursive calls. Refer to [2] [3] for an algorithm and a survey on the related matrix decompositions: PLUQ, CRE, etc.

Lemma 2. \( T_{\text{bi}}(m, n, r) = \alpha mn r^\omega - 2 + \beta (m + n) r^{\omega - 1} \) with

\[
\alpha = \frac{3^4 - \omega C_\omega}{4(4^\omega - 4)}, \quad \beta = \max \left( 0, \frac{C_\omega (3 - 2^\omega - 1)}{2^\omega - 4} \right).
\]

Proof. Following [2]

\[
T_{\text{bi}}(m, n, r) \leq T_{\text{bi}}(m, n, r_1) + T_{\text{bi}}(m, n - r_1, r_2) + T_{\text{bi}}(m, r_2 - r_1, r_3) + T_{\text{bi}}(m, n - r_2 - r_3, r_4)
\]

\[
+ \frac{C_\omega}{2} \left( \frac{1}{2^\omega - 1 - 2} - 1 \right) \left( mn r^{\omega - 1} + mr^{\omega - 1} + mr^{\omega - 1} \right) + \frac{C_\omega}{4} mn \left( 3r_1^{\omega - 2} + r_2^{\omega - 2} + r_3^{\omega - 2} \right)
\]

\[
\leq mn r^\omega - 2 \left( 3^4 - \omega a + 4^\omega C_\omega \right)
\]

\[
+ (m + n) r^{\omega - 1} \left( \frac{C_\omega (3 - 2^\omega - 1)}{2(2^\omega - 4)} \right) \left( mn r^{\omega - 1} + mr^{\omega - 1} + nr^{\omega - 1} \right)
\]

\[
+ \beta \left( mn + r \right) \left( r_1^{\omega - 1} + r_2^{\omega - 1} + r_3^{\omega - 1} + r_4^{\omega - 1} \right)
\]

For any \( \omega \geq \log_2 3 + 1 \), we have \( 3 - 2^\omega \leq 0 \) and therefore setting \( \beta = 0 \) we get

\[
T_{\text{bi}}(m, n, r) \leq mn r^\omega - 2 \left( 3^4 - \omega a + 4^\omega C_\omega \right).
\]

Lemma \( \square \) is satisfied as soon as \( \alpha \geq \frac{3^4 - \omega a + 4^\omega C_\omega}{4(4^\omega - 4)} \) which gives \( \alpha = \frac{3^4 - \omega a + 4^\omega C_\omega}{4(4^\omega - 4)} \).

When \( \omega \leq \log_2 3 + 1 \), then the same value for \( \alpha \) and \( \beta = \frac{C_\omega (3 - 2^\omega - 1)}{2^\omega - 4} \) satisfy Lemma \( \square \)
References


