



**HAL**  
open science

# 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 \*  
Grenoble INP – UGA, LJK

Hippolyte Signargout †  
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 overestimate 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

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

## 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.

---

\*clement.pernet@univ-grenoble-alpes.fr

†hippolyte.signargout@ens-lyon.fr

‡gilles.villard@ens-lyon.fr

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

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

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

*Proof.*

$$\begin{aligned} T_{\text{uni}}(m, n, r) &= T_{\text{uni}}\left(\frac{m}{2}, n, r_1\right) + T_{\text{uni}}\left(\frac{m}{2}, n - r_1, r_2\right) + \frac{C_\omega}{2^{\omega-1}-2} \frac{m}{2} r_1^{\omega-1} + C_\omega \frac{m}{2} (n - r_1) r_1^{\omega-2} \\ &\leq mn \left( \frac{\alpha}{2} (r_1^{\omega-2} + r_2^{\omega-2}) + \frac{C_\omega}{2} \right) + m \left( \frac{\beta}{2} (r_1^{\omega-1} + r_2^{\omega-1}) + \frac{C_\omega}{2} \left( \frac{3-2^{\omega-1}}{2^{\omega-1}-2} \right) r_1^{\omega-1} \right) \end{aligned}$$

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

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

Lemma 1 is satisfied as soon as  $\alpha \geq 2^{2-\omega} \alpha + \frac{C_\omega}{2}$  which gives  $\alpha = \frac{C_\omega}{2-2^{3-\omega}}$ .

When  $\omega \leq 1 + \log_2 3$ , we have

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

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

### 3 Bi-dimensional recursive algorithms

Alternatively, a bi-dimensionnal 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

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

*Proof.* Following [2]

$$\begin{aligned} T_{\text{bi}}(m, n, r) &\leq T_{\text{bi}}\left(\frac{m}{2}, \frac{n}{2}, r_1\right) + T_{\text{bi}}\left(\frac{m}{2} - r_1, \frac{n}{2}, r_2\right) + T_{\text{bi}}\left(\frac{m}{2}, \frac{n}{2} - r_1, r_3\right) + T_{\text{bi}}\left(\frac{m}{2}, \frac{n}{2} - r_2 - r_3, r_4\right) \\ &\quad + \frac{C_\omega}{2} \left( \frac{1}{2^{\omega-1}-2} - 1 \right) ((m+n)r_1^{\omega-1} + mr_2^{\omega-1} + nr_3^{\omega-1}) + \frac{C_\omega}{4} mn (3r_1^{\omega-2} + r_2^{\omega-2} + r_3^{\omega-2}) \\ &\leq mn r^{\omega-2} \left( \frac{4^{3-\omega} \alpha + 3^{4-\omega} C_\omega}{4} \right) \\ &\quad + (m+n)r^{\omega-1} \left( \frac{C_\omega(3-2^{\omega-1})}{2(2^\omega - 4)} \right) ((m+n)r_1^{\omega-1} + mr_2^{\omega-1} + nr_3^{\omega-1}) \\ &\quad + \frac{\beta}{2} (m+n) (r_1^{\omega-1} + r_2^{\omega-1} + r_3^{\omega-1} + r_4^{\omega-1}) \end{aligned}$$

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

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

Lemma 2 is satisfied as soon as  $\alpha \geq \frac{4^{3-\omega} \alpha + 3^{4-\omega} C_\omega}{4}$ , which gives  $\alpha = \frac{3^{4-\omega} C_\omega}{(4-4^{3-\omega})}$

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 2.  $\square$

## References

- [1] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A. J. van der Veen. Fast stable solver for sequentially semi-separable linear systems of equations. In Sartaj Sahni, Viktor K. Prasanna, and Uday Shukla, editors, *High Performance Computing — HiPC 2002*, pages 545–554, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [2] Jean-Guillaume Dumas, Clément Pernet, and Ziad Sultan. Simultaneous computation of the row and column rank profiles. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, ISSAC '13, page 181–188, New York, NY, USA, 2013. Association for Computing Machinery.
- [3] Jean-Guillaume Dumas, Clément Pernet, and Ziad Sultan. Fast computation of the rank profile matrix and the generalized bruhat decomposition. *Journal of Symbolic Computation*, 83:187 – 210, 2017. Special issue on the conference ISSAC 2015: Symbolic computation and computer algebra.
- [4] Yuli Eidelman and I. Gohberg. On generators of quasiseparable finite block matrices. *Calcolo*, 42:187–214, 12 2005.
- [5] Oscar H. Ibarra, Shlomo Moran, and Roger Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *Journal of Algorithms*, 3(1):45–56, March 1982.
- [6] 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.
- [7] P. Martinsson. A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix. *SIAM J. Matrix Analysis Applications*, 32:1251–1274, 10 2011.
- [8] Clément Pernet and Arne Storjohann. Time and space efficient generators for quasiseparable matrices. *Journal of Symbolic Computation*, 85:224 – 246, 2018. Special issue on the 41th International Symposium on Symbolic and Algebraic Computation (ISSAC'16).
- [9] Arne Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Institut für Wissenschaftliches Rechnen, ETH-Zentrum, Zürich, Switzerland, November 2000.