CUR LRA at Sublinear Cost Based on Volume Maximization

Qi Luan[1] and Victor Y. Pan[2,3](✉)

[1] Mathematics, The Graduate Center of the City University of New York, New York, NY 10036, USA
qi_luan@yahoo.com

[2] Computer Science and Mathematics,The Graduate Center of the City University of New York, New York, NY 10036, USA

[3] Computer Science, Lehman College of the City University of New York, Bronx, NY 10468, USA
victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan/

Abstract. A matrix algorithm runs at *sublinear cost* if it uses much fewer memory cells and arithmetic operations than the input matrix has entries. Such algorithms are indispensable for Big Data Mining and Analysis, where input matrices are so immense that one can only access a small fraction of all their entries. Typically, however, such matrices admit their Low Rank Approximation *(LRA)*, which one can access and process at sublinear cost. Can, however, we compute LRA at sublinear cost? Adversary argument shows that no algorithm running at sublinear cost can output accurate LRA of worst case input matrices or even of the matrices of small families of our Appendix A, but we prove that some sublinear cost algorithms output a reasonably close LRA of a matrix W if (i) this matrix is sufficiently close to a low rank matrix or (ii) it is a Symmetric Positive Semidefinite *(SPSD)* matrix that admits LRA. In both cases supporting algorithms are deterministic and output LRA in its special form of CUR LRA, particularly memory efficient. The design of our algorithms and the proof of their correctness rely on the results of extensive previous study of CUR LRA in Numerical Linear Algebra using volume maximization. In case (i) we apply Cross-Approximation *(C-A)* iterations, running at sublinear cost and computing accurate LRA worldwide for more than a decade. We provide the first formal support for this long-known empirical efficiency assuming non-degeneracy of the initial submatrix of at least one C-A iteration. We cannot ensure non-degeneracy at sublinear cost for a worst case input but prove that it holds with a high probability *(whp)* for any initialization in the case of a random or randomized input. Empirically we can replace randomization with sparse multiplicative preprocessing of an input matrix, performed at sublinear cost. In case (ii) we make no additional assumptions about the input class of SPSD matrices admitting LRA or about initialization of our sublinear cost algorithms for CUR LRA, which promise to be practically valuable. We hope that proper combination of our deterministic techniques with randomized LRA methods, popular among Computer Science researchers, will lead them to further progress in LRA.

D. Slamanig et al. (Eds.): MACIS 2019, LNCS 11989, pp. 105–121, 2020.
https://doi.org/10.1007/978-3-030-43120-4_10

Keywords: Low Rank Approximation (LRA) · CUR LRA · Sublinear cost · Symmetric Positive Semidefinite (SPSD) matrices · Cross-Approximation (C-A) · Maximal volume

2000 Math. Subject Classification: 65Y20 · 65F30 · 68Q25 · 15A52

1 Introduction

1.1. LRA Problem. An $m \times n$ matrix W admits its close approximation of rank at most r if and only if the matrix W has *numerical rank* at most r (and then we write $\text{nrank}(W) \leq r$), that is,

$$W = AB + E, \ ||E||/||W|| \leq \epsilon, \tag{1.1}$$

for $A \in \mathbb{C}^{m \times r}$, $B \in \mathbb{C}^{r \times n}$, a matrix norm $|| \cdot ||$, and a small tolerance ϵ. Such an LRA approximates the mn entries of W by using $(m + n)r$ entries of A and B. This is a crucial benefit in applications of LRA to Big Data Mining and Analysis, where the size mn of an input matrix is usually immense, and one can only access a tiny fraction of its mn entries. Quite typically, however, such matrices admit LRA of (1.1) where $(m+n)r \ll mn$. (Hereafter $a \ll b$ and $b \gg a$ mean that the ratio $|a/b|$ is small in context.)

Can we, however, compute close LRA *at sublinear cost*, that is, by using much fewer memory cells and flops than an input matrix has entries? Based on adversary argument one can prove that no algorithm running at sublinear cost can output close LRA of the worst case inputs and even of the matrices of small families of our Appendix A, but for more than a decade Cross-Approximation (C-A) iterations, running at sublinear cost, have been routinely computing close LRA worldwide. Moreover they output LRA in its special form of CUR LRA (see Sect. 2), which is particularly memory efficient and is defined by a proper choice of a submatrix G of W, said to be a generator of CUR LRA or a *CUR generator*.

1.2. Our First Main Result. The main result of Part I of our paper, made up of Sects. 2–5, provides partial formal support for this empirical phenomenon.

Let us elaborate. Let $\sigma_j(M)$ denote the jth largest singular value of a matrix M, which is the minimal distance from M to a matrix of rank $j + 1$ in spectral norm. Suppose that C-A iterations are applied to an $m \times n$ matrix W that admits a sufficiently close LRA (1.1). Let W_i and V_i denote the input and output submatrices of W at the ith C-A iteration for $i = 1, 2, \ldots$ and let $||\cdot||$ denote the spectral or Frobenius matrix norm. Then we prove (see Corollary 3 and Remark 3) that the approximation error norm $||W - V_{i+1}||$ is within a factor f from optimal, which is reasonably bounded unless the ratio $\sigma_r(W_i)/\sigma_r(W)$ is small.

Our proof relies on Theorems 1 and 2, recalled from [OZ18], which extend long study traced back to [CI94, GE96, GTZ97, GT01] and which bound the output errors of CUR LRA in term of maximization of the volume $v_2(G)$ or r-projective volume $v_{2,r}(G)$ of a CUR generator G (see Definition 1 for these concepts).

The ratio $\sigma_r(W_i)/\sigma_r(W)$ is small where one applies C-A iterations to a worst case input matrix, but one can prove that it is not small whp where an input matrix of small numerical rank is random or randomized by means of its preand post-multiplying by random multipliers. Empirically the ratio tends to be not small even where an input matrix of small numerical rank is pre-processed with any fixed rather than random orthogonal multipliers, and in particular at sublinear cost for proper sparse multipliers. The above error factor f can be considered a price for obtaining CUR LRA at sublinear cost, but if the ratio $\sigma_{r+1}(W)/\sigma_r(W)$ is small enough, we can iteratively refine LRA at sublinear cost by means of our algorithms of [PLa].

1.3. Our Results About CUR LRA of SPSD Matrices. Our novel sublinear cost algorithm computes reasonably close CUR LRA of any SPSD matrix admitting LRA. Then again we devise and analyze our algorithm based on the cited link of the error bounds of an output CUR LRA and maximization of the volume or r-projective volume of a CUR generator, and we can reapply our comments on deviation from optimum and iterative refinement of the output.

1.4. Earlier Works. Our results of Part I appeared in [PLSZ16, Section 5] and [PLSZ17, Part II] together with various results on LRA of random input matrices.[1] Our progress in Part II has been inspired by the results of [OZ18] and [CKM19]. Section 1.4 of [LPa] covers relevant earlier works in more details.

1.5. Organization of Our Paper. We define CUR LRA and C-A iterations in the next section. We devote Sect. 3 to background material on matrix volumes, their maximization and its impact on LRA. In Sect. 4 we recall C-A iterations and in Sect. 5 prove that they output reasonably close LRA of a matrix having sufficiently low numerical rank. These sections make up Part I of our paper, while Sects. 6–8 make up its Part II. In Sect. 6 we state our main results for SPSD inputs. We prove the correctness of our algorithms in Sect. 7 and [LPa] and estimate their complexity in Sect. 8. In the Appendix we recall the relevant definitions and auxiliary results and specify small matrix families that are hard for LRA at sublinear cost.

Part I. CUR LRA by Means of C-A Iterations

2 Background: CUR LRA

We use basic definitions for matrix computations recalled in Appendix B. We simplify our presentation by confining it to the case of real matrices, but the extension to the case of complex matrices is straightforward.

CUR LRA of a matrix W of numerical rank at most r is defined by three matrices C, U, and R, with C and R made up of l columns and k rows of W,

[1] The papers [PLSZ16], unsuccessfully submitted to ACM STOC 2017 and widely circulated at that time, and [PLSZ17] provided the first formal support for LRA at sublinear cost, which they called "superfast" LRA. Their approach has extended to LRA the earlier study in [PQY15, PZ17a], and [PZ17b] of randomized Gaussian elimination with no pivoting and other fundamental matrix computations. It was followed by sublinear cost randomized LRA algorithms of [MW17].

respectively, $U \in \mathbb{C}^{l\times k}$ said to be the *nucleus* of CUR LRA,

$$0 < r \leq k \leq m,\ r \leq l \leq n,\ kl \ll mn, \tag{2.1}$$

$$W = CUR + E,\ \text{and}\ ||E||/||W|| \leq \epsilon,\ \text{for a small tolerance}\ \epsilon > 0. \tag{2.2}$$

CUR LRA is a special case of LRA of (1.1) where $k = l = r$ and, say, $A = LU$, $B = R$. Conversely, given LRA of (1.1) one can compute CUR LRA of (2.2) at sublinear cost (see [PLa] and [PLSZa]).

Define a *canonical* CUR LRA as follows.

(i) Fix two sets of columns and rows of W and define its two submatrices C and R made up of these columns and rows, respectively.
(ii) Define the $k \times l$ submatrix $W_{k,l}$ made up of all common entries of C and R, and call it a *CUR generator*.
(iii) Compute its rank-r truncation $W_{k,l,r}$ by setting to 0 all its singular values, except for the r largest ones.
(iv) Compute the Moore–Penrose pseudo inverse $U =: W^+_{k,l,r}$ and call it the *nucleus* of CUR LRA of the matrix W (cf. [DMM08, OZ18]); see an alternative choice of a nucleus in [MD09]).

$W_{r,r} = W_{r,r,r}$, and if a CUR generator $W_{r,r}$ is nonsingular, then $U = W^{-1}_{r,r}$.

3 Background: Matrix Volumes

3.1 Definitions and Hadamard's Bound

Definition 1. *For three integers k, l, and r such that $1 \leq r \leq \min\{k, l\}$, define the* volume $v_2(M) := \prod_{j=1}^{\min\{k,l\}} \sigma_j(M)$ *and* r-projective volume $v_{2,r}(M) := \prod_{j=1}^{r} \sigma_j(M)$ *of a $k \times l$ matrix M such that $v_{2,r}(M) = v_2(M)$* if $r = \min\{k, l\}$, *$v_2^2(M) = \det(MM^*)$ if $k \geq l$; $v_2^2(M) = \det(M^*M)$ if $k \leq l$, $v_2^2(M) = |\det(M)|^2$ if $k = l$.*

Definition 2. *The volume of a $k \times l$ submatrix $W_{\mathcal{I},\mathcal{J}}$ of a matrix W is h-*maximal *over all $k \times l$ submatrices if it is maximal up to a factor of h. The volume $v_2(W_{\mathcal{I},\mathcal{J}})$ is* column-wise *(resp.* row-wise*) h-maximal if it is h-maximal in the submatrix $W_{\mathcal{I},:}$ (resp. $W_{:,\mathcal{J}}$). The volume of a submatrix $W_{\mathcal{I},\mathcal{J}}$ is* column-wise *(resp.* row-wise*)* locally *h-maximal if it is h-maximal over all submatrices of W that differ from the submatrix $W_{\mathcal{I},\mathcal{J}}$ by a single column (resp. single row). Call volume* (h_c, h_r)-maximal *if it is both column-wise h_c-maximal and row-wise h_r-maximal. Likewise define* locally (h_c, h_r)-maximal *volume. Write* maximal *instead of 1-maximal and $(1, 1)$-maximal in these definitions. Extend all of them to r-projective volumes.*

For a $k \times l$ matrix $M = (m_{ij})_{i,j=1,1}^{k,l}$ write $\mathbf{m}_j := (m_{ij})_{i=1}^{k}$ and $\bar{\mathbf{m}}_i := ((m_{ij})_{j=1}^{l})^*$ for all i and j. For $k = l = r$ recall *Hadamard's bound*

$$v_2(M) = |\det(M)| \leq \min\ \{\prod\nolimits_{j=1}^{r} ||\mathbf{m}_j||,\ \prod\nolimits_{i=1}^{r} ||\bar{\mathbf{m}}_j^*||,\ r^{r/2} \max\nolimits_{i,j=1}^{r} |m_{ij}|^r\}.$$

3.2 The Impact of Volume Maximization on CUR LRA

The estimates of the two following theorems in the Chebyshev matrix norm $||\cdot||_C$ increased by a factor of $\sqrt{mn}$ turn into estimates in the Frobenius norm $||\cdot||_F$ (see (B.3)).

Theorem 1 [OZ18].[2] *Suppose $r := \min\{k, l\}$, $W_{\mathcal{I},\mathcal{J}}$ is the $k\times l$ CUR generator, $U = W^+_{\mathcal{I},\mathcal{J}}$ is the nucleus of a canonical CUR LRA of an $m \times n$ matrix W, $E = W - CUR$, $h \geq 1$, and the volume of $W_{\mathcal{I},\mathcal{J}}$ is locally h-maximal, that is,*

$$h\ v_2(W_{\mathcal{I},\mathcal{J}}) = \max_B v_2(B)$$

where the maximum is over all $k \times l$ submatrices B of the matrix W that differ from $W_{\mathcal{I},\mathcal{J}}$ in at most one row and/or column. Then

$$||E||_C \leq h\ f(k,l)\ \sigma_{r+1}(W) \ \text{ for } \ f(k,l) := \sqrt{\frac{(k+1)(l+1)}{|l-k|+1}}.$$

Theorem 2 [OZ18]. *Suppose that $W_{k,l} = W_{\mathcal{I},\mathcal{J}}$ is a $k \times l$ submatrix of an $m \times n$ matrix W, $U = W^+_{k,l,r}$ is the nucleus of a canonical CUR LRA of W, $E = W - CUR$, $h \geq 1$, and the r-projective volume of $W_{\mathcal{I},\mathcal{J}}$ is locally h-maximal, that is,*

$$h\ v_{2,r}(W_{\mathcal{I},\mathcal{J}}) = \max_B v_{2,r}(B)$$

where the maximum is over all $k \times l$ submatrices B of the matrix W that differ from $W_{\mathcal{I},\mathcal{J}}$ in at most one row and/or column. Then

$$||E||_C \leq h\ f(k,l,r)\ \sigma_{r+1}(W) \ \text{ for } \ f(k,l,r) := \sqrt{\frac{(k+1)(l+1)}{(k-r+1)(l-r+1)}}.$$

Corollary 1. *Suppose that $BW = (BU|BV)$ for a nonsingular matrix B and that the submatrix U is h-maximal in the matrix $W = (U|V)$. Then the submatrix BU is h-maximal in the matrix BW.*

Remark 1. Theorems 1 and 2 have been stated in [OZ18] under assumptions that the matrix $W_{\mathcal{I},\mathcal{J}}$ has (globally) h-maximal volume or r-projective volume, respectively, but their proofs in [OZ18] support the above extensions to the case of locally maximal volume and r-projective volume.

4 C-A Iterations

C-A iterations recursively apply two auxiliary Subalgorithms $\mathcal{A}$ and $\mathcal{B}$ (see Algorithm 1).

[2] The theorem first appeared in [GT01, Corollary 2.3] in the special case where $k = l = r$ and $m = n$.

Given a 4-tuple of integers k, l, p, and q such that $r \le k \le p$ and $r \le l \le q$ subalgorithm $\mathcal{A}$ is applied to a $p \times q$ matrix and computes its $k \times l$ submatrix whose volume or projective volume is maximal up to a fixed factor $h \ge 1$ among all its $k \times l$ submatrices. For simplicity first consider the case where $k = l = p = q = r$ (see Fig. 1, borrowed from [PLSZa]).

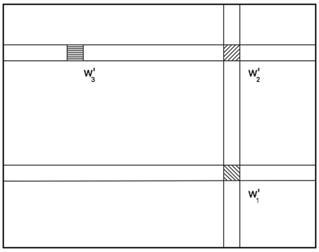

Fig. 1. The three successive C-A steps output three striped matrices.

Subalgorithm $\mathcal{B}$ verifies whether the error norm of the CUR LRA built on a fixed CUR generator is within a fixed tolerance τ (see [PLa] for some verification recipes).

5 CUR LRA by Means of C-A Iterations

We can apply C-A steps by choosing deterministic algorithms of [GE96] for Subalgorithm $\mathcal{A}$. In this case mq and pn memory cells and $O(mq^2)$ and $O(p^2n)$ flops are involved in "vertical" and "horizontal" C-A iterations, respectively. They run at sublinear cost if $p^2 = o(m)$ and $q^2 = o(n)$ and output submatrices having h-maximal volumes for h being a low degree polynomial in $m + n$. Every iteration outputs a matrix that has locally h-maximal volume in a "vertical" or "horizontal" submatrix, and the hope is to obtain globally $\bar{h}$-maximal submatrix (for reasonably bounded $\bar{h}$) when maximization is performed recursively in alternate directions.

Of course, the contribution of C-A step is nil where it is applied to a $p \times q$ input whose volume is 0 or nearly vanishes compared to the target maximum, but the consistent success of C-A iterations in practice suggests that in a small number of loops such a degeneration is regularly avoided.

In the next subsection we show that already two successive C-A iterations output a CUR generator having h-maximal volume (for any $h > 1$) if these iterations begin at a $p \times q$ submatrix of W that shares its rank $r > 0$ with W. By continuity of the volume the result is extended to small perturbations of such matrices within a norm bound estimated in Theorem 13. In Sect. 5.2 we extend these results to the case where r-projective volume rather than the volume of a CUR generator is maximized. (Theorem 2 shows benefits of such a maximization.) In Sect. 5.3 we summarize our study in this section and comment on the estimated and empirical performance of C-A iterations.

5.1 Volume of the Output of a C-A Loop

By comparing SVDs of the matrices W and W^+ obtain the following lemma.

Algorithm 1. C-A Iterations

Input: $W \in \mathbb{C}^{m\times n}$, four positive integers r, k, l, and ITER; a number $\tau > 0$.

Output: A CUR LRA of W with an error norm at most τ or FAILURE.

Initialization: Fix a submatrix W_0 made up of l columns of W and obtain an initial set $\mathcal{I}_0$.

Computations:
for $i = 1, 2, \dots,$ ITER **do**
  if i is even **then**
    "Horizontal" C-A step:
    1. Let $R_i := W_{\mathcal{I}_{i-1},:}$ be a $p \times n$ submatrix of W.
    2. Apply Subalgorithm $\mathcal{A}$ for $q = n$ to R_i and obtain a $k \times l$ submatrix $W_i = W_{\mathcal{I}_{i-1},\mathcal{J}_i}$.
  else
    "Vertical" C-A step:
    1. Let $C_i := W_{:,\mathcal{J}_{i-1}}$ be an $m \times q$ submatrix of W.
    2. Apply Subalgorithm $\mathcal{A}$ for $p = m$ to C_i and obtain a $k \times l$ submatrix $W_i = W_{\mathcal{I}_i,\mathcal{J}_{i-1}}$.
  end if
  Apply subalgorithm $\mathcal{B}$ and obtain E, the error bound of CUR LRA built on the generator W_i.
  if $E \le \tau$ **then**
    return CUR LRA built on the generator W_i.
  end if
end for
return Failure

Lemma 1. *$\sigma_j(W)\sigma_{\mathrm{rank}(W)+1-j}(W^+) = 1$ for all matrices W and all subscripts j, $j \le \mathrm{rank}(W)$.*

Corollary 2. *$v_2(W)v_2(W^+) = 1$ and $v_{2,r}(W)v_{2,r}(W_r^+) = 1$ for all matrices W of full rank and all integers r such that $1 \le r \le \mathrm{rank}(W)$.*

We are ready to prove that a $k \times l$ submatrix of rank r that has (h, h')-locally maximal nonzero volume in a rank-r matrix W has hh'-maximal volume globally in W, that is, over all $k \times l$ submatrices of W.

Theorem 3. *Suppose that the volume of a $k \times l$ submatrix $W_{\mathcal{I},\mathcal{J}}$ is nonzero and (h, h')-maximal in a matrix W for $h \ge 1$ and $h' \ge 1$ where $\mathrm{rank}(W) = r = \min\{k, l\}$. Then this volume is hh'-maximal over all its $k \times l$ submatrices of the matrix W.*

Proof. The matrix $W_{\mathcal{I},\mathcal{J}}$ has full rank because its volume is nonzero.

Fix any $k \times l$ submatrix $W_{\mathcal{I}',\mathcal{J}'}$ of the matrix W, recall that $W = CUR$, and obtain that

$$W_{\mathcal{I}',\mathcal{J}'} = W_{\mathcal{I}',\mathcal{J}} W^+_{\mathcal{I},\mathcal{J}} W_{\mathcal{I},\mathcal{J}'}.$$

If $k \leq l$, then first apply claim (iii) of Theorem 14 for $G := W_{\mathcal{I}',\mathcal{J}}$ and $H := W^+_{\mathcal{I},\mathcal{J}}$; then apply claim (i) of that theorem for $G := W_{\mathcal{I}',\mathcal{J}} W^+_{\mathcal{I},\mathcal{J}}$ and $H := W_{\mathcal{I},\mathcal{J}'}$ and obtain that

$$v_2(W_{\mathcal{I}',\mathcal{J}}) = v_2(W_{\mathcal{I}',\mathcal{J}} W^+_{\mathcal{I},\mathcal{J}} W_{\mathcal{I},\mathcal{J}'}) \leq v_2(W_{\mathcal{I}',\mathcal{J}}) v_2(W^+_{\mathcal{I},\mathcal{J}}) v_2(W_{\mathcal{I},\mathcal{J}'}).$$

If $k > l$ deduce the same bound by applying the same argument to the matrix equation

$$W^T_{\mathcal{I}',\mathcal{J}'} = W^T_{\mathcal{I},\mathcal{J}'} W^{+T}_{\mathcal{I},\mathcal{J}} W^T_{\mathcal{I}',\mathcal{J}}.$$

Combine this bound with Corollary 2 for W replaced by $W_{\mathcal{I},\mathcal{J}}$ and deduce that

$$v_2(W_{\mathcal{I}',\mathcal{J}'}) = v_2(W_{\mathcal{I}',\mathcal{J}} W^+_{\mathcal{I},\mathcal{J}} W_{\mathcal{I},\mathcal{J}'}) \leq v_2(W_{\mathcal{I}',\mathcal{J}}) v_2(W_{\mathcal{I},\mathcal{J}'}) / v_2(W_{\mathcal{I},\mathcal{J}}). \quad (5.1)$$

Recall that the matrix $W_{\mathcal{I},\mathcal{J}}$ is (h, h')-maximal and conclude that

$$h v_2(W_{\mathcal{I},\mathcal{J}}) \geq v_2(W_{\mathcal{I},\mathcal{J}'}) \text{ and } h' v_2(W_{\mathcal{I},\mathcal{J}}) \geq v_2(W_{\mathcal{I}',\mathcal{J}}).$$

Substitute these inequalities into the above bound on the volume $v_2(W_{\mathcal{I}',\mathcal{J}'})$ and obtain that $v_2(W_{\mathcal{I}',\mathcal{J}'}) \leq hh' v_2(W_{\mathcal{I},\mathcal{J}})$.

5.2 From Maximal Volume to Maximal r-Projective Volume

Recall that the CUR LRA error bound of Theorem 1 is strengthened when we shift to Theorem 2, that is, maximize r-projective volume for $r < k = l$ rather than the volume. Next we reduce maximization of r-projective volume of a CUR generators to volume maximization.

Corollary 1 implies the following lemma.

Lemma 2. *Let M and N be a pair of $k \times l$ submatrices of a $k \times n$ matrix and let Q be a $k \times k$ unitary matrix. Then $v_2(M)/v_2(N) = v_2(QM)/v_2(QN)$, and if $r \leq \min\{k, l\}$ then also $v_{2,r}(M)/v_{2,r}(N) = v_{2,r}(QM)/v_{2,r}(QN)$.*

The submatrices R' and $\begin{pmatrix} R' \\ O \end{pmatrix}$ of R of Algorithm 2 have maximal volume and maximal r-projective volume in the matrix R, respectively, by virtue of Theorem 14 and because $v_2(R) = v_{2,r}(R) = v_{2,r}(R')$. Therefore the submatrix $W_{:,\mathcal{J}}$ has maximal r-projective volume in the matrix W by virtue of Lemma 2.

Remark 2. By transposing a horizontal input matrix W and interchanging the integers m and n and the integers k and l we extend the algorithm to computing a $k \times l$ submatrix of maximal or nearly maximal r-projective volume in an $m \times l$ matrix of rank r.

Algorithm 2. From maximal volume to maximal r-projective volume

Input: Four integers k, l, n, and r such that $0 < r \leq k \leq n$ and $r \leq l \leq n$; a $k \times n$ matrix W of rank r; a black box algorithm that finds an $r \times l$ submatrix having locally maximal volume in an $r \times n$ matrix of full rank r.
Output: A column set $\mathcal{J}$ such that $W_{:,\mathcal{J}}$ has maximal r-projective volume in W.
Computations:

1. Compute a rank-revealing QRP factorization $W = QRP$, where Q is a unitary matrix, P is a permutation matrix, $R = \begin{pmatrix} R' \\ O \end{pmatrix}$, and R' is an $r \times n$ matrix. (See [GL13, Sections 5.4.3 and 5.4.4] and [GE96].)
2. Compute an $r \times l$ submatrix $R'_{:,\mathcal{J}}$ of R' having maximal volume.

return $\mathcal{J}'$ such that $P : \mathcal{J}' \longrightarrow \mathcal{J}$.

5.3 Complexity and Accuracy of a Two-Step C-A Loop

The following theorem summarizes our study in this section.

Theorem 4. *Given five integers k, l, m, n, and r such that $0 < r \leq k \leq m$ and $r \leq l \leq n$, suppose that two successive C-A steps (say, based on the algorithms of [GE96]) combined with Algorithm 2 have been applied to an $m \times n$ matrix W of rank r and have output $k \times l$ submatrices W'_1 and $W'_2 = W_{\mathcal{I}_2,\mathcal{J}_2}$ with nonzero r-projective column-wise locally h-maximal and nonzero r-projective row-wise locally h'-maximal volumes, respectively. Then the submatrix W'_2 has $h'h$-maximal r-projective volume in the matrix W.*

By combining Theorems 1, 2, and 4 we obtain the following corollary.

Corollary 3. *Under the assumptions of Theorem 4 apply a two-step C-A loop to an $m \times n$ matrix W of rank r and suppose that both its C-A steps output $k \times l$ submatrices having nonzero r-projective column-wise and row-wise locally h-maximal volumes (see Remark 3 below). Build a canonical CUR LRA on a CUR generator $W'_2 = W_{k,l}$ of rank r output by the second C-A step. Then*

(i) the computation of this CUR LRA by using the auxiliary algorithms of [GE96] involves $(m+n)r$ memory cells and $O((m+n)r^2)$ flops[3] and
(ii) the error matrix E of the output CUR LRA satisfies the bound $||E||_C \leq g(k,l,r)\ \bar{h}\ \sigma_{r+1}(W)$ for $\bar{h}$ of Theorem 4 and $g(k,l,r)$ denoting the functions $f(k,l)$ of Theorem 1 or $f(k,l,r)$ of Theorem 2. In particular $||E||_C \leq 2hh'\sigma_2(W)$ for $k = l = r = 1$.

Remark 3. Theorem 13 enables us to extend Theorem 4 and Corollary 3 to the case of an input matrix W of numerical rank r if the input matrix of any C-A

[3] For $r = 1$ an input matrix turns into a vector of dimension m or n, and then we compute its absolutely maximal coordinate just by applying $m-1$ or $n-1$ comparisons, respectively (cf. [O17]).

step shares its numerical rank with W. This is fulfilled whp for a random matrix W that admits LRA (see our full paper, arXiv:1907.10481).

Part II. CUR LRA for SPSD Matrices

6 CUR LRA of SPSD Matrices: Two Main Results

For SPSD matrices we can a little improve our estimates of Theorem 13 by applying Wielandt–Hoffman theorem (see [GL13, Theorem 8.6.4]), but we are going to compute reasonably close CUR LRA of an SPSD matrix at sublinear cost with no restriction on its distance from a low rank matrix.

Theorem 5 (Main Result 1). *Suppose that $A \in \mathbb{R}^{n\times n}$ is an SPSD matrix, r and n are two positive integers, $r < n$, ξ is a positive number, and $\mathcal{I}$ is the output of Algorithm 6. Write $C := A_{:,\mathcal{I}}$, $U := A^{-1}_{\mathcal{I},\mathcal{I}}$, and $R := A_{\mathcal{I},:}$. Then*

$$||A - CUR||_C \le (1+\xi)(r+1)\sigma_{r+1}(A). \tag{6.1}$$

Furthermore Algorithm 6 runs at an arithmetic cost in $O(nr^4 \log r)$.

Theorem 6 (Main Result 2, *proven in [LPa], due to size limitation for this paper)***.** *Suppose that $A \in \mathbb{R}^{n\times n}$ is an SPSD matrix, r, K and n are three positive integers such that $r < K < n$, ξ is a positive number, and $\mathcal{I}$ is the output of Algorithm 6. Write $C := A_{:,\mathcal{I}}$, $U := (A_{\mathcal{I},\mathcal{I}})^+_r$, and $R := A_{\mathcal{I},:}$. Then*

$$||A - CUR||_C \le (1+\xi)\frac{K+1}{K-r+1}\sigma_{r+1}(A). \tag{6.2}$$

In particular, let $K = cr - 1$ for $c > 1$. Then

$$||A - CUR||_C \le (1+\frac{1}{c-1})(1+\xi)\sigma_{r+1}(A). \tag{6.3}$$

Furthermore Algorithm 6 runs at an arithmetic cost in $O(r^2K^4n + rK^4n\log n) = O((r+\log n)K^4n)$, which turns into $O((r+\log n)n^5)$ in case of a constant $c > 1$.

7 Proof of Main Result 1

Theorem 7 *(Adapted from [OZ18, Thm. 6] and [GT01, Thm. 2.1])***.** *Suppose that $W \in \mathbb{R}^{(r+1)\times(r+1)}$,*

$$W = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix},$$

and $A \in \mathbb{R}^{r\times r}$ has maximal volume among all $r \times r$ submatrices of W. Then

$$\frac{v_2(W)}{v_2(A)} \le (1+r)\sigma_{r+1}(W). \tag{7.1}$$

Algorithm 3. Greedy Column Subset Selection [CM09].

Input: $A \in \mathbb{R}^{m\times n}$an a positive integer $K < n$.
Output: $\mathcal{I}$.

Initialize $\mathcal{I} = \{\}$.
$M^1 \leftarrow A$.
for t = 1, 2, ..., K **do**
  $i \leftarrow \arg\max_{a\in[n]} ||M^t_{:,a}||$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$.
  $M^{t+1} \leftarrow M^t - ||M^t_{:,i}||^{-2}(M^t_{:,i})(M^t_{:,i})^T M^t$
end for
return $\mathcal{I}$.

Hereafter $[n]$ denotes the set of n integers $\{1, 2, \dots, n\}$, and $|\mathcal{T}|$ denotes the cardinality (the number of elements) of a set $\mathcal{T}$.

The theorem is readily deduced from the following result.

Theorem 8 *(Cf. [CKM19]). Suppose that W is an $n\times n$ SPSD matrix and $\mathcal{I}$ and $\mathcal{J}$ are two sets of integers in $[n]$ having the same cardinality. Then $v_2\left(W_{\mathcal{I},\mathcal{J}}\right)^2 \leq v_2\left(W_{\mathcal{I},\mathcal{I}}\right) v_2\left(W_{\mathcal{J},\mathcal{J}}\right)$.*

Theorem 8 shows that the maximal volume submatrix M of an SPSD matrix A can be chosen to be principal. This can be exploited to greatly reduce the cost of searching for the maximal volume submatrix. As pointed out in [CKM19] and implied in [CM09] searching for a maximal volume submatrix in a general matrix or even in an SPSD matrix is NP hard and therefore is impractical for inputs of even moderately large size. [CKM19] proposed to search for a submatrix with a large volume by means of algorithm that is equivalent to **Gaussian Elimination with Complete Pivoting** (Algorithm 4). Such a submatrix, however, only guarantees an upper bound of $4^r\sigma_{r+1}(A)$ on the Chebyshev error norm for the output CUR LRA (see the definition of Chebyshev's norm in Appendix B).

Next we seek a principal submatrix $A_{\mathcal{I},\mathcal{I}}$ having maximal volume in every matrix $A_{\mathcal{S},\mathcal{S}}$ such that $\mathcal{S} \supset \mathcal{I}$ and $|\mathcal{S}| = |\mathcal{I}| + 1$. Such a submatrix generates a CUR LRA with Chebyshev error norm bound $(r+1)\sigma_{r+1}(A)$, thus considerably improving the aforementioned exponential bound. According to the following theorem, we arrive at such a submatrix $A_{\mathcal{I},\mathcal{I}}$ by recursively replacing a single index in an initial set $\mathcal{I}$.

Theorem 9. *Suppose that $A \in \mathbb{R}^{n\times n}$ is an SPSD matrix, $\mathcal{I}$ is an index set, and $0 < |\mathcal{I}| = r < n$. Let $v_2\left(A_{\mathcal{I},\mathcal{I}}\right) \geq v_2\left(A_{\mathcal{J},\mathcal{J}}\right)$ for any index set $\mathcal{J}$ where $|\mathcal{J}| = r$, and $\mathcal{J}$ only differs from $\mathcal{I}$ at a single element. Then $A_{\mathcal{I},\mathcal{I}}$ is a maximal volume submatrix of $A_{\mathcal{S},\mathcal{S}}$ for any superset $\mathcal{S}$ of $\mathcal{I}$ lying in $[n]$ and such that $|\mathcal{S}| = r+1$.*

Proof. Apply [CKM19] Thm. 1 to such an SPSD matrix $A_{\mathcal{S},\mathcal{S}}$ and obtain that there exists a subset $\mathcal{I}'$ of $\mathcal{S}$ such that $|\mathcal{I}'| = r$ and $A_{\mathcal{I}',\mathcal{I}'}$ is a maximal volume submatrix of $A_{\mathcal{S},\mathcal{S}}$. $v_2\left(A_{\mathcal{I},\mathcal{I}}\right) \geq v_2\left(A_{\mathcal{I}',\mathcal{I}'}\right)$ since $\mathcal{I}'$ and $\mathcal{I}$ differs at most at a single element, and this proves the theorem.

Algorithm 4. An SPSD Matrix: Gaussian Elimination with Complete Pivoting (cf. [B00] and [CKM19]).

Input: An SPSD matrix $A \in \mathbb{R}^{n\times n}$ and a positive integer $K < n$.
Output: $\mathcal{I}$.

Initialize $R \leftarrow A$, and $\mathcal{I} = \{\}$.
for t = 1, 2, ..., K **do**
  $i_t \leftarrow \arg\max_{j\in[n]} |R_{j,j}|$.
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i_t\}$.
  $R \leftarrow R - R_{:,i_t} \cdot r^{-1}_{i_t,i_t} \cdot R_{i_t,:}$.
end for
return $\mathcal{I}$.

The papers [GT01] and [OZ18] have considerably relaxed the condition that the generator $A_{\mathcal{I},\mathcal{I}}$ is a maximal volume submatrix: if $v_2(A_{\mathcal{I},\mathcal{I}})$ is increased by a factor of $h > 1$ from maximal, then the error bound only increases by at most the same factor of h. In the case of SPSD inputs, we extend this relaxation further to $A_{\mathcal{I},\mathcal{I}}$ having **close-to-maximal** volume among "nearby" principal submatrices.

Theorem 10. *For an SPSD matrix $A \in \mathbb{R}^{n\times n}$, a positive integer $r < n$, and a positive number ξ, let $\mathcal{I} \subset [n]$ be an index set and let $|\mathcal{I}| = r$. Suppose that $(1+\xi)v_2(A_{\mathcal{I},\mathcal{I}}) \geq v_2(A_{\mathcal{J},\mathcal{J}})$ for any subset $\mathcal{J}$ of $[n]$ such that $|\mathcal{J}| = r$ and $\mathcal{J}$ differs from $\mathcal{I}$ at one element. Then*

$$||A - A_{:,\mathcal{I}}A^{-1}_{\mathcal{I},\mathcal{I}}A_{\mathcal{I},:}||_C \leq (1+\xi)(r+1)\sigma_{r+1}(A). \tag{7.2}$$

If $v_2(A_{\mathcal{I},\mathcal{I}})$ is increased by at most a factor of $1+\xi$ each time when we replace an index in $\mathcal{I}$, then Algorithm 6 would not run into infinite loop due to rounding to machine precision. Furthermore, Theorem 10 guarantees that the accuracy is mostly preserved, that is, upon termination, the returned index set $\mathcal{I}$ satisfies inequality (7.2).

Let t denote the number of times a single index in $\mathcal{I}$ is replaced. In the following, we show that t is bounded by $O(r\log r)$, if the initial set $\mathcal{I}_0$ is greedily chosen in Algorithm 3.

Theorem 11 *(Adapted from [CM09] Thm. 10)***.** *For a matrix $C \in \mathbb{R}^{m\times n}$ and a positive integer $r < n$, let Algorithm 3 with input C and r output a set $\mathcal{I}$. Then*

$$v_2(C_{:,\mathcal{I}}) \geq \frac{1}{r!} \max_{\mathcal{S}\subset[n]:|\mathcal{S}|=r} v_2(C_{:,\mathcal{S}}). \tag{7.3}$$

Theorem 12. *For an SPSD matrix $A \in \mathbb{R}^{n\times n}$ and a positive integer $r < n$, let Algorithm 4 with inputs A and r output a set $\mathcal{I}$. Then*

$$v_2(A_{\mathcal{I},\mathcal{I}}) \geq \frac{1}{(r!)^2} \max_{\mathcal{S}\subset[n]:|\mathcal{S}|=r} v_2(A_{\mathcal{S},\mathcal{S}}). \tag{7.4}$$

Corollary 4. *For an SPSD matrix $A \in \mathbb{R}^{n\times n}$, a positive integer $r < n$, and a positive number ξ, Algorithm 6 calls Algorithm 5 at most $O(r\log r)$ times.*

Algorithm 5. Index Swap

Input: An SPSD matrix $A \in \mathbb{R}^{n\times n}$, a set $\mathcal{I} \in [n]$, a positive integer $r \le |\mathcal{I}|$, and a positive number ξ.
Output: $\mathcal{J}$

```
Compute v_{2,r}(A_{I,I})
for all i ∈ I do
  I' ← I − {i}
  for all j ∈ [n] − I do
    J ← I' ∪ {j}
    Compute v_{2,r}(A_{J,J})
    if v_{2,r}(A_{J,J}) / v_{2,r}(A_{I,I}) > 1 + ξ then
      return J
    end if
  end for
end for
return I
```

8 Complexity Analysis

In this section, we estimate the time complexity of performing the Main Algorithm (Algorithm 6) in the case of both $r = K$ and $r < K$. The cost of finding the initial set $\mathcal{I}_0$ by means of Algorithm 4 is $O(nK^2)$. Let t denote the number of iterations and let $c(r, K)$ denote the arithmetic cost of performing Algorithm 5 with parameters r and K. Then the complexity is in $O(nK^2 + t \cdot c(r, K))$.

In the case of $r = K$, Corollary 4 implies that $t = O(r \log r)$. Algorithm 5 may need up to nr comparisons of $v_2(A_{\mathcal{I},\mathcal{I}})$ and $v_2(A_{\mathcal{J},\mathcal{J}})$. Since $\mathcal{I}$ and $\mathcal{J}$ differs at most at one index, we compute $v_2(A_{\mathcal{J},\mathcal{J}})$ faster by using small rank update of

Algorithm 6. Main Algorithm

Input: An SPSD matrix $A \in \mathbb{R}^{n\times n}$, two positive integers K and r such that $r \le K < n$, and a positive number ξ.
Output: $\mathcal{I}$

```
I ← Algorithm 4(A, K)
while TRUE do
  J ← Algorithm 5(A, I, r, ξ)
  if J = I then
    BREAK
  else
    I ← J
  end if
end while
return I.
```

$A_{\mathcal{I},\mathcal{I}}$ instead of computing from the scratch; this saves a factor of k. Therefore $c(r,r) = O(r^3 n)$, and the time complexity of the Main Algorithm is $O(nr^4 \log r)$.

In the case of $r < K$, according to [GE96, Theorem 7.2] and [CM09, Theorem 10], t increases slightly to $O(r^2 + r \log n)$, and if $v_{2,r}(A_{\mathcal{J},\mathcal{J}})$ is computed by using SVD, then $c(r,K) = O(K^4 n)$, and the time complexity of the Main Algorithm is $O(r^2 K^4 n + rK^4 n \log n)$.

Acknowledgements. Our research has been supported by NSF Grants CCF-1116736, CCF-1563942, and CCF-133834 and PSC CUNY Award 69813 00 48. We also thank A. Cortinovis, A. Osinsky, N. L. Zamarashkin for pointers to their papers [CKM19] and [OZ18], S. A. Goreinov for reprints, of his papers, and E. E. Tyrtyshnikov for pointers to the bibliography and the challenge of formally supporting empirical power of C-A algorithms.

Appendix

A Small Families of Hard Inputs for Sublinear Cost LRA

Any sublinear cost LRA algorithm fails on the following small input families.

Example 1. Define a family of $m \times n$ matrices of rank 1 (we call them *δ-matrices*):

$$\{\Delta_{i,j},\ i = 1, \ldots, m;\ j = 1, \ldots, n\}.$$

Also include the $m \times n$ null matrix $O_{m,n}$ into this family. Now fix any sublinear cost algorithm; it does not access the (i,j)th entry of its input matrices for some pair of i and j. Therefore it outputs the same approximation of the matrices $\Delta_{i,j}$ and $O_{m,n}$, with an undetected error at least 1/2. Apply the same argument to the set of $mn + 1$ small-norm perturbations of the matrices of the above family and to the $mn + 1$ sums of the latter matrices with any fixed $m \times n$ matrix of low rank. Finally, the same argument shows that a posteriori estimation of the output errors of an LRA algorithm applied to the same input families cannot run at sublinear cost.

This example actually covers randomized LRA algorithms as well. Indeed suppose that with a positive constant probability an LRA algorithm does not access K entries of an input matrix. Apply this algorithm to two matrices of low rank whose difference at all these K entries is equal to a large constant C. Then, clearly, with a positive constant probability the algorithm has errors at least $C/2$ at at least $K/2$ of these entries.

B Definitions for Matrix Computations and a Lemma

Next we recall some basic definitions for matrix computations (cf. [GL13]).

$\mathbb{C}^{m \times n}$ is the class of $m \times n$ matrices with complex entries.

I_s denotes the $s \times s$ identity matrix. $O_{q,s}$ denotes the $q \times s$ matrix filled with zeros.

$\mathrm{diag}(B_1,\ldots,B_k) = \mathrm{diag}(B_j)_{j=1}^k$ denotes a $k \times k$ block diagonal matrix with diagonal blocks $B_1,\ldots,B_k$.

$(B_1 \mid \ldots \mid B_k)$ and $(B_1,\ldots,B_k)$ denote a $1 \times k$ block matrix with blocks $B_1,\ldots,B_k$.

W^T and W^* denote the transpose and the Hermitian transpose of an $m \times n$ matrix $W = (w_{ij})_{i,j=1}^{m,n}$, respectively. $W^* = W^T$ if the matrix W is real.

For two sets $\mathcal{I} \subseteq \{1,\ldots,m\}$ and $\mathcal{J} \subseteq \{1,\ldots,n\}$ define the submatrices

$$W_{\mathcal{I},:} := (w_{i,j})_{i\in\mathcal{I};j=1,\ldots,n}, W_{:,\mathcal{J}} := (w_{i,j})_{i=1,\ldots,m;j\in\mathcal{J}},\ W_{\mathcal{I},\mathcal{J}} := (w_{i,j})_{i\in\mathcal{I};j\in\mathcal{J}}. \tag{B.1}$$

An $m \times n$ matrix W is *unitary* (also *orthogonal* when real) if $W^*W = I_n$ or $WW^* = I_m$.

Compact SVD of a matrix W, hereafter just *SVD*, is defined by the equations

$$W = S_W \Sigma_W T_W^*,$$
$$\text{where } S_W^* S_W = T_W^* T_W = I_\rho,\ \Sigma_W := \mathrm{diag}(\sigma_j(W))_{j=1}^{\rho},\ \rho = \mathrm{rank}(W), \tag{B.2}$$

$\sigma_j(W)$ denotes the jth largest singular value of W for $j = 1,\ldots,\rho$; $\sigma_j(W) = 0$ for $j > \rho$.

$||W|| = ||W||_2$, $||W||_F$, and $||W||_C$ denote spectral, Frobenius, and Chebyshev norms of a matrix W, respectively, such that (see [GL13, Section 2.3.2 and Corollary 2.3.2])

$$||W|| = \sigma_1(W),\ ||W||_F^2 := \sum_{i,j=1}^{m,n} |w_{ij}|^2 = \sum_{j=1}^{\mathrm{rank}(W)} \sigma_j^2(W),\ ||W||_C := \max_{i,j=1}^{m,n} |w_{ij}|,$$

$$||W||_C \le ||W|| \le ||W||_F \le \sqrt{mn}\ ||W||_C,\ ||W||_F^2 \le \min\{m,n\}\ ||W||^2. \tag{B.3}$$

$W^+ := T_W \Sigma_W^{-1} S_W^*$ is the Moore–Penrose pseudo inverse of an $m \times n$ matrix W.

$$||W^+||\sigma_r(W) = 1 \tag{B.4}$$

for a full rank matrix W.

A matrix W has *ϵ-rank* at most $r > 0$ for a fixed tolerance $\epsilon > 0$ if there is a matrix W' of rank r such that $||W' - W||/||W|| \le \epsilon$. We write $\mathrm{nrank}(W) = r$ and say that a matrix W has *numerical rank* r if it has ϵ-rank r for a small ϵ.

Lemma 3. *Let $G \in \mathbb{C}^{k\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$ and $H \in \mathbb{C}^{r\times l}$ and let the matrices G, H and Σ have full rank $r \le \min\{k,l\}$. Then $||(G\Sigma H)^+|| \le ||G^+||\ ||\Sigma^+||\ ||H^+||$.*

A proof of this well-known result is included in [LPa].

C The Volume and r-Projective Volume of a Perturbed Matrix

Theorem 13. *Suppose that W' and E are $k \times l$ matrices,* $\mathrm{rank}(W') = r \le \min\{k,l\}$*, $W = W' + E$, and $||E|| \le \epsilon$. Then*

$$\Big(1 - \frac{\epsilon}{\sigma_r(W)}\Big)^r \le \prod_{j=1}^{r}\Big(1 - \frac{\epsilon}{\sigma_j(W)}\Big) \le \frac{v_{2,r}(W)}{v_{2,r}(W')} \le \prod_{j=1}^{r}\Big(1 + \frac{\epsilon}{\sigma_j(W)}\Big) \le \Big(1 + \frac{\epsilon}{\sigma_r(W)}\Big)^r. \tag{C.1}$$

If $\min\{k, l\} = r$, *then* $v_2(W) = v_{2,r}(W)$, $v_2(W') = v_{2,r}(W')$, *and*

$$\left(1 - \frac{\epsilon}{\sigma_r(W)}\right)^r \leq \frac{v_2(W)}{v_2(W')} = \frac{v_{2,r}(W)}{v_{2,r}(W')} \leq \left(1 + \frac{\epsilon}{\sigma_r(W)}\right)^r. \tag{C.2}$$

Proof. Bounds (C.1) follow because a perturbation of a matrix within a norm bound ϵ changes its singular values by at most ϵ (see [GL13, Corollary 8.6.2]). Bounds (C.2) follow because $v_2(M) = v_{2,r}(M) = \prod_{j=1}^{r} \sigma_j(M)$ for any $k \times l$ matrix M with $\min\{k, l\} = r$, in particular for $M = W'$ and $M = W = W' + E$.

If the ratio $\frac{\epsilon}{\sigma_r(W)}$ is small, then $\left(1 - \frac{\epsilon}{\sigma_r(W)}\right)^r = 1 - O\left(\frac{r\epsilon}{\sigma_r(W)}\right)$ and $\left(1 + \frac{\epsilon}{\sigma_r(W)}\right)^r = 1 + O\left(\frac{r\epsilon}{\sigma_r(W)}\right)$, which shows that the relative perturbation of the volume is amplified by at most a factor of r in comparison to the relative perturbation of the r largest singular values.

D The Volume and *r*-Projective Volume of a Matrix Product

Theorem 14 *(Cf. [OZ18]).* [Examples 2 and 3 below show some limitations on the extension of the theorem.]

Suppose that $W = GH$ *for an* $m \times q$ *matrix* G *and a* $q \times n$ *matrix* H. *Then*

(i) $v_2(W) = v_2(G)v_2(H)$ *if* $q = \min\{m, n\}$; $v_2(W) = 0 \leq v_2(G)v_2(H)$ *if* $q < \min\{m, n\}$.
(ii) $v_{2,r}(W) \leq v_{2,r}(G)v_{2,r}(H)$ *for* $1 \leq r \leq q$,
(iii) $v_2(W) \leq v_2(G)v_2(H)$ *if* $m = n \leq q$.

Example 2. If G and H are unitary matrices and if $GH = O$, then $v_2(G) = v_2(H) = v_{2,r}(G) = v_{2,r}(H) = 1$ and $v_2(GH) = v_{2,r}(GH) = 0$ for all $r \leq q$.

Example 3. If $G = (1 \mid 0)$ and $H = \text{diag}(1, 0)$, then $v_2(G) = v_2(GH) = 1$ and $v_2(H) = 0$.

References

[B00] Bebendorf, M.: Approximation of boundary element matrices. Numer. Math. **86**(4), 565–589 (2000)

[CI94] Chandrasekaran, S., Ipsen, I.: On rank revealing QR factorizations. SIAM J. Matrix Anal. Appl. **15**, 592–622 (1994)

[CKM19] Cortinovis, A., Kressner, D., Massei, S.: MATHICSE technical report: on maximum volume submatrices and cross approximation for symmetric semidefinite and diagonally dominant matrices. MATHICSE, 12 February 2019

[CM09] Çivril, A., Magdon-Ismail, M.: On selecting a maximum volume submatrix of a matrix and related problems. Theor. Comput. Sci. **410**(47–49), 4801–4811 (2009)

[DMM08] Drineas, P., Mahoney, M.W., Muthukrishnan, S.: Relative-error CUR matrix decompositions. SIAM J. Matrix Anal. Appl. **30**(2), 844–881 (2008)

[GE96] Gu, M., Eisenstat, S.C.: An efficient algorithm for computing a strong rank revealing QR factorization. SIAM J. Sci. Comput. **17**, 848–869 (1996)

[GL13] Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. The Johns Hopkins University Press, Baltimore (2013)

[GT01] Goreinov, S.A., Tyrtyshnikov, E.E.: The maximal-volume concept in approximation by low rank matrices. Contemp. Math. **208**, 47–51 (2001)

[GTZ97] Goreinov, S.A., Tyrtyshnikov, E.E., Zamarashkin, N.L.: A theory of pseudo-skeleton approximations. Linear Algebra Appl. **261**, 1–21 (1997)

[LPa] Luan, Q., Pan, V.Y.: Low rank approximation of a matrix at sublinear cost, 21 July 2019. arXiv:1907.10481

[MD09] Mahoney, M.W., Drineas, P.: CUR matrix decompositions for improved data analysis. Proc. Natl. Acad. Sci. USA **106**, 697–702 (2009)

[MW17] Musco, C., Woodruff, D.P.: Sublinear time low-rank approximation of positive semidefinite matrices. In: IEEE 58th FOCS, pp. 672–683 (2017)

[O17] Osinsky, A.I.: Probabilistic estimation of the rank 1 cross approximation accuracy, submitted on 30 June 2017. arXiv:1706.10285

[OZ18] Osinsky, A.I., Zamarashkin, N.L.: Pseudo-skeleton approximations with better accuracy estimates. Linear Algebra Appl. **537**, 221–249 (2018)

[PLa] Pan, V.Y., Luan, Q.: Refinement of low rank approximation of a matrix at sub-linear cost, submitted on 10 June 2019. arXiv:1906.04223

[PLSZ16] Pan, V.Y., Luan, Q., Svadlenka, J., Zhao, L.: Primitive and cynical low rank approximation, preprocessing and extensions, submitted on 3 November 2016. arXiv:1611.01391v1

[PLSZ17] Pan, V.Y., Luan, Q., Svadlenka, J., Zhao, L.: Superfast accurate approximation of low rank matrices, submitted on 22 October 2017. arXiv:1710.07946v1

[PLSZa] Pan, V.Y., Luan, Q., Svadlenka, J., Zhao, L.: CUR low rank approximation at sub-linear cost, submitted on 10 June 2019. arXiv:1906.04112

[PQY15] Pan, V.Y., Qian, G., Yan, X.: Random multipliers numerically stabilize Gaussian and block Gaussian elimination: proofs and an extension to low-rank approximation. Linear Algebra Appl. **481**, 202–234 (2015)

[PZ17a] Pan, V.Y., Zhao, L.: New studies of randomized augmentation and additive preprocessing. Linear Algebra Appl. **527**, 256–305 (2017)

[PZ17b] Pan, V.Y., Zhao, L.: Numerically safe Gaussian elimination with no pivoting. Linear Algebra Appl. **527**, 349–383 (2017)