arXiv:1906.04327v3 [math.NA] 20 Jul 2019

Low Rank Approximation at Sublinear Cost by Means of Subspace Sampling

Victor Y. Pan[1,2],[a], Qi Luan[2],[b], John Svadlenka[2],[c], and Liang Zhao[1,2],[d]

[1] Department of Computer Science
Lehman College of the City University of New York
Bronx, NY 10468 USA
[2] Ph.D. Programs in Computer Science and Mathematics
The Graduate Center of the City University of New York
New York, NY 10036 USA
[a] victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan/
[b] qi_luan@yahoo.com
[c] jsvadlenka@gradcenter.cuny.edu
[d] Liang.Zhao1@lehman.cuny.edu

Abstract

Low Rank Approximation (LRA) of a matrix is a hot research subject, fundamental for Matrix and Tensor Computations and Big Data Mining and Analysis. Computations with LRA can be performed *at sub-linear cost*, that is, by using much fewer arithmetic operations and memory cells than an input matrix has entries. Although every sub-linear cost algorithm for LRA fails to approximate the worst case inputs, we prove that our sub-linear cost variations of a popular *subspace sampling* algorithm output accurate LRA of a large class of inputs. Namely, they do so with a high probability (hereafter *whp*) for a random input matrix that admits its LRA. In other papers we proposed and analyzed sub-linear cost algorithms for other important matrix computations. Our numerical tests are in good accordance with our formal results.

Key Words: Low-rank approximation, Sub-linear cost, Subspace sampling

2000 Math. Subject Classification: 65Y20, 65F30, 68Q25, 68W20, 15A52

1 Introduction

LRA Background. Low rank approximation (LRA) of a matrix is a hot research area of Numerical Linear Algebra (NLA) and Computer Science (CS) with applications to fundamental matrix and tensor computations and data mining and analysis (see surveys [HMT11], [M11], [KS16], and [CLO16]). Matrices defining Big Data (e.g., unfolding matrices of multidimensional tensors) are frequently so immense that realistically one can access and process only a tiny fraction of their entries, although quite typically these matrices admit their LRA, that is, are close to low rank matrices or equivalently have *low numerical rank*. One can operate with low rank matrices *at sub-linear computational cost*, that is, by using much fewer arithmetic operations and memory cells

than an input matrix has entries, but can we compute LRA at sub-linear cost? Yes and no. No, because every sub-linear cost LRA algorithm fails even on the small input families of Appendix C. Yes, because our sub-linear cost variations of a popular *subspace sampling* algorithm output accurate LRA for a large class of input. Let us provide some details.

Subspace sampling algorithms compute LRA of a matrix M by using auxiliary matrices FM, MH or FMH for random multipliers F and H, commonly called *test matrices* and having smaller sizes. Their output LRA are nearly optimal whp provided that F and/or H are Gaussian, Rademacher's, SRHT or SRFT matrices;[1] furthermore the algorithms consistently output accurate LRA in their worldwide application with these and some other random multipliers F and H, all of which are multiplied by M at super-linear cost (see [TYUC17, Section 3.9], [HMT11, Section 7.4], and the bibliography therein).

Our modifications of these algorithms use sparse orthogonal (e.g., *sub-permutation*) multipliers[2] F and H, run at sub-linear cost, and as we prove, whp output reasonably accurate *dual LRA*, that is, LRA of a random input admitting LRA; we deduce our error estimates under three distinct models of random matrix computations in Sections 4.1 – 4.3.

How meaningful is our result? Our definitions of three classes of random matrices of low numerical rank are quite natural for various real world applications of LRA, but are odd for some other ones. This, however, applies to any definition of that kind.

Our approach enables new insight into the subject, and our formal study is in good accordance with our numerical tests for both synthetic and real world inputs, some from [HMT11].

Our upper bounds on the output error of LRA of an $m \times n$ matrix of numerical rank r exceed the optimal error bound by a factor of $\sqrt{\min\{m,n\}r}$, but if this optimal bound is small enough we can apply iterative refinement of LRA running at sub-linear cost (see [PLa]).

As we have pointed out, any sub-linear cost LRA algorithm (and ours are no exception) fails on some families of hard inputs, but our analysis and tests show that the class of such inputs is narrow. We conjecture that it shrinks fast if we recursively apply the same algorithm with new multipliers; we propose some heuristic recipes for these recursive processes, and our numerical tests confirm their efficiency.

Impact of our study, its extensions and by-products:

(i) Our duality approach enables new insight into some fundamental matrix computations besides LRA: [PQY15], [PZ17a], and [PZ17b] provide formal support for empirical efficiency of dual Gaussian elimination with no pivoting, while [PLb] proposes a sub-linear cost modification of Sarlós' algorithm of 2006 and then proves that whp it outputs nearly optimal solution of the highly important problem of *Linear Least Squares Regression (LLSR)* provided that its input is random. Then again this formal proof is in good accordance with the test results.

(ii) In [PLSZa] we proved that popular Cross-Approximation LRA algorithms running at sub-linear cost as well as our simplified sub-linear cost variations of these algorithms output accurate solution of dual LRA whp, and we also devised a sub-linear cost algorithm for transformation of any LRA into its special form of CUR LRA, which is particularly memory efficient.

(iii) The paper [PLa] has proposed and elaborated upon sub-linear cost refinement of a crude but reasonably close LRA.

Related Works. Huge bibliography on LRA can be partly accessed via [M11], [HMT11], [KS16], [PLSZ17], [TYUC17], [OZ18], and the references therein. [PLSZ16] and [PLSZ17] were the first papers that provided formal support for dual accurate randomized LRA computations

[1] Here and hereafter "SRHT and SRFT" are the acronyms for "Subsample Random Hadamard and Fourier transforms"; *"Gaussian"* stands for "standard Gaussian (normal) random"; Rademacher's are the matrices filled with iid variables, each equal to 1 or -1 with probability 1/2.

[2] We define sub-permutation matrices as full-rank submatrices of permutation matrices.

performed at sub-linear cost (in these papers such computations are called superfast). The earlier papers [PQY15], [PLSZ16], [PZ17a], and [PZ17b] studied duality for other fundamental matrix computations besides LRA, while the paper [PLb] has extended our study to a sub-linear cost dual algorithm for the popular problem of Linear Least Squares Regression and confirmed accuracy of this solution by the results of numerical experiments.

Organization of the paper. In Section 2 we recall random sampling for LRA. In Sections 3 and 4 we prove deterministic and randomized error bounds, respectively, for our dual LRA algorithms running at sub-linear cost. In Section 5 we discuss multiplicative pre-processing and generation of multipliers for both pre-processing and sampling. In Section 6 we cover our numerical tests. Appendix A is devoted to background on matrix computations. In Appendix B we prove our error bounds for dual LRA. In Appendix C we specify some small families of hard inputs for sub-linear cost LRA.

Some definitions. The concepts "large", "small", "ill-" and "well-conditioned", "near", "close", and "approximate" are usually quantified in the context. "$\ll$" and "$\gg$" mean "much less than" and "much greater than", respectively. *"Flop"* stands for "floating point arithmetic operation"; "*iid*" for "independent identically distributed". In context a *"perturbation of a matrix"* can mean a perturbation having a small relative norm. $\mathbb{R}^{p\times q}$ denotes the class of $p\times q$ real matrices, We assume dealing with real matrices throughout, and so the Hermitian transpose of M turns into transpose, $M^* = M^T$, but most of our study can be readily extended to complex matrices; see some relevant results about complex Gaussian matrices in [E88], [CD05], [ES05], and [TYUC17].

2 LRA by means of subspace sampling

2.1 Four subspace sampling algorithms

Hereafter $||\cdot||$ and $||\cdot||_F$ denote the spectral and the Frobenius matrix norms, respectively; $|\cdot|$ can denote either of them. M^+ denotes the Moore – Penrose pseudo inverse of M.

Algorithm 2.1. Column Subspace Sampling or Range Finder (see Remark 2.1).

INPUT: *An $m\times n$ matrix M and a target rank r.*

OUTPUT: *Two matrices $X\in\mathbb{R}^{m\times l}$ and $Y\in\mathbb{R}^{l\times m}$ defining an LRA $\tilde{M}=XY$ of M.*

INITIALIZATION: *Fix an integer l, $r\le l\le n$, and an $n\times l$ matrix H of full rank l.*

COMPUTATIONS: *1. Compute the $m\times l$ matrix MH.*

2. Fix a nonsingular $l\times l$ matrix T^{-1} and output the $m\times l$ matrix $X:=MHT^{-1}$.

3. Output an $l\times n$ matrix $Y:=\mathrm{argmin}_V\ |XV-M|$.

Remark 2.1. *Let* $\mathrm{rank}(MH)=l$. *Then $Y=(MH)^+M$ and $XY=MH(MH)^+M$ independently of the choice of T^{-1}, but its proper choice numerically stabilizes the computations of the algorithm. For $l>r\ge \mathrm{nrank}(MH)$ the matrix MH is ill-conditioned,*[3] *but MHT^{-1} is orthogonal for $T=R$, $X:=Q=MHR^{-1}$ and $Y:=Q^*M$ where Q and R are the factors of the thin QR factorization of MH (cf. [HMT11, Algorithm 4.1]). It is also orthogonal for $T=R\Pi$ and the factors R and Π in a rank-revealing* QRΠ *factorization $MH=QR\Pi$.*

[3] $\mathrm{nrank}(W)$ denotes *numerical rank of W* (see Appendix A.1).

Column Subspace Sampling turns into *Column Subset Selection* in the case of a sub-permutation matrix H and turns into *Row Subspace Sampling* if it is applied to the transpose M^T.

Algorithm 2.2. Row Subspace Sampling or Transposed Range Finder. See Remark 2.2.

INPUT: *As in Algorithm 2.1.*

OUTPUT: *Two matrices* $X \in \mathbb{R}^{k\times n}$ *and* $Y \in \mathbb{R}^{m\times k}$ *defining an LRA* $\tilde{M} = YX$ *of* M.

INITIALIZATION: *Fix an integer* k, $r \le k \le m$, *and a* $k \times m$ *matrix* F *of full numerical rank* k.

COMPUTATIONS: *1. Compute the* $k \times m$ *matrix* FM.

2. Fix a nonsingular $k \times k$ *matrix* S^{-1}*; then output* $k \times n$ *matrix* $X := S^{-1}FM$.

3. Output an $m \times k$ *matrix* $Y := \mathrm{argmin}_V\ |VX - M|$.

Row Subspace Sampling turns into *Row Subset Selection* in case of a sub-permutation matrix F.

Remark 2.2. *Let* $\mathrm{rank}(FM) = l$. *Then* $Y = M(FM)^+$ *and* $YX = M(FM)^+FM$ *independently of the choice of* S^{-1}*, but a proper choice of* S *numerically stabilizes the computations of the algorithm. For* $k > r \ge \mathrm{nrank}(FM)$ *the matrix* FM *is ill-conditioned, but* $S^{-1}FM$ *is orthogonal if* $S = L$, $X := Q = L^{-1}FM$, $Y := Q^*M$, *and* L *and* Q *are the factors of the thin LQ factorization of* FM *or if* $S = \Pi L$ *and* Π *and* L *are the factors in a rank-revealing* ΠLQ *factorization* $FM = \Pi LQ$.

The following algorithm combines row and column subspace sampling. In the case of the identity matrix S it turns into the algorithm of [CW09, Theorems 4.7 and 4.8] and [TYUC17, Section 1.4], whose origin can be traced back to [WLRT08].

Algorithm 2.3. Row and Column Subspace Sampling. See Remark 2.3.

INPUT: *As in Algorithm 2.1.*

OUTPUT: *Two matrices* $X \in \mathbb{R}^{m\times k}$ *and* $Y \in \mathbb{R}^{k\times m}$ *defining an LRA* $\tilde{M} = XY$ *of* M.

INITIALIZATION: *Fix two integers* k *and* l, $r \le k \le m$ *and* $r \le l \le n$*; fix two matrices* $F \in \mathbb{R}^{k\times m}$ *and* $H \in \mathbb{R}^{n\times l}$ *of full numerical ranks and two nonsingular matrices* $S \in \mathbb{R}^{k\times k}$ *and* $T \in \mathbb{R}^{l\times l}$.

COMPUTATIONS: *1. Output the matrix* $X = MHT^{-1} \in \mathbb{R}^{m\times l}$.

2. Compute the matrices $V = S^{-1}FM \in \mathbb{R}^{k\times n}$ *and* $W = S^{-1}FX \in \mathbb{R}^{m\times l}$.

3. Output the $l \times n$ *matrix* $Y := \mathrm{argmin}_V |W^+V - S^{-1}FM|$.

Remark 2.3. *If the matrix* FMH *has full rank* $\min\{k, n\}$, *then* $YX = MH(FHM)^+FM$, *independently of the choice of the matrices* S^{-1} *and* T^{-1}*, but a proper choice of* S *numerically stabilizes the computations of the algorithm. For* $\min\{k, l\} > r \ge \mathrm{nrank}(FX)$ *the matrix* FM *is ill-conditioned, but* $S^{-1}FM$ *is orthogonal if* $S = L$, $X := Q = L^{-1}FM$, $Y := Q^*M$, *and* L *and* Q *are the factors of the thin LQ factorization of* FM *or if* $S = \Pi L$ *and* Π *and* L *are the factors in a rank-revealing* ΠLQ *factorization* $FM = \Pi LQ$.

Remark 2.4. *By applying Algorithm 2.3 to the transpose matrix* M^* *we obtain* **Algorithm 2.4**. *It begins with column subspace sampling followed by row subspace sampling. We only study Algorithms 2.1 and 2.3 for input* M, *but they turn into Algorithms 2.2 and 2.4 for the input* M^*.

2.2 The known error bounds

Theorem 2.1. *(i) Let $2 \le r \le l-2$ and apply Algorithm 2.1 with a Gaussian multiplier H. Then (cf. [HMT11, Theorems 10.5 and 10.6])* [4]

$$\mathbb{E}||M - XY||_F^2 \le \left(1 + \frac{r}{l-r-1}\right) \sigma_{F,r+1}^2(M),$$

$$\mathbb{E}||M - XY|| \le \left(1 + \sqrt{\frac{r}{l-r-1}}\ \right) \sigma_{r+1}(M) + \frac{e\sqrt{l}}{l-r}\sigma_{F,r+1}(M).$$

(ii) Let $4[\sqrt{r} + \sqrt{8\log(rn)}]^2 \log(r) \le l \le n$ and apply Algorithm 2.1 with an SRHT or SRFT multiplier H. Then (cf. [T11], [HMT11, Theorem 11.2])

$$|M - XY| \le \sqrt{1 + 7n/l}\ \ \bar{\sigma}_{r+1}(M) \text{ with a probability in } 1 - O(1/r).$$

Clarkson and Woodruff prove in [CW09] that Algorithm 2.3 reaches the bound $\bar{\sigma}_{r+1}(M)$ within a factor of $1+\epsilon$ whp if the multipliers $F \in \mathcal{G}^{k\times m}$ and $H \in \mathcal{G}^{n\times l}$ are Rademacher's matrices and if k and l are sufficiently large, having order of r/ϵ and r/ϵ^2 for small ϵ, respectively.

Tropp et al. argue in [TYUC17, Section 1.7.3] that LRA is not practical if the numbers k and l of row and column samples are large; iterative refinement of LRA at sub-linear cost in [PLa] can be a partial remedy. [TYUC17, Theorem 4.3] shows that the output LRA XY of Algorithm 2.3 applied with Gaussian multipliers F and H satisfies[5]

$$\mathbb{E}||M - XY||_F^2 \le \frac{kl}{(k-l)(l-r)}\sigma_{F,r+1}^2(M) \text{ if } k > l > r. \tag{2.1}$$

3 Deterministic output error bounds for sampling algorithms

3.1 Deterministic error bounds of Range Finder

Theorem 3.1. [HMT11, Theorem 9.1]. *Suppose that Algorithm 2.1 has been applied to a matrix M with a multiplier H and let*

$$C_1 = V_1^*H,\ C_2 = V_2^*H, \tag{3.1}$$

$$M = \begin{pmatrix} U_1 & \Sigma_1 & \\ U_2 & & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix},\ M_r = U_1\Sigma_1 V_1^*, \text{ and } M - M_r = U_2\Sigma_2 V_2^* \tag{3.2}$$

be SVDs of the matrices M, its rank-r truncation M_r, and $M - M_r$, respectively. [$\Sigma_2 = O$ and $XY = M$ if $\operatorname{rank}(M) = r$. The columns of V_1^* span the top right singular space of M.] Then

$$|M - XY|^2 \le |\Sigma_2|^2 + |\Sigma_2 C_2 C_1^+|^2. \tag{3.3}$$

Notice that $|\Sigma_2| = \bar{\sigma}_{r+1}(M)$, $|C_2| \le 1$, and $|\Sigma_2 C_2 C_1^+| \le |\Sigma_2|\ |C_2|\ |C_1^+|$ and obtain

$$|M - XY| \le (1 + |C_1^+|^2)^{1/2}\bar{\sigma}_{r+1}(M) \text{ for } C_1 = V_1^*H. \tag{3.4}$$

It follows that the output LRA is optimal up to a factor of $(1 + |C_1^+|^2)^{1/2}$.

Next we deduce an upper bound on the norm $|C_1^+|$ in terms of $||((MH)_r)^+||$, $||M||$, and $\eta := 2\sigma_{r+1}(M)\ ||((MH)_r)^+||$. Given MH we can compute the norm $||((MH)_r)^+||$ at sub-linear cost if $l \ll m$, and in some applications reasonable upper estimates for $||M||$ and σ_{r+1} are available.

[4] [HMT11, Theorems 10.7 and 10.8] estimate the norm $|M - XY|$ in probability.

[5] In words, the expected output error norm $\mathbb{E}||M - XY||_F$ is within a factor of $\left(\frac{kl}{(k-l)(l-r)}\right)^{1/2}$ from its minimum value $\sigma_{F,r+1}(M)$; this factor is just 2 for $k = 2l = 4r$.

Corollary 3.1. *Under the assumptions of Theorem 3.1 let the matrix M_rH have full rank r. Then*

$$|(M_rH)^+|/|M_r^+| \le |C_1^+| \le |(M_rH)^+|\ |M_r| \le |(M_rH)^+|\ |M|.$$

Proof. Deduce from (3.1) and (3.2) that $M_rH = U_1\Sigma_1 C_1$. Hence $C_1 = \Sigma_1^{-1}U_1^*M_rH$.

Recall that the matrix M_rH has full rank r, apply Lemma A.1, recall that U_1 is an orthogonal matrix, and obtain $|(M_rH)^+|/|\Sigma_1^{-1}| \le |C_1^+| \le |(M_rH)^+|\ |\Sigma_1|$.

Substitute $|\Sigma_1| = |M_r|$ and $|\Sigma_1^{-1}| = |M_r^+|$ and obtain the corollary. □

Corollary 3.2. *Under the assumptions of Corollary 3.1 let $\eta := 2\sigma_{r+1}(M)\ ||((MH)_r)^+|| < 1$ and $\eta' := \frac{2\sigma_{r+1}(M)}{1-\eta}\ ||((MH)_r)^+|| < 1$. Then*

$$\frac{1-\eta'}{||M_r^+||}\ ||((MH)_r)^+|| \le ||C_1^+\ || \le \frac{||M||}{1-\eta}\ ||((MH)_r)^+||.$$

Proof. Lemma A.3 implies that $\max\{|M_rH - MH|, |MH - (MH)_r|\} \le \bar\sigma_{r+1}(M)$.

Consequently $|M_rH - (MH)_r| \le 2\bar\sigma_{r+1}(M)$, and so $||(M_rH)^+|| \le \frac{1}{1-\eta}\ ||((MH)_r)^+||$ by virtue of Lemma A.2 if $\eta = 2\sigma_{r+1}(M)\ ||((MH)_r)^+|| < 1$.

If in addition $\eta' = \frac{2\sigma_{r+1}(M)}{1-\eta}\ ||((MH)_r)^+|| < 1$, then $2\sigma_{r+1}(M)\ ||(M_rH)^+|| < 1$ and therefore $||((MH)_r)^+|| \le \frac{1}{1-\eta'}\ ||(M_rH)^+||$ by virtue of Lemma A.2.

Combine these bounds and obtain $(1-\eta')\ ||((MH)_r)^+|| \le |(M_rH)^+| \le \frac{1}{1-\eta}\ ||((MH)_r)^+||$.

Together with Corollary 3.1 this implies Corollary 3.2. □

For a given matrix MH we can compute the norm $||((MH)_r)^+||$ at sub-linear cost if $l \ll m$. If also some reasonable upper bounds on $||M||$ and $\sigma_{r+1}(M)$ are known, then Corollary 3.2 implies *a posteriori estimates* for the output errors of Algorithm 2.1.

3.2 Impact of pre-multiplication on the errors of LRA (deterministic estimates)

Lemma 3.1. [The impact of pre-multiplication on LRA errors.] *Suppose that Algorithm 2.3 outputs a matrix XY for $Y = (FX)^+FM$ and that $m \ge k \ge l = \mathrm{rank}(X)$. Then*

$$M - XY = W(M - XX^+M) \text{ for } W = I_m - X(FX)^+F, \tag{3.5}$$

$$|M - XY| \le |W|\ |M - XX^+M|,\ |W| \le |I_m| + |X|\ |F|\ |(XF)^+|. \tag{3.6}$$

Proof. Recall that $Y = (FX)^+FM$ and notice that $(FX)^+FX = I_l$ if $k \ge l = \mathrm{rank}(FX)$. Therefore $Y = X^+M + (FX)^+F(M - XX^+M)$. Consequently (3.5) and (3.6) hold. □

We bounded the norm $|M - XX^+M|$ in the previous subsection; next we bound the norms $|(FX)^+|$ and $|W|$ at sub-linear cost for $kl \ll n$, a fixed orthogonal X, and proper choice of sparse F.

Theorem 3.2. *[P00, Algorithm 1] for a real $h > 1$ applied to an $m \times l$ orthogonal matrix X performs $O(ml^2)$ flops and outputs an $l \times m$ sub-permutation matrix F such that $||(FX)^+|| \le \sqrt{(m-l)lh^2+1}$, and $||W|| \le 1 + \sqrt{(m-l)lh^2+1}$, for $W = I_m + X(FX)^+F$ of (3.5) and any fixed $h > 1$; $||W|| \approx \sqrt{ml}$ for $m \gg l$ and $h \approx 1$.*

[P00, Algorithm 1] outputs $l \times m$ matrix F. One can strengthen deterministic bounds on the norm $|W|$ by computing $k \times m$ sub-permutation matrices F for k of at least order l^2.

Theorem 3.3. *For k of at least order l^2 and a fixed orthogonal multiplier X compute a $k \times m$ sub-permutation multiplier F by means of deterministic algorithms by Osinsky, running at sub-linear cost and supporting [O18, equation (1)]. Then $||W|| \le 1 + ||(FX)^+|| = O(l)$ for W of (3.5).*

4 Accuracy of sub-linear cost dual LRA algorithms

Next we estimate the output errors of Algorithm 2.1 for a fixed orthogonal matrix H and two classes of random inputs of low numerical rank, in particular for perturbed factor-Gaussian inputs of Definition A.1. These estimates formally support the observed accuracy of Range Finder with various dense multipliers (see [HMT11, Section 7.4], and the bibliography therein), but also with sparse multipliers, with which Algorithms 2.3 and 2.4 run at sub-linear cost. By applying the results of the previous section we extend these upper estimates for output accuracy to variations of Algorithm 2.3 that run at sub-linear cost; then we extend them to Algorithm 2.4 by means of transposition of an input matrix.
Our estimates involve the norms of a Gaussian matrix and its pseudo inverse (cf. Appendix A.4).

Definition 4.1. *A matrix is* Gaussian *if its entries are iid Gaussian variables.* $\mathcal{G}^{p\times q}$ *is the class of* $p\times q$ *Gaussian matrices.* $\nu_{p,q}=|G|$, $\nu_{\mathrm{sp},p,q}=||G||$, $\nu_{F,p,q}=||G||_F$, $\nu^+_{p,q}=|G^+|$, $\nu^+_{\mathrm{sp},p,q}=||G^+||$, *and* $\nu^+_{F,p,q}=||G^+||_F$ *for a matrix* $G\in\mathcal{G}^{p\times q}$. *[*$\nu_{p,q}=\nu_{q,p}$ *and* $\nu^+_{p,q}=\nu^+_{q,p}$, *for all pairs of* p *and* q.*]*

Theorem 4.1. [Non-degeneration of a Gaussian Matrix.] *Let* $F\in\mathcal{G}^{r\times p}$, $H\in\mathcal{G}^{q\times r}$, $M\in\mathbb{R}^{p\times q}$ *and* $r\le\mathrm{rank}(M)$. *Then the matrices* F, H, FM, *and* MH *have full rank* r *with probability 1.*

Assumption 4.1. *We simplify the statements of our results by assuming that a Gaussian matrix has full rank and ignoring the probability 0 of its degeneration.*

4.1 Output errors of Range Finder for a perturbed factor-Gaussian input

Theorem 4.2. [Errors of Range Finder for a perturbed factor-Gaussian matrix.] *Apply Algorithm 2.1 to a perturbation* $M=\tilde{M}+E$ *of a right* $m\times n$ *factor-Gaussian matrix* $\tilde{M}$ *of rank* r *such that*

$$\alpha:=||E||_F/(\sigma_r(M)-\sigma_{r+1}(M))\le 0.2 \text{ and } \xi:=4\alpha\phi<1 \text{ for } \phi=\nu_{\mathrm{sp},r,n}\nu^+_{\mathrm{sp},r,l}||H||_F||H^+||. \quad (4.1)$$

(i) Then

$$||M-XY||^2\le\Big(1+\Big(\frac{\phi}{1-\xi}\Big)^2\Big)\sigma^2_{r+1}(M).$$

(ii) Let $\xi\le 1/2$ *with a probability close to 1 and let the integer* $l-r$ *be at least moderately large. Then with a probability close to 1*

$$\mathbb{E}||M-XY||\le\theta\sigma_{r+1}(M) \text{ for } \theta^2\approx 1+\Big((2e||H||_F||H^+||(\sqrt{nl}+\sqrt{rl})/(l-r)\Big)^2,$$

which is close to $1+(2e||H||_F||H^+||)^2n/l$ *if* $r\ll l$. *Here and hereafter* $e:=2.71828182\dots$

Proof. We prove claim (i) in Appendix B. Let us deduce claim (ii). Recall from Theorems A.4 and A.5 that the random variables $\nu_{\mathrm{sp},r,n}$ and $\nu^+_{\mathrm{sp},r,l}$ are strongly concentrated about their expected values $\mathbb{E}(\nu_{\mathrm{sp},r,n})=\sqrt{n}+\sqrt{r}$ and $\mathbb{E}(\nu^+_{\mathrm{sp},r,l})=\frac{e\sqrt{l}}{l-r}$, respectively. Substitute these equations into the bound of claim (i), apply Jensen's inequality, and deduce claim (ii) of the theorem. □

4.2 Output errors of Range Finder near a matrix with a random singular space

Next we prove similar estimates under an alternative randomization model for dual LRA.

Theorem 4.3. [Errors of Range Finder for an input with a perturbed random singular space.] *Let the matrix V_1 in Theorem 3.1 be the $n \times r$ Q factor in a QR factorization of a normalized $n \times r$ Gaussian matrix $G_{n,r}$ and let the multiplier H be any $n \times l$ matrix of full rank $l \geq r$.*

(i) Then for $\nu_{r,n}$ and $\nu^+_{r,l}$ of Definition 4.1 it holds that

$$|M - XY|/\bar{\sigma}_{r+1}(M) \leq \phi_{r,l,n} := (1 + (\nu_{n,r}\nu^+_{r,l}|H^+|)^2)^{1/2}.$$

(ii) For a large or reasonably large integer $l-r$, the random variable $\phi_{r,l,n}$ is strongly concentrated about its expected values

$$\mathbb{E}(\phi_{\mathrm{sp},r,l,n}) = \Big(1 + \Big(\frac{\sqrt{nl} + \sqrt{rl})e||H^+||}{l-r}\Big)^2\Big)^{1/2} \text{ and } \mathbb{E}(\phi_{F,r,l,n}) = \Big(1 + \Big(\frac{nr^2||H^+||_F}{l-r-1}\Big)^2\Big)^{1/2},$$

which turn into

$$\mathbb{E}(\phi_{\mathrm{sp},r,l,n}) \approx \Big(1 + \frac{(e||H^+||)^2 n}{l}\Big)^{1/2} \text{ and } \mathbb{E}(\phi_{F,r,l,n}) \approx \Big(1 + \frac{(e||H^+||)^2 n}{l}\Big)^{1/2}$$

if $r \ll l$. Here $||H^+|| = 1$ and $||H^+||_F = l$ if the matrix H is orthogonal.

Proof. Write

$$G_{n,r} = V_1 R,\ V_1 = G_{n,r}R^{-1},\ R = V_1^* G_{n,r}, \tag{4.2}$$

and so $V_1^* = (R^*)^{-1}G_{r,n}$ and $V_1^* H = (R^*)^{-1}G_{r,n}H$.

Let $H = U_H \Sigma_H V_H^*$ be SVD. Then $G_{r,n}U_H = G_{r,l}$ (cf. Lemma A.4), and hence

$$V_1^* H = (R^*)^{-1} G_{r,l} \Sigma_H V_H^*.$$

Therefore $|(V_1^* H)^+| \leq |R|\ \nu^+_{r,l}|\Sigma_H^{-1}|$ (apply Lemma A.1 and recall that V_H is an orthogonal matrix).

Substitute $|\Sigma_H^{-1}| = |H^+|$ and obtain $|(V_1^* H)^+| \leq \nu^+_{r,l}|H^+|\ |R|$.

Deduce from (4.2) that the matrices R and $G_{n,r}$ share all their singular values. Therefore $|R| = \nu_{n,r}$, and so $|(V_1^* H)^+| \leq \nu_{n,r}\nu^+_{r,l}|H^+|$.

By combining this bound with (3.4) prove claim (i) of the theorem.

Already for a reasonably large ratio l/r the random variable $\phi^2_{r,l,n} = 1 + (\nu_{n,r}\nu^+_{r,l}|H^+|)^2$ is strongly concentrated about its expected values

$$\mathbb{E}(\phi^2_{\mathrm{sp},r,l,n}) = 1 + \Big(\frac{(\sqrt{nl} + \sqrt{rl})e||H^+||}{l-r}\Big)^2 \text{ and } \mathbb{E}(\phi^2_{F,r,l,n}) = 1 + \Big(\frac{nr^2}{l-r-1}||H^+||_F\Big)^2,$$

respectively (cf. Theorems A.4 and A.5), which are close to

$$1 + (e||H^+||)^2 n/l \text{ and } 1 + (nr^2||H^+||_F/l)^2,$$

respectively, if $r \ll l \leq n$. Apply Jensen's inequality and deduce claim (ii) of the theorem. □

Bound the output errors of Algorithms 2.3–2.4 by combining the estimates of this section and Section 3.2 and by transposing the input matrix M.

4.3 Impact of pre-multiplication in the case of Gaussian noise

Next we deduce randomized estimates for the impact of pre-multiplication in the case where an input matrix M includes considerable additive white Gaussian noise,[6] which is a classical representation of natural noise in information theory, is widely adopted in signal and image processing, and in many cases properly represents errors of measurement and rounding (cf. [SST06]).

Theorem 4.4. *Suppose that the multipliers F and H are orthogonal and let the input matrix M be the sum of a fixed matrix A and a scaled Gaussian matrix E,*

$$M = A + E,\ \frac{1}{\lambda_E}\ E = G_{m,n} \in \mathcal{G}^{m\times n} \tag{4.3}$$

for a constant λ_E proportional to the norm $||E||$. Then for any pair of orthogonal multipliers F and H it holds that

$$|W| \le |I_m| + \lambda_E\ |M| \min\{\nu^+_{n-l,l}, \nu^+_{l,l}\},\quad |W| \le |I_m| + \lambda_E\ |M|\nu^+_{n-l,l} \text{ for } n \ge 2l, \tag{4.4}$$

$$\mathbb{E}\Big(\frac{||W||_F - \sqrt{m}}{\lambda_E\ ||M||_F}\Big) \le \frac{l}{n-2l-1},\ \mathbb{E}\Big(\frac{(||W|| - 1}{\lambda_E\ ||M||}\Big) \le \frac{e\sqrt{n}}{n-2l} \text{ for } n - l \ge l \ge 2. \tag{4.5}$$

Proof. Assumption (4.3) and Lemma A.4 together imply that FEH is a scaled Gaussian matrix: $\frac{1}{\lambda_E}FEH \in \mathcal{G}^{k\times l}$. Hence $FMH = FAH + \frac{1}{\lambda_E}G_{k,l}$. Apply Theorem A.3 and obtain that $|(FMH)^+| \le \lambda_E \min\{\nu^+_{n-l,l}, \nu^+_{l,l}\}$. Recall from (3.5) that $|W| \le |I_m| + |(FMH)^+|\ |M|$ since the multipliers F and H are orthogonal. Substitute the above bound on $|(FMH)^+|$ and obtain (4.4). Substitute equations $||I_m||_F = \sqrt{m}$ and $||I_m|| = 1$ and claim (iii) of Theorem A.5 and obtain (4.5). □

Remark 4.1. *For $k = l = \rho$, $S = T = I_k$, sub-permutation matrices F and H, and a nonsingular matrix FMH, Algorithms 2.3 and 2.4 output LRA in the form CUR where $C \in \mathbb{R}^{m\times\rho}$ and $R \in \mathbb{R}^{\rho\times n}$ are two submatrices made up of ρ columns and ρ rows of M and $U = (FMH)^{-1}$. [PLSZa] extends our current study to devising and analyzing algorithms for the computation of such CUR LRA in the case where k and l are arbitrary integers not exceeded by ρ.*

5 Multiplicative pre-processing and generation of multipliers

5.1 Multiplicative pre-processing for LRA

We proved that sub-linear cost variations of Algorithms 2.3 and 2.4 tend to output accurate LRA of random input whp. In the real world computations input matrices are not random, but we can randomize them by multiplying them by random matrices.

Algorithms 2.1 – 2.4 output accurate LRA whp if the multipliers are Gaussian, SRHT, SRFT or Rademacher's (cf. [HMT11, Sections 10 and 11], [T11], [CW09]), but multiplication by these matrices run at super-linear cost. Our heuristic recipe is to apply these algorithms with a small variety of sparse multipliers F_i and/or H_i, $i = 1, 2, \dots$, with which computational cost becomes sub-linear and then to monitor the accuracy of the output LRA by applying the criteria of the previous section, [PLa], and/or [PLSZa].

[6] Additive white Gaussian noise is statistical noise having a probability density function (PDF) equal to that of the Gaussian (normal) distribution. Additive white Gaussian noise is widely adopted in information theory and used in signal and image processing; in many cases it properly represents the errors of measurement and rounding (cf. [SST06]).

Various families of sparse multipliers have been proposed in [PLSZ16] and [PLSZ17]. One can readily complement these families with sub-permutation matrices and, say, sparse quasi Radmacher's multipliers (see [PLSZa]) and then combine these basic multipliers together by using orthogonalized sums, products or other lower degree polynomials of these matrices as multipliers (cf. [HMT11, Remark 4.6]).

Next we specify a particular family of sparse multipliers, which was highly efficient in our tests when we applied them both themselves and in combination with other sparse multipliers.

5.2 Generation of abridged Hadamard and Fourier multipliers

We define multipliers of this family by means of abridging the classical recursive processes of the generation of $n \times n$ SRHT and SRFT matrices for $n = 2^t$. These matrices are obtained from the $n \times n$ dense matrices H_n of *Walsh-Hadamard transform* (cf. [M11, Section 3.1]) and F_n of *discrete Fourier transform (DFT)* at n points (cf. [P01, Section 2.3]), respectively. Recursive representation in t recursive steps enables multiplication of the matrices H_n and F_n by a vector in $2tn$ additions and subtractions and $O(tn)$ flops, respectively.

We end these processes in d recursive steps for a fixed recursion depth d, $1 \le d \le t$, and obtain the *d-abridged Hadamard (AH) and Fourier (AF)* matrices $H_{d,d}$ and $F_{d,d}$, respectively, such that $H_{t,t} = H_n$ and $F_{t,t} = F_n$. Namely write $H_{d,0} = F_{d,0} = I_{n/2^d}$, $\mathbf{i} = \sqrt{-1}$, and $\omega_s = \exp(2\pi\mathbf{i}/s)$ denoting a primitive s-th root of 1, and then specify two recursive processes:

$$H_{d,0} = I_{n/2^d},\ H_{d,i+1} = \begin{pmatrix} H_{d,i} & H_{d,i} \\ H_{d,i} & -H_{d,i} \end{pmatrix} \text{ for } i = 0, 1, \dots, d-1, \tag{5.1}$$

$$F_{d,i+1} = \widehat{P}_{i+1} \begin{pmatrix} F_{d,i} & F_{d,i} \\ F_{d,i}\widehat{D}_{i+1} & -F_{d,i}\widehat{D}_{i+1} \end{pmatrix},\ \widehat{D}_{i+1} = \operatorname{diag}\left(\omega_{2^{i+1}}^{j}\right)_{j=0}^{2^i-1},\ i = 0, 1, \dots, d-1, \tag{5.2}$$

where $\widehat{P}_i$ denotes the $2^i \times 2^i$ matrix of odd/even permutations such that $\widehat{P}_i\mathbf{u} = \mathbf{v}$, $\mathbf{u} = (u_j)_{j=0}^{2^i-1}$, $\mathbf{v} = (v_j)_{j=0}^{2^i-1}$, $v_j = u_{2j}$, $v_{j+2^{i-1}} = u_{2j+1}$, $j = 0, 1, \dots, 2^{i-1} - 1$.[7]

For any fixed pair of d and i, each of the matrices $H_{d,i}$ (resp. $F_{d,i}$) is orthogonal (resp. unitary) up to scaling and has 2^d nonzero entries in every row and column. Now make up multipliers F and H of $k \times m$ and $n \times l$ submatrices of $F_{d,d}$ and $H_{d,d}$, respectively. Then in view of sparseness of $F_{d,d}$ or $H_{d,d}$, we can compute the products FM and MH by using $O(kn2^d)$ and $O(lm2^d)$ flops, respectively, and they are just additions or subtractions in the case of submatrices of $H_{d,d}$.

By combining random permutation with either Rademacher's diagonal scaling for AH matrices $H_{d,d}$ or or random unitary diagonal scaling for AF matrices $F_{d,d}$, we obtain the *d–Abridged Scaled and Permuted Hadamard (ASPH)* matrices, PDH_n, and *d–Abridged Scaled and Permuted Fourier (ASPF)* $n \times n$ matrices, PDF_n, where P and D are two matrices of permutation and diagonal scaling. Likewise define the families of ASH, ASF, APH, and APF matrices, $DH_{n,d}$, $DF_{n,d}$, $H_{n,d}P$, and $F_{n,d}P$, respectively. Each random permutation or scaling contributes up to n random parameters. We can involve more random parameters by applying random permutation and scaling to the intermediate matrices $H_{d,i}$ and $F_{d,i}$ for $i = 0, 1, \dots, d$.

Now the first k rows for $r \le k \le n$ or first l columns for $r \le l \le n$ of $H_{d,d}$ and $F_{d,d}$ form a d-abridged Hadamard or Fourier multiplier, which turns into a SRHT or SRFT matrix, respectively, for $d = t$. For k and l of order $r\log(r)$ Algorithm 2.1 with a SRHT or SRFT multiplier outputs whp accurate LRA of any matrix M admitting LRA (see [HMT11, Section 11]), but in our tests the

[7] For $d = t$ this is a decimation in frequency (DIF) radix-2 representation of FFT. Transposition turns it into the decimation in time (DIT) radix-2 representation of FFT.

output was consistently accurate even with sparse abridged SRHT or SRFT multipliers computed just in three recursive steps.

6 Numerical tests

In this section we cover our tests of dual sub-linear cost variants of Algorithm 2.1. The tests for Tables 6.1–6.4 have been performed by using MatLab on a Dell computer with the Intel Core 2 2.50 GHz processor and 4G memory running Windows 7; the standard normal distribution function randn of MATLAB has been applied in order to generate Gaussian matrices. The MATLAB function "svd()" has been applied in order to calculate the ξ-rank, i.e., the number of singular values exceeding ξ for $\xi = 10^{-5}$ in Sections 6.2 and 6.3 and $\xi = 10^{-6}$ in Section 6.4. The tests for Tables 6.5–6.7 have been performed on a 64-bit Windows machine with an Intel i5 dual-core 1.70 GHz processor by using custom programmed software in C^{++} and compiled with LAPACK version 3.6.0 libraries.

6.1 Input matrices for LRA

We generated the following classes of input matrices M for testing LRA algorithms.

Class I: Perturbed $n \times n$ factor-Gaussian matrices with expected rank r, $M = G_1 * G_2 + 10^{-10} G_3$, for three Gaussian matrices $G_1 \in \mathcal{G}^{n \times r}$, $G_2 \in \mathcal{G}^{r \times n}$, and $G_3 \in \mathcal{G}^{n \times n}$.

Class II: $M = U_M \Sigma_M V_M^*$, for U_M and V_M being the Q factors of the thin QR orthogonalization of $n \times n$ Gaussian matrices and $\Sigma_M = \mathrm{diag}(\sigma_j)_{j=1}^n$; $\sigma_j = 1/j$, $j = 1, \dots, r$, $\sigma_j = 10^{-10}$, $j = r+1, \dots, n$ (cf. [H02, Section 28.3]), and $n = 256, 512, 1024$. (Hence $||M|| = 1$ and $\kappa(M) = 10^{10}$.)

Class III: (i) The matrices M of the discretized single-layer Laplacian operator of [HMT11, Section 7.1]: $[S\sigma](x) = c \int_{\Gamma_1} \log|x - y| \sigma(y) dy, x \in \Gamma_2$, for two circles $\Gamma_1 = C(0,1)$ and $\Gamma_2 = C(0,2)$ on the complex plane. We arrived at a matrix $M = (m_{ij})_{i,j=1}^n$, $m_{i,j} = c \int_{\Gamma_{1,j}} \log|2\omega^i - y| dy$ for a constant c, $||M|| = 1$ and the arc $\Gamma_{1,j}$ of Γ_1 defined by the angles in the range $[\frac{2j\pi}{n}, \frac{2(j+1)\pi}{n}]$.

(ii) The matrices that approximate the inverse of a large sparse matrix obtained from a finite-difference operator of [HMT11, Section 7.2].

Class IV: The dense matrices of six classes with smaller ratios of "numerical rank/n" from the built-in test problems in Regularization Tools, which came from discretization (based on Galerkin or quadrature methods) of the Fredholm Integral Equations of the first kind:[8]

baart: Fredholm Integral Equation of the first kind,
shaw: one-dimensional image restoration model,
gravity: 1-D gravity surveying model problem,
wing: problem with a discontinuous solution,
foxgood: severely ill-posed problem,
inverse Laplace: inverse Laplace transformation.

[8] See http://www.math.sjsu.edu/singular/matrices and http://www2.imm.dtu.dk/∼pch/Regutools
For more details see Chapter 4 of the Regularization Tools Manual at
http://www.imm.dtu.dk/∼pcha/Regutools/RTv4manual.pdf

6.2 Tests for LRA of inputs of class II (generated via SVD)

Next we present the results of our tests of Algorithm 2.1 applied to matrices M of class II.

Table 6.1 shows the average output error norms over 1000 tests for the matrices M for each pair of n and r, $n = 256, 512, 1024$, $r = 8, 32$, and for either 3-AH multipliers or 3-ASPH multipliers, both defined by Hadamard recursion (5.2), for $d = 3$.

Table 6.1: Error norms for SVD-generated inputs and 3-AH and 3-ASPH multipliers

n	r	3-AH	3-ASPH
256	8	2.25e-08	2.70e-08
256	32	5.95e-08	1.47e-07
512	8	4.80e-08	2.22e-07
512	32	6.22e-08	8.91e-08
1024	8	5.65e-08	2.86e-08
1024	32	1.94e-07	5.33e-08

Table 6.2 displays the average error norms in the case of multipliers B of two families defined below, both generated from the *Basic Set* of $n \times n$ 3-APF multipliers defined by three Fourier recursive steps of equation (5.2), for $d = 3$, with no scaling, but with a random column permutation.

For multipliers B we used the $n \times r$ leftmost blocks of (1) either $n \times n$ matrices from the Basic Set or (2) the product of two such matrices. Both tables show similar tests results.

In sum, for all classes of input pairs M and B and all pairs of integers n and r, Algorithm 2.1 with our pre-processing has consistently output approximations to rank-r input matrices with the average error norms ranged from 10^{-7} or 10^{-8} to about 10^{-9} in all our tests.

Table 6.2: Error norms for SVD-generated inputs of class II and multipliers of two classes

n	r	class 1	class 2
256	8	5.94e-09	2.64e-08
256	32	2.40e-08	8.23e-08
512	8	1.11e-08	2.36e-09
512	32	1.61e-08	1.61e-08
1024	8	5.40e-09	6.82e-08
1024	32	2.18e-08	8.72e-08

6.3 Tests for LRA of input matrices of class III (from [HMT11])

In Tables 6.3 and 6.4 we show the results of the application of Algorithm 2.1 to the matrices of class III with multipliers B being the $n \times r$ leftmost submatrices of $n \times n$ *Gaussian multipliers, Abridged permuted Fourier (3-APF) multipliers, and Abridged permuted Hadamard (3-APH) multipliers.*

Then again we defined each 3-APF and 3-APH matrix by applying three recursive steps of equation (5.2) followed by a single random column permutation.

We performed 1000 tests for every class of pairs of $n \times n$ or $m \times n$ matrices of classes III(i) or III(ii), respectively, and $n \times r$ multipliers for every fixed triple of m, n, and r or pair of n and r.

Tables 6.3 and 6.4 display the resulting data for the error norm $||UV - M||$.

Table 6.3: LRA of Laplacian matrices of class III(i)

n	multiplier	r	mean	std
200	Gaussian	3.00	1.58e-05	1.24e-05
200	3-APF	3.00	8.50e-06	5.15e-15
200	3-APH	3.00	2.18e-05	6.48e-14
400	Gaussian	3.00	1.53e-05	1.37e-06
400	3-APF	3.00	8.33e-06	1.02e-14
400	3-APH	3.00	2.18e-05	9.08e-14
2000	Gaussian	3.00	2.10e-05	2.28e-05
2000	3-APF	3.00	1.31e-05	6.16e-14
2000	3-APH	3.00	2.11e-05	4.49e-12
4000	Gaussian	3.00	2.18e-05	3.17e-05
4000	3-APF	3.00	5.69e-05	1.28e-13
4000	3-APH	3.00	3.17e-05	8.64e-12

Table 6.4: LRA of the matrices of discretized finite-difference operator of class III(ii)

m	n	multiplier	r	mean	std
88	160	Gaussian	5.00	1.53e-05	1.03e-05
88	160	3-APF	5.00	4.84e-04	2.94e-14
88	160	3-APH	5.00	4.84e-04	5.76e-14
208	400	Gaussian	43.00	4.02e-05	1.05e-05
208	400	3-APF	43.00	1.24e-04	2.40e-13
208	400	3-APH	43.00	1.29e-04	4.62e-13
408	800	Gaussian	64.00	6.09e-05	1.75e-05
408	800	3-APF	64.00	1.84e-04	6.42e-12
408	800	3-APH	64.00	1.38e-04	8.65e-12

6.4 Tests with additional families of multipliers

In the next three tables we display the output error norms of Algorithm 2.1 applied to the input matrices of classes II–IV with six additional families of multipliers to be specified later.

In particular we used 1024×1024 SVD-generated input matrices of class II having numerical rank $r = 32$, 400×400 Laplacian input matrices of class III(i) having numerical rank $r = 36$, 408×800 matrices having numerical rank $r = 145$ and representing finite-difference inputs of class III(ii), and 1000×1000 matrices of class IV (from the San Jose University database).

Then again we repeated the tests 100 times for each class of input matrices and each size of an input and a multiplier, and we display the resulting average error norms in Tables 6.5–6.7.

We generated our $n \times (r + p)$ multipliers for random $p = 1, 2, \dots 21$ by using *3-ASPH, 3-APH, and Random permutation matrices.*

We obtained every 3-APH and every 3-ASPH matrix by applying three Hadamard's recursive steps (5.1) followed by random column permutation defined by random permutation of the integers from 1 to n inclusive. While generating a 3-ASPH matrix we also applied random scaling with a diagonal matrix $D = \mathrm{diag}(d_i)_{i=1}^{n}$ where we have chosen the values of random iid variables d_i under the uniform probability distribution from the set $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$.

We used the following families of multipliers: (0) Gaussian (for control), (1) sum of a 3-ASPH and a permutation matrix, (2) sum of a 3-ASPH and two permutation matrices, (3) sum of a 3-ASPH and three permutation matrices, (4) sum of a 3-APH and three permutation matrices, and (5) sum of a 3-APH and two permutation matrices.

The test results in Tables 6.5–6.7 show high output accuracy with error norms in the range from about 10^{-6} to 10^{-9} with the exception of multiplier families 1–5 for the inverse Laplace input matrix, in which case the range was from about 10^{-3} to 10^{-5}.

The numbers in parentheses in the first line of Tables 6.6 and 6.7 show the numerical rank of input matrices.

	SVD-generated Matrices		Laplacian Matrices		Finite Difference Matrices	
Family No.	Mean	Std	Mean	Std	Mean	Std
Family 0	4.97e-09	5.64e-09	1.19e-07	1.86e-07	2.44e-06	2.52e-06
Family 1	4.04e-09	3.17e-09	2.32e-07	2.33e-07	5.99e-06	7.51e-06
Family 2	5.49e-09	7.15e-09	1.91e-07	2.13e-07	3.74e-06	4.49e-06
Family 3	6.22e-09	7.47e-09	1.66e-07	1.82e-07	2.64e-06	3.34e-06
Family 4	3.96e-09	3.21e-09	1.91e-07	1.95e-07	1.90e-06	2.48e-06
Family 5	4.05e-09	3.01e-09	1.81e-07	2.01e-07	2.71e-06	3.33e-06

Table 6.5: Relative error norms in tests for matrices of classes II and III

Appendix

A Background on matrix computations

A.1 Some definitions

- An $m \times n$ matrix M is *orthogonal* if $M^*M = I_n$ or $MM^* = I_m$.

	wing (4)		baart (6)		inverse Laplace (25)	
Family No.	Mean	Std	Mean	Std	Mean	Std
Family 0	1.20E-08	6.30E-08	1.82E-09	1.09E-08	2.72E-08	7.50E-08
Family 1	2.00E-09	1.34E-08	2.46E-09	1.40E-08	1.21E-03	4.13E-03
Family 2	7.96E-09	4.18E-08	5.31E-10	3.00E-09	6.61E-04	2.83E-03
Family 3	3.01E-09	2.23E-08	5.55E-10	2.74E-09	3.35E-04	1.81E-03
Family 4	2.27E-09	1.07E-08	2.10E-09	1.28E-08	3.83E-05	1.66E-04
Family 5	3.66E-09	1.57E-08	1.10E-09	5.58E-09	3.58E-04	2.07E-03

Table 6.6: Relative error norms for input matrices of class IV (of San Jose University database)

	foxgood (10)		shaw (12)		gravity (25)	
Family No.	Mean	Std	Mean	Std	Mean	Std
Family 0	1.56E-07	4.90E-07	2.89E-09	1.50E-08	2.12E-08	4.86E-08
Family 1	3.70E-07	2.33E-06	1.79E-08	8.70E-08	3.94E-08	1.14E-07
Family 1	1.76E-06	3.76E-06	1.46E-08	5.92E-08	4.81E-08	1.26E-07
Family 2	9.77E-07	1.71E-06	1.11E-08	6.67E-08	2.82E-08	8.37E-08
Family 3	7.16E-07	1.14E-06	1.87E-08	1.04E-07	5.70E-08	2.52E-07
Family 4	7.52E-07	1.24E-06	4.77E-09	1.79E-08	6.32E-08	1.99E-07
Family 5	9.99E-07	2.27E-06	1.03E-08	3.81E-08	3.94E-08	1.00E-07

Table 6.7: Relative error norms for input matrices of class IV (of San Jose University database)

- For a matrix $M = (m_{i,j})_{i,j=1}^{m,n}$ and two sets $\mathcal{I} \subseteq \{1,\dots,m\}$ and $\mathcal{J} \subseteq \{1,\dots,n\}$, define the submatrices $M_{\mathcal{I},:} := (m_{i,j})_{i\in\mathcal{I};j=1,\dots,n}$, $M_{:,\mathcal{J}} := (m_{i,j})_{i=1,\dots,m;j\in\mathcal{J}}$, and $M_{\mathcal{I},\mathcal{J}} := (m_{i,j})_{i\in\mathcal{I};j\in\mathcal{J}}$.
- $\mathrm{rank}(M)$ denotes the *rank* of a matrix M. ϵ-$\mathrm{rank}(M)$ is $\mathrm{argmin}_{|E|\le\epsilon|M|}\,\mathrm{rank}(M+E)$, called *numerical rank*, $\mathrm{nrank}(M)$, if ϵ is small in context.
- M_r is the *rank-r truncation*, obtained from M by setting $\sigma_j(M) = 0$ for $j > r$.
- $\kappa(M) = ||M||\ ||M^+||$ is the spectral *condition number* of M.

A.2 Auxiliary results

Lemma A.1. [The norm of the pseudo inverse of a matrix product.] *Suppose that $A \in \mathbb{R}^{k\times r}$, $B \in \mathbb{R}^{r\times l}$ and the matrices A and B have full rank $r \le \min\{k,l\}$. Then $|(AB)^+| \le |A^+|\ |B^+|$.*

Lemma A.2. (The norm of the pseudo inverse of a perturbed matrix, [B15, Theorem 2.2.4].) *If $\mathrm{rank}(M+E) = \mathrm{rank}(M) = r$ and $\eta = ||M^+||\ ||E|| < 1$, then*

$$\frac{1}{\sqrt{r}}||(M+E)^+|| \le ||(M+E)^+|| \le \frac{1}{1-\eta}||M^+||.$$

Lemma A.3. (The impact of a perturbation of a matrix on its singular values, [GL13, Corollary 8.6.2].) *For $m \ge n$ and a pair of $m \times n$ matrices M and $M+E$ it holds that*

$$|\sigma_j(M+E) - \sigma_j(M)| \le ||E|| \text{ for } j = 1,\dots,n.$$

Theorem A.1. (The impact of a perturbation of a matrix on its top singular spaces, [GL13, Theorem 8.6.5].) *Let $g =: \sigma_r(M) - \sigma_{r+1}(M) > 0$ and $||E||_F \le 0.2g$. Then for the left and right singular spaces associated with the r largest singular values of the matrices M and $M+E$, there exist orthogonal matrix bases $B_{r,\mathrm{left}}(M)$, $B_{r,\mathrm{right}}(M)$, $B_{r,\mathrm{left}}(M+E)$, and $B_{r,\mathrm{right}}(M+E)$ such that*

$$\max\{||B_{r,\mathrm{left}}(M+E) - B_{r,\mathrm{left}}(M)||_F, ||B_{r,\mathrm{right}}(M+E) - B_{r,\mathrm{right}}(M)||_F\} \le 4\frac{||E||_F}{g}.$$

For example, if $\sigma_r(M) \ge 2\sigma_{r+1}(M)$, which implies that $g \ge 0.5\ \sigma_r(M)$, and if $||E||_F \le 0.1\ \sigma_r(M)$, then the upper bound on the right-hand side is approximately $8||E||_F/\sigma_r(M)$.

A.3 Gaussian and factor-Gaussian matrices of low rank and low numerical rank

Lemma A.4. [Orthogonal invariance of a Gaussian matrix.] *Suppose that k, m, and n are three positive integers, $k \le \min\{m,n\}$, $G_{m,n} \in \mathcal{G}^{m\times n}$, $S \in \mathbb{R}^{k\times m}$, $T \in \mathbb{R}^{n\times k}$, and S and T are orthogonal matrices. Then SG and GT are Gaussian matrices.*

Definition A.1. [Factor-Gaussian matrices.] *Let $r \le \min\{m,n\}$ and let $\mathcal{G}^{m\times n}_{r,B}$, $\mathcal{G}^{m\times n}_{A,r}$, and $\mathcal{G}^{m\times n}_{r,C}$ denote the classes of matrices $G_{m,r}B$, $AG_{r,n}$, and $G_{m,r}CG_{r,n}$, respectively, which we call* left, right, *and* two-sided factor-Gaussian matrices of rank r, *respectively, provided that $G_{p,q}$ denotes a $p\times q$ Gaussian matrix, $A \in \mathbb{R}^{m\times r}$, $B \in \mathbb{R}^{r\times n}$, and $C \in \mathbb{R}^{r\times r}$, and A, B and C are well-conditioned matrices of full rank r.*

Theorem A.2. *The class $\mathcal{G}^{m\times n}_{r,C}$ of two-sided $m\times n$ factor-Gaussian matrices $G_{m,r}\Sigma G_{r,n}$ does not change if in its definition we replace the factor C by a well-conditioned diagonal matrix $\Sigma = (\sigma_j)_{j=1}^r$ such that $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r > 0$.*

Proof. Let $C = U_C\Sigma_C V_C^*$ be SVD. Then $A = G_{m,r}U_C \in \mathcal{G}^{m\times r}$ and $B = V_C^*G_{r,n} \in \mathcal{G}^{r\times n}$ by virtue of Lemma A.4, and so $G_{m,r}CG_{r,n} = A\Sigma_C B$ for $A \in \mathcal{G}^{m\times r}$ and $B \in \mathcal{G}^{r\times n}$. □

Definition A.2. The relative norm of a perturbation of a Gaussian matrix *is the ratio of the perturbation norm and the expected value of the norm of the matrix (estimated in Theorem A.4).*

We refer to all three matrix classes above as *factor-Gaussian matrices of rank r*, to their perturbations within a relative norm bound ϵ as *factor-Gaussian matrices of ϵ-rank r*, and to their perturbations within a small relative norm as *factor-Gaussian matrices of numerical rank r* to which we also refer as *perturbations of factor-Gaussian matrices.*

Clearly $||(A\Sigma)^+|| \le ||\Sigma^{-1}||\,||A^+||$ and $||(\Sigma B)^+|| \le ||\Sigma^{-1}||\,||B^+||$ for a two-sided factor-Gaussian matrix $M = A\Sigma B$ of rank r of Definition A.1, and so whp such a matrix is both left and right factor-Gaussian of rank r.

Theorem A.3. *Let $M_{k,l} \in \mathbb{R}^{k\times l}$ and $G_{k,l} \in \mathcal{G}^{k\times l}$ for $k \ge l$. Then $|(M_{k,l}+G_{k,l})^+| \le \min\{\nu^+_{l,l}, \nu^+_{k-l,l}\}$.*

Proof. Let $M_{k,l} = U\Sigma V^*$ be *full SVD* such that $U \in \mathbb{R}^{k\times k}$, $V \in \mathbb{R}^{l\times l}$, U and V are orthogonal matrices, $\Sigma = (D \mid O_{l,k-l})^T$, and D is an $l\times l$ diagonal matrix. Write $W_{k,l} := U^*(M_{k,l}+G_{k,l})V$ and observe that $U^*M_{k,l}V = \Sigma$ and $U^*G_{k,l}V \in \mathcal{G}^{k\times l}$ by virtue of Lemma A.4. Hence $W^T_{k,l} = (D + G_{l,l} \mid G_{l,k-l})$, and so $|W^+_{k,l}| \le \min\{|(D+G_{l,l})^+|, \nu^+_{k-l,l}\}$. Now Theorem A.3 follows because $|(M_{k,l}+G_{k,l})^+| = |W^+_{k,l}|$ and because $|(D+G_{l,l})^+| \le \nu^+_{l,l}$ by virtue of claim (iv) of Theorem A.5. □

A.4 Norms of a Gaussian matrix and its pseudo inverse

$\Gamma(x) = \int_0^\infty \exp(-t)t^{x-1}dt$ denotes the Gamma function.

Theorem A.4. [Norms of a Gaussian matrix. See [DS01, Theorem II.7] and our Definition 4.1.]
(i) Probability$\{\nu_{\mathrm{sp},m,n} > t + \sqrt{m} + \sqrt{n}\} \le \exp(-t^2/2)$ *for $t \ge 0$,* $\mathbb{E}(\nu_{\mathrm{sp},m,n}) \le \sqrt{m} + \sqrt{n}$.
(ii) $\nu_{F,m,n}$ is the χ-function, with $\mathbb{E}(\nu_{F,m,n}) = mn$ and probability density $\frac{2x^{n-i}\exp(-x^2/2)}{2^{n/2}\Gamma(n/2)}$.

Theorem A.5. [Norms of the pseudo inverse of a Gaussian matrix (see Definition 4.1).]
(i) Probability $\{\nu^+_{\mathrm{sp},m,n} \ge m/x^2\} < \frac{x^{m-n+1}}{\Gamma(m-n+2)}$ *for $m \ge n \ge 2$ and all positive x,*
(ii) Probability $\{\nu^+_{F,m,n} \ge t\sqrt{\frac{3n}{m-n+1}}\} \le t^{n-m}$ *and* Probability $\{\nu^+_{\mathrm{sp},m,n} \ge t\frac{e\sqrt{m}}{m-n+1}\} \le t^{n-m}$ *for all $t \ge 1$ provided that $m \ge 4$,*

(iii) $\mathbb{E}((\nu^+_{F,m,n})^2) = \frac{n}{m-n-1}$ *and* $\mathbb{E}(\nu^+_{\mathrm{sp},m,n}) \le \frac{e\sqrt{m}}{m-n}$ *provided that* $m \ge n+2 \ge 4$,

(iv) Probability $\{\nu^+_{\mathrm{sp},n,n} \ge x\} \le \frac{2.35\sqrt{n}}{x}$ *for* $n \ge 2$ *and all positive* x, *and furthermore* $||M_{n,n} + G_{n,n}||^+ \le \nu_{n,n}$ *for any* $n \times n$ *matrix* $M_{n,n}$ *and an* $n \times n$ *Gaussian matrix* $G_{n,n}$.

Proof. See [CD05, Proof of Lemma 4.1] for claim (i), [HMT11, Proposition 10.4 and equations (10.3) and (10.4)] for claims (ii) and (iii), and [SST06, Theorem 3.3] for claim (iv). □

Theorem A.5 implies reasonable probabilistic upper bounds on the norm $\nu^+_{m,n}$ even where the integer $|m-n|$ is close to 0; whp the upper bounds of Theorem A.5 on the norm $\nu^+_{m,n}$ decrease very fast as the difference $|m-n|$ grows from 1.

B Proof of Theorem 4.2

Readily obtain Theorem 4.2 by combining bound (3.2) with the following lemma.

Lemma B.1. *Under the assumptions of Theorem 4.2 let* Σ_2, C_1, *and* C_2 *denote the matrices of (3.3)–(3.2). Then* $||C_1^+|| \le \frac{1}{1-\xi}\, \nu_{\mathrm{sp},r,n}\nu^+_{\mathrm{sp},r,l}\kappa(H)$.

Proof. At first assume that $E = M - \tilde{M} = O$. Write $M = \tilde{M} := AB$ where $B = G_{r,n}$.

Let $A = U_A\Sigma_A V_A^*$ and $B = U_B\Sigma_B V_B^*$ be SVDs. Then $AB = U_A P V_B^*$ for $P = \Sigma_A V_A^* U_B \Sigma_B$, where P, Σ_A, V_A^*, U_B, and Σ_B are $r \times r$ matrices. Let $P = U_P\Sigma_P V_P^*$ be SVD. Write $U := U_A U_P$, $V^* := V_P^* V_B^*$, and $\tilde{C}_1 := C_1$ and observe that $U \in \mathbb{R}^{m\times r}$ and $V^* \in \mathbb{R}^{r\times n}$ are orthogonal matrices of sizes $m \times r$ and $r \times n$, respectively. Therefore $\tilde{M} = AB = U\Sigma_P V^*$ is SVD. Furthermore this is the top rank-r SVD of $\tilde{M}$ because $\mathrm{rank}(AB) = r$. Therefore $\tilde{C}_1 = V^*H = V_P^* V_B^* H$.

Recall that U_B and Σ_B are $r \times r$ matrices and deduce from SVD $B = U_B\Sigma_B V_B^*$ that $V_B^* = U_B^*\Sigma_B^{-1}B$. Substitute this expression and obtain that $\tilde{C}_1 = V_P^* U_B^* \Sigma_B^{-1} BH$. Notice that V_P and U_B are $r \times r$ orthogonal matrices, and so $|\tilde{C}_1^+| = |(\Sigma_B^{-1}BH)^+|$.

Deduce that $|\tilde{C}_1^+| \le |\Sigma_B|\, |(BH)^+|$ from Lemma A.1.

Recall that $B \in \mathcal{G}^{r\times n}$, substitute $|\Sigma_B| = |B| = \nu_{r,n}$, and obtain $|\tilde{C}_1^+| \le \nu_{r,n}|(BH)^+|$.

In the SVD $H = U_H\Sigma_H V_H^*$ the matrix V_H is orthogonal, Σ_H and V_H are $r \times r$ matrices, and $|\Sigma_H^{-1}| = |H^+|$. Lemma A.4 implies that $G_{r,n} := BU_H \in \mathcal{G}^{r\times l}$. Hence $|(BH)^+| \le \nu^+_{r,l}|H^+|$.

Substitute this inequality into the above bound on $|\tilde{C}_1^+|$ and obtain

$$|\tilde{C}_1^+| \le \nu_{r,n}\nu^+_{r,l}|H^+|. \tag{B.1}$$

Next suppose that $E = M - \tilde{M} \ne O$ but that $\alpha := ||E||_F/(\sigma_r(M) - \sigma_{r+1}(M)) \le 0.2$ (cf. (4.1)). Let $B_{r,\mathrm{right}}(M)$ and $B_{r,\mathrm{right}}(\tilde{M})$ denote two orthogonal matrix bases of the two linear spaces spanned by top r right singular vectors of M and $\tilde{M}$, respectively. Then deduce from Theorem A.1 that $||B_{r,\mathrm{right}}(M) - B_{r,\mathrm{right}}(\tilde{M})||_F \le 4\alpha$.

Define SVDs of $\tilde{M}$ and M where these matrix bases are $V_{\tilde{M}}$ and V_M, respectively. Then $|V_{\tilde{M}} - V_M| \le ||V_{\tilde{M}} - V_M||_F \le 4\alpha$, implying that $|\tilde{C} - C| = |V_{\tilde{M}}^* H - V_M^* H| \le 4\alpha|H|$.

It follows that $|\sigma_r(\tilde{C}_1) - \sigma_r(\tilde{C}_1)| \le 4\alpha||H||$ by virtue of Lemma A.3. Deduce that $||C_1^+||^{-1} = \sigma_r(C_1) \ge \sigma_r(\tilde{C}_1) - 4\alpha||H|| = ||(\tilde{C}_1)^+||^{-1} - 4\alpha||H||$.

Substitute(B.1), obtain $||C_1^+||^{-1} \ge (\nu_{\mathrm{sp},r,n}\nu^+_{\mathrm{sp},r,l}||H_r^+||)^{-1} - 4\alpha||H||$, and deduce Lemma B.1. □

C Small families of hard inputs for sub-linear cost LRA

Any sub-linear cost LRA algorithm fails on the following small families of LRA inputs.

Example C.1. *Define the following family of $m \times n$ matrices of rank 1 (we call them δ-*matrices*): $\{\Delta_{i,j},\ i = 1, \dots, m;\ j = 1, \dots, n\}$. Also include the $m \times n$ null matrix $O_{m,n}$ into this family. Now fix any sub-linear 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 sub-linear cost.*

The example actually covers randomized LRA algorithms as well. Indeed suppose that an LRA algorithm does not access a constant fraction of the entries of an input matrix. Then with a constant probability the algorithm misses an entry whose value greatly exceeds those of all other entries, in which case the algorithm can hardly approximate that entry closely. The paper [Pa] shows, however, that close LRA can be computed at sub-linear cost in two successive C-A iterations provided that we avoid choosing degenerating initial submatrix, which is precisely the problem with the matrix families of Example C.1. The sub-linear cost algorithms of [MW17] and [BW18] compute LRA of matrices of two important special matrix classes.

Acknowledgements: We were supported by NSF Grants CCF–1116736, CCF–1563942, CCF–1733834 and PSC CUNY Award 69813 00 48. N. L. Zamarashkin pointed us out reference [O18].

References

[B15] A. Björk, *Numerical Methods in Matrix Computations*, Springer, New York, 2015.

[BW18] A. Bakshi, D. P. Woodruff: Sublinear Time Low-Rank Approximation of Distance Matrices, *Procs. 32nd Intern. Conf. Neural Information Processing Systems (NIPS'18),* 3786–3796, Montral, Canada, 2018.

[CD05] Z. Chen, J. J. Dongarra, Condition Numbers of Gaussian Random Matrices, *SIAM. J. on Matrix Analysis and Applications*, **27**, 603–620, 2005.

[CLO16] C. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao and D. P. Mandic, "Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions", Foundations and Trends in Machine Learning: **9, 4-5**, 249–429, 2016. http://dx.doi.org/10.1561/2200000059

[CW09] K. L. Clarkson, D. P. Woodruff, Numerical Linear Algebra in the Streaming Model, *ACM Symp. Theory of Computing (STOC 2009)*, 205–214, ACM Press, NY, 2009.

[DMM08] P. Drineas, M.W. Mahoney, S. Muthukrishnan, Relative-error CUR Matrix Decompositions, *SIAM Journal on Matrix Analysis and Applications*, **30, 2**, 844–881, 2008.

[DS01] K. R. Davidson, S. J. Szarek, Local Operator Theory, Random Matrices, and Banach Spaces, in *Handbook on the Geometry of Banach Spaces* (W. B. Johnson and J. Lindenstrauss editors), pages 317–368, North Holland, Amsterdam, 2001.

[E88] A. Edelman, Eigenvalues and Condition Numbers of Random Matrices, *SIAM J. on Matrix Analysis and Applications*, **9**, **4**, 543–560, 1988.

[ES05] A. Edelman, B. D. Sutton, Tails of Condition Number Distributions, *SIAM J. on Matrix Analysis and Applications*, **27**, **2**, 547–560, 2005.

[GL13] G. H. Golub, C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 2013 (fourth edition).

[HMT11] N. Halko, P. G. Martinsson, J. A. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, *SIAM Review*, **53, 2**, 217–288, 2011.

[KS16] N. Kishore Kumar, J. Schneider, Literature Survey on Low Rank Approximation of Matrices, arXiv:1606.06511v1 [math.NA] 21 June 2016.

[M11] M. W. Mahoney, Randomized Algorithms for Matrices and Data, *Foundations and Trends in Machine Learning*, NOW Publishers, **3, 2**, 2011. arXiv:1104.5557 (2011)

[MW17] Cameron Musco, D. P. Woodruff: Sublinear Time Low-Rank Approximation of Positive Semidefinite Matrices, *IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS),* 672–683, 2017.

[O18] A. Osinsky, Rectangular Maximum Volume and Projective Volume Search Algorithms, September, 2018, arXiv:1809.02334

[OZ18] A.I. Osinsky, N. L. Zamarashkin, Pseudo-skeleton Approximations with Better Accuracy Estimates, *Linear Algebra and Its Applications*, **537**, 221–249, 2018.

[P00] C.-T. Pan, On the Existence and Computation of Rank-Revealing LU Factorizations, *Linear Algebra and Its Applications*, **316**, 199–222, 2000.

[P01] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser/Springer, Boston/New York, 2001.

[Pa] V. Y. Pan, Low Rank Approximation of a Matrix at Sub-linear Cost, preprint, 2019.

[PLa] V. Y. Pan, Q. Luan, Refinement of Low Rank Approximation of a Matrix at Sublinear Cost, arXiv:1906.04223 (Submitted on 10 Jun 2019).

[PLb] V. Y. Pan, Q. Luan, Randomized Approximation of Linear Least Squares Regression at Sub-linear Cost, arXiv:1906.03784 (Submitted on 10 Jun 2019).

[PLSZ16] V. Y. Pan, Q. Luan, J. Svadlenka, L.Zhao, Primitive and Cynical Low Rank Approximation, Preprocessing and Extensions, arXiv 1611.01391 (3 November, 2016).

[PLSZ17] V. Y. Pan, Q. Luan, J. Svadlenka, L. Zhao, Superfast Accurate Low Rank Approximation, preprint, arXiv:1710.07946 (22 October, 2017).

[PLSZa] V. Y. Pan, Q. Luan, J. Svadlenka, L. Zhao, CUR Low Rank Approximation at Sublinear Cost, arXiv:1906.04112 (Submitted on 10 Jun 2019).

[PQY15] V. Y. Pan, G. Qian, X. Yan, Random Multipliers Numerically Stabilize Gaussian and Block Gaussian Elimination: Proofs and an Extension to Low-rank Approximation, *Linear Algebra and Its Applications*, **481**, 202–234, 2015.

[PZ17a] V. Y. Pan, L. Zhao, New Studies of Randomized Augmentation and Additive Preprocessing, *Linear Algebra and Its Applications*, **527**, 256–305, 2017.

[PZ17b] V. Y. Pan, L. Zhao, Numerically Safe Gaussian Elimination with No Pivoting, *Linear Algebra and Its Applications*, **527**, 349–383, 2017.

[SST06] A. Sankar, D. Spielman, S.-H. Teng, Smoothed Analysis of the Condition Numbers and Growth Factors of Matrices, *SIAM J. Matrix Anal. Appl.*, **28**, **2**, 446–476, 2006.

[T11] J. A. Tropp, Improved Analysis of the Subsampled Randomized Hadamard Transform, *Adv. Adapt. Data Anal.*, **3, 1–2** (Special issue "Sparse Representation of Data and Images"), 115–126, 2011. Also arXiv math.NA 1011.1595.

[TYUC17] J. A. Tropp, A. Yurtsever, M. Udell, V. Cevher, Practical Sketching Algorithms for Low-rank Matrix Approximation, *SIAM J. Matrix Anal. Appl.*, **38**, 1454–1485, 2017.

[WLRT08] F. Woolfe, E. Liberty, V. Rokhlin, M. Tygert, A Fast Randomized Algorithm for the Approximation of Matrices, *Appl. Comput. Harmon. Anal.*, **25**, 335–366, 2008.