Some Recent and New Techniques for Superfast (Sublinear Cost)
Low Rank Approximation

Victor Y. Pan

Department of Computer Science
Lehman College of the City University of New York
Bronx, NY 10468 USA and
Ph.D. Programs in Mathematics and Computer Science
The Graduate Center of the City University of New York
New York, NY 10036 USA
victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan/

Abstract

We review and extend our recent superfast (sublinear cost) algorithms for accurate computation of low rank approximation of a matrix.

Key Words: Low-rank approximation, Sublinear cost, Superfast algorithms, Subspace sampling

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

1. Introduction: Superfast accurate LRA – the State of the Art and our study.

Low-rank approximation (LRA) of a matrix is one of the most fundamental problems of Numerical Linear Algebra and Data Mining and Analysis, with applications ranging from machine learning theory and neural networks to term document data and DNA SNP data (see surveys [HMT11], [MI1], and [KS16]).

Matrices representing Big Data are so immense in size that realistically one can only access and process a tiny fraction of their entries. Quite typically, however, they admit their LRA, that is, are close to low rank matrices or equivalently have low numerical rank. This can be decisive because one can operate with low rank matrices superfast – by using sublinear arithmetic time and memory space, that is, by using much fewer flops and memory cells than an input matrix has entries.

Can, however, one compute accurate LRA superfast? Adversary argument shows that every superfast algorithm fails already on a small family of matrices filled with 0s except for a single entry filled with 1 (see Appendix B), but as we showed in [PLSZ16] and [PLSZ17] some specified superfast algorithms compute accurate LRA of all matrices admitting LRA except for a rather narrow subclass. We are going to revisit and to extend that study.

---

1 We use such concepts as “low”, “small”, “nearby”, etc. defined in context.
2 Likewise superfast correctness verification is possible under some additional input information or assumptions on the input, although not for the worst case input and not even for the small input family of Appendix B.
2. Randomized accurate LRA. In [PLSZ16] and [PLSZ17] we rely on an algorithm from [TYUC17], where [CW09] Theorems 4.7 and 4.8 are cited as the source. The algorithm reduces the computation of LRA of a fixed $m \times n$ matrix $M$ of numerical rank $\rho$ to the computation of LRA of a $k \times l$ auxiliary matrix $FMH$ for any pair of integers $k$ and $l$ such that $\rho \leq k \leq m$ and $\rho \leq l \leq n$ and for random multipliers $F$ of size $k \times m$ and $H$ of size $n \times l$.

Based on the known techniques of random subspace sampling (cf. [HMTT1]) Tropp et al. proved in [TYUC17] that with a high probability (hereafter whp) the expected value of the Frobenius error norm of the computed LRA of $M$ stays within a factor of $\frac{k}{(k-l)(l-r)}$ from its minimum provided that $F$ and $H$ are standard Gaussian (aka normal) random matrices; this factor is close to 1 for $k \gg l \gg \rho$. Similar results, with a little smaller upper bound on the error probability and under lower bounds of order $r \log(r)$ on $k$ and $l$, have been proved in the case where the multipliers $F$ and $H$ are SHRT or SRFT matrices (that is, the matrices of subsampled Hadamard or Fourier randomized transforms). For $k \ll m$ and/or $l \ll n$ the matrix $FMH$ has small size such that the known algorithms compute its LRA superfast. The computation of this matrix, however, is not superfast in the cases of Gaussian, SRHT and SRFT matrices.

3. Dual superfast accurate LRA. In [PLSZ16] and [PLSZ17] we considered the latter approach primal and studied its dual variation where the multipliers $F$ and $H$ are fixed and an input matrix $M$ admitting LRA is random. Then we proved that the computed LRA is accurate whp for any pair of orthogonal or well-conditioned multipliers $F$ and $H$. In [PLSZ16] and [PLSZ17] we described some classes of sparse multipliers with which the LRA algorithms are superfast and proposed some policies of their successive application which increased the probability of computing accurate LRA of random inputs. We observe good accordance of this formal study with the results of our extensive tests on real world matrices, including some test matrices from [HMTT1].

4. Generation of sparse multipliers. Next we recall and extend a particular family of sparse multipliers in [PLSZ16] and [PLSZ17] defined by means of abridging the classical recursive processes of the generation of $n \times n$ SRHT and SRFT matrices in $k = \log_2(n)$ recursive steps for $n = 2^k$. The processes begin with the sparse matrix $\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$ for the $\frac{n}{2} \times \frac{n}{2}$ identity matrix $I = I_{n/2}$ and recursively fill it with nonzero entries. In $k$ steps it becomes dense, and after random unitary diagonal scaling its submatrix made up of a fixed smaller number of random columns or rows turns into SRHT or SRFT multiplier. With such a multiplier accurate LRA of any matrix $M$ admitting LRA is output whp, but in all our extensive tests the output remained as accurate when we used instead abridged SRHT or SRFT multipliers computed in at most three recursive steps.

Similarly a square Gaussian matrix has been decomposed in [PLSZ16] into the product of random bidiagonal and random permutation matrices, and then a rectangular multiplier has been obtained from it by means of row or column sampling.

Alternatively, in the spirit of [HMTT1] Remark 4.6), we can compute QR orthogonalization to any $n \times n$ matrix $G$, thus decomposing it into the product $G = Q_1Q_2 \cdots Q_s R$ where $Q_1, Q_2, \ldots, Q_s$ are sparse orthogonal matrices of Givens rotations or Householder transforms and $R$ is a right (upper) triangular matrix. Then the input matrix $M$ can be multiplied by a matrix $F$ or $H$ obtained by abridging a partial product $P_i = Q_1Q_2 \cdots Q_i$ for $i \leq s$. Here the multipliers $F$ and $H$ are made of a fixed number of the first rows or columns of $P_i$, respectively. One can combine such preprocessing with using diagonal scaling and random permutations.

---

3 The algorithms and estimates of [CW09] Theorems 4.7 and 4.8 use Rademacher (rather than Gaussian) matrices.

4 Hereafter we call such matrices Gaussian and write ”$\gg$” for “much greater than” and ”$\ll$” for “much smaller than”.

5 We assume that in a customary representation of a matrix $M$ admitting its LRA one or both of the matrices $U$ and $V$ are Gaussian and the other one or two factors are well-conditioned matrices of full rank; in that case with no loss of generality we can assume that the factor $T$ is a diagonal matrix.
5. Three-factor LRA. Before proceeding any further we formalize the computation of LRA. Recall that an \( m \times n \) matrix \( \tilde{M} \) has rank at most \( \rho \), \( \text{rank}(\tilde{M}) \leq \rho \), if
\[
\tilde{M} = XY, \ X \in \mathbb{C}^{m \times \rho}, \text{ and } Y \in \mathbb{C}^{\rho \times n}.
\] (0.1)

Likewise an \( m \times n \) matrix \( M \) has numerical rank at most \( \rho \), \( \text{nrk}(M) \leq \rho \), if it admits its close approximation by a matrix \( \tilde{M} \) of rank at most \( \rho \) or equivalently if there exist three matrices \( X, Y \) and \( E \) such that
\[
M = XY + E \text{ where } E \approx O, \ X \in \mathbb{C}^{m \times \rho}, \text{ and } Y \in \mathbb{C}^{\rho \times n}.
\] (0.2)

We naturally generalize such a 2-factor LRA \( XY \) of \( M \) to 3-factor LRA
\[
\tilde{M} = UTV + E, \text{ where } E \approx O, \ U \in \mathbb{C}^{m \times k}, \ T \in \mathbb{C}^{k \times l}, \ V \in \mathbb{C}^{l \times n},
\] (0.3)
\( \rho \leq k \leq m, \rho \leq l \leq n \), and matrices \( U, T, \) and \( V \) may have full rank exceeding \( \rho \).

As a special case of a 3-factor LRA \( UTV \) of (0.3) for \( k = l \) and \( T = I_k \) we obtain a 2-factor LRA, which satisfies (0.2) for \( \rho = \rho \). Moreover the pairs of maps \( UT \to X \) and \( V \to Y \) as well as \( U \to X \) and \( TV \to Y \) turn a 3-factor LRA \( UTV \) of (0.3) into a 2-factor LRA \( XY \).

6. SVD and its truncation. \( M = U_M \Sigma_M V_M^\dagger \) is Compact Singular Value Decomposition (SVD) of a matrix \( M \) of rank \( \rho \) for two orthogonal matrices of its left and right singular vectors \( U_M \in \mathbb{C}^{m \times \rho} \) and \( V_M \in \mathbb{C}^{n \times \rho} \), respectively, and the diagonal matrix of its singular values \( \Sigma_M = \text{diag}(\sigma_j(M))_{j=1}^\rho \) such that \( \sigma_1(M) \geq \sigma_2(M) \geq \cdots \geq \sigma_\rho(M) > 0 \). The matrix \( M^* = V_M \Sigma_M^{-1} U_M^* \) is the Moore–Penrose pseudo inverse of \( M \), \( M^* = M^{-1} \) for a nonsingular matrix \( M \).

An important 3-factor LRA of \( M \) is given by the \( \rho \)-truncation of its SVD, obtained by means of keeping \( \rho \) top (that is, largest) singular values of \( M \) and setting to 0 all the other ones, thus transforming SVD of \( M \) into its top SVD and minimizing both spectral and Frobenius norms \( ||E|| \), and \( ||E||_F \) of the error matrix \( E \) of rank-\( \rho \) approximation of \( M \).

**Lemma 0.1.** (The minimal error norm for an LRA: [GL13, Theorem 2.4.8].) For a matrix \( M \) and a positive integer \( \rho \), the \( \rho \)-truncation of SVD of \( M_\rho = UTV \) with \( T = \text{diag}(\sigma_j(M))_{j=1}^\rho \) denoting a diagonal matrix filled with the \( \rho \) top singular values of \( M \) in nonincreasing order, is a closest rank-\( \rho \) approximation of \( M \) under both spectral and Frobenius norms,
\[
||M_\rho - M|| = \sigma_{\rho+1}(M) \text{ and } \tau_{\rho+1}(M) := ||M_\rho - M||_F^2 = \sum_{j=\rho}^n \sigma_j^2(M),
\]
or in a unified way
\[
\rho + 1(M) := |M_\rho - M| = \min_{N: \text{rank}(N) = \rho} |M - N|.
\]

**Lemma 0.2.** (The impact of a perturbation of a matrix on its singular values: [GL13, Corollary 8.6.2].) For \( m \geq n \) and a pair of \( m \times n \) matrices \( M \) and \( M + E \) it holds that
\[
|\sigma_j(M + E) - \sigma_j(M)| \leq ||E|| \text{ for } j = 1, \ldots, n.
\]

The estimates of the following theorem imply that the top SVD of a matrix \( M \) is stable in its perturbation within \( 0.2(\sigma_\rho(M) - \sigma_{\rho+1}(M)) \); the estimates are more explicit than those by Davis-Kahan 1972 and Wedin 1973, which involve angles between singular spaces.
Theorem 0.1. (The impact of a perturbation of a matrix on its top singular vectors. [GL13, Theorem 8.6.5].) Suppose that
\[ g =: \sigma_\rho(M) - \sigma_{\rho+1}(M) > 0 \] and \( \|E\|_F \leq 0.2g \).

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 \( B_{\rho,\text{left}}(M) \), \( B_{\rho,\text{right}}(M) \), \( B_{\rho,\text{left}}(M + E) \), and \( B_{\rho,\text{right}}(M + E) \) such that
\[
\max\{\|B_{\rho,\text{left}}(M + E) - B_{\rho,\text{left}}(M)\|_F, \|B_{\rho,\text{right}}(M + E) - B_{\rho,\text{right}}(M)\|_F\} \leq 4\|E\|_F/g.
\]

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

7. CUR LRA. CUR LRA is another important 3-factor LRA, highly popular and particularly memory efficient (see [GTZ97], [GE96], [P00], [DMM08], [BW17], [OZ18], and the references therein). In this case we seek LRA \( M' \approx M \) in the form of CUR where \( R \) and \( C \) are two submatrices of \( M \) made up of its fixed or random sets of \( k \) rows and \( l \) columns, respectively, and \( U \) is an \( l \times k \) matrix.

In a popular option (cf. [DMM08]), which we call canonical CUR LRA, we define a nucleus \( U \) as follows: first define a CUR generator \( G \) made up of the \( kl \) common entries of the factors \( C \) and \( R \) of CUR, then compute the \( \rho \)-truncation \( G_\rho \) of its SVD, and finally let \( U \) be the Moore–Penrose pseudo inverse \( G_\rho^+ \). In this case we only need to fix two index sets that define the submatrices \( G, C \) and \( R \), and then to compute the nucleus \( U \) by using about \( kl \log(\min\{k,l\}) \) flops deterministically or \( O(kl \log(\min\{k,l\})) \) flops whp. In particular \( U = G^{-1} \) if \( k = l = \rho \) and if \( G \) is a nonsingular submatrix of \( M \).

Theorem 0.2. For a canonical CUR LRA \( M' \) it holds that (i) \( M = CG^+R \) if and only if \( \text{rank}(G) = \text{rank}(M) = \rho > 0 \), and (ii) \( M' \approx M \) if and only if \( \text{rank}(G) = \text{rank}(M) \).

Proof. Readily verify claim (i). See a proof of claim (ii) in [PLSZ16], [PLSZ17]. □

Suppose that we are given a rank-\( \rho \) truncation \( M' \) of SVD of a matrix \( M \) defining its LRA. Then by virtue of claim (i) we can rewrite this LRA as a canonical CUR decomposition \( M' = CUR \) for \( U = G^{-1} \) as soon as we find a nonsingular \( \rho \times \rho \) submatrix \( G \) of the LRA \( M' \). In [PLSZ17] Appendix D.2 such a submatrix, computed superfast, has the norm \( ||G^{-1}|| \) within a small factor from its minimum value \( 1/\sigma_\rho(M) \).

8. The size of LRA versus its accuracy. The algorithms of [TYUC17], [PLSZ16] and [PLSZ17] output LRA \( XY \approx M \) for \( M \in \mathbb{C}^{m \times n}, X \in \mathbb{C}^{m \times l} \) and \( Y \in \mathbb{C}^{n \times \rho} \), where high output accuracy is ensured for an appropriate choice of integers \( k \) and/or \( l \) greatly exceeding the numerical rank \( \rho = \text{rank}(M) \). Such an increase of the size of LRA is undesirable, but the size can be decreased to its minimum at the price of a mild increase of the output error norm bound. Namely, according to [TYUC17, Proposition 6.1],
\[
||(XY)_{\rho} - M||_F \leq \tau_{\rho+1}(M) + 2||XY - M||_F \tag{0.4}
\]
for any LRA of (0.2) and \( \tau_{\rho+1}(M) \) of Lemma (1.1).

Computation of SVD of \( M \) involves at least order of \( mn \min\{m,n\} \) flops; one can extend the randomized LRA algorithms in [HMT11] with SRHT and SRFT multipliers to the computation of the \( \rho \)-truncation \( M_{\rho} \) whp by using \( O(mn \log(\rho)) \) flops. Furthermore given any 3-factor LRA of (0.3)
for $\rho \leq \min\{k, l\}$ and $k \ll m$ or $l \ll n$, Algorithm [A, in Appendix A] computes the $\rho$-truncation of its SVD superfast, thus obtaining rank-$\rho$ approximation with only minor sacrifice in error bound.

If a computed LRA $\tilde{M}$ is too crude, one can try to refine it by computing LRA anew with some alternative sparse multipliers, but here are some recipes for the refinement that use a computed crude LRA $\tilde{M} = UTV$. In the first two recipes we assume that this crude LRA is reasonably accurate so that the top SVD of LRA is close to that of an input matrix (see Lemma 0.2 and Theorem 0.1).

(i) Suppose that we have computed SVD $\tilde{M} = U_{\tilde{M}}\Sigma_{\tilde{M}}V_{\tilde{M}}^*$ and have applied the algorithm of [TYUC17] with no randomization, by choosing the multipliers $F = U_{\tilde{M}}^*$ and $H = V_{\tilde{M}}^*$. Then by extending the analysis of [HMT11, Section 10] we can prove that both spectral and Frobenius norms of the output error matrix are bounded by some values close to $2\tilde{\sigma}_{\rho+1}(M)$.

These orthogonal multipliers $F$ and $H$ are not sparse, and so the algorithm is not superfast, but we can replace them by sparse orthogonal ones according to a recipe of section 4.

(ii) The random subspace sampling algorithms of Drineas et al. [DMM08] and of various subsequent papers (see [KS16], [BW17], and the references therein) compute accurate LRA of a matrix $M$ superfast whp provided that its top SVD is available. The top SVD of LRA $M$ can serve instead of the top SVD of an input matrix $M$ as long as these top SVDs are close enough to one another.

(iii) Even if the top SVD of an LRA $\tilde{M}$ is not close to that of $M$, we can apply iterative refinement of an LRA as follows:

(a) compute the rank-$\rho$ truncation $\tilde{M}_\rho$ of SVD of a given LRA $\tilde{M}$ of an input matrix $M$;

(b) compute a rank-$2\rho$ approximation $\Delta$ of the matrix $M - \tilde{M}_\rho$ (notice that $\text{nrank}(M - \tilde{M}_\rho) \leq \text{nrank}(M) + \text{rank}(\tilde{M}_\rho) \leq 2\rho$);

(c) compute the rank-$\rho$ truncation $(\Delta + \tilde{M})_\rho$ of the SVD of the matrix $\Delta + \tilde{M}$.

Stages (a) and (c) can be performed superfast because their inputs have small size. At stage (b) we can apply any superfast algorithm for LRA and should extend the classical techniques of iterative refinement (cf. [H02], [S98]). In particular if we have computed $\tilde{M}$ by applying superfast dual variant of the algorithm of [TYUC17], then we can reuse the multipliers $F$ and $H$ involved in it, but we should compute the product $F(M - \tilde{M}_\rho)H$ with higher precision in order to decrease the output error of stage (a). We can do this because $\tilde{M}_\rho \approx \tilde{M} \approx M$ and so $||M - \tilde{M}_\rho|| \ll ||M||$.

One can recursively repeat stages (a)–(c) hoping that $(\Delta + \tilde{M})_\rho \rightarrow M_\rho$. Recall that if $(\Delta + \tilde{M})_\rho$ becomes a random matrix of rank $\rho$, then LRA algorithms applied to it should output an accurate LRA of a matrix $M$ whp (cf. [PLSZ16] and [PLSZ17]).

9. Homotopy continuation technique. In the proposed recipes (i) and (ii) for iterative refinement we assume that LRA $\tilde{M}$ is close to a matrix $M$, and clearly recipe (iii) is most effective where $\tilde{M}$ is close to $M$. Thus one can apply these recipes more efficiently by using homotopy continuation technique, that is, by applying the recipes recursively to the pairs of matrices $M_i$ and $\tilde{M}_i$ for $i = 0, 1, \ldots$ where $M_0 = \tilde{M}$, $M_{i+1} = M_i + s_i(M - M_i)$, $s_i$ are sufficiently small positive scalars, $i = 0, 1, \ldots$, and $\tilde{M}_i$ for a positive $i$ is the LRA of the matrix $M_{i-1}$ computed according to the selected recipe (i), (ii) or (iii). In order to choose a scalar $s_i$ one may first estimate or guess the error norm $||M - M_i||$ and then choose the maximal $s_i$ for which Lemma 0.2 and Theorem 0.1 enable us to control the errors of the current refinement iteration.

**Appendix**

In this case the algorithm can superfast compute so called leverage scores, which then serve as sampling probabilities defining superfast random subspace sampling.

5
A Transition from an LRA to top SVD

Algorithm A.1. (Transition from an LRA to the Top SVD. [HMTT11] Algorithms 5.1 and 5.2.)

**INPUT:** Four matrices $A \in \mathbb{C}^{m \times l}$, $W \in \mathbb{C}^{l \times k}$, $B \in \mathbb{C}^{k \times n}$, and $M \in \mathbb{C}^{m \times n}$ such that

$$M = AWB + E, \quad ||E|| = O(\tilde{\sigma}_{r+1}), \quad r \leq \min\{k, l\}, \quad k \ll m, \quad \text{and} \quad l \ll n \text{ for } \tilde{\sigma}_{r+1} \text{ of Lemma 0.1}$$

**OUTPUT:** Three matrices $U \in \mathbb{C}^{m \times r}$ (orthogonal), $\Sigma \in \mathbb{C}^{r \times r}$ (diagonal), and $V^* \in \mathbb{C}^{r \times n}$ (orthogonal) such that $M = U\Sigma V^* + E'$ for $||E'|| = O(\tilde{\sigma}_{r+1})$.

**COMPUTATIONS:**

1. Compute QRP rank-revealing factorization of the matrices $A$ and $B$:

   $$A = (Q \mid E_{m,l-r})RP \quad \text{and} \quad B = P'R'(\tilde{Q}'E_{k-r,n})$$

   where $Q \in \mathbb{C}^{m \times r}$, $Q' \in \mathbb{C}^{r \times n}$ and $||E_{m,l-r}|| + ||E_{k-r,n}|| = O(\tilde{\sigma}_{r+1})$ (cf. [GE96]). Substitute the expressions for $A$ and $B$ into the matrix equation $M = AWB + E$ and obtain $M = QXQ' + E'$ where $X = RPW'P'R' \in \mathbb{C}^{r \times r}$ and $||E'|| = O(\tilde{\sigma}_{r+1})$.

2. Compute SVD $X = \bar{U}\Sigma\bar{V}^*$. Output the $r \times r$ diagonal matrix $\Sigma$.

3. Compute and output the orthogonal matrices $U = Q\bar{U}$ and $V^* = \bar{V}^*Q'$.

This algorithm uses $ml + lk + kn$ memory cells and $O(ml^2 + nk^2)$ flops.

B Two Small Families of Hard Inputs for Sublinear Cost LRA

**Example B.1.** Define $\delta$-matrices of rank 1 filled with zeros except for a single entry filled with 1. There are exactly $mn$ such $m \times n$ matrices, e.g., four matrices of size $2 \times 2$:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. $$

The output matrix of any sublinear cost algorithm approximates nearly 50% of all these matrices as poorly as the matrix filled with the values $1/2$ does. Indeed any superfast algorithm only depends on a small subset of all $mn$ input entries, and so its output is invariant in the input values at all the other entries. In contrast nearly $mn$ pairs of $\delta$-matrices vary on these entries by 1. The approximation by a single value is off by at least 1/2 for one or both of the matrices of such a pair, that is, it is off at least as much as the approximation by the matrix filled with the values 1/2. Likewise if a superfast LRA algorithm is randomized and accesses an input entry with a probability $p$, then with probability $1-p$ it approximates some $\delta$-matrix with an error at least 1/2 at that entry.

Furthermore if a superfast algorithm has been applied to a perturbed $\delta$-matrix $W$ and only accesses its nearly vanishing entries, then it would optimize LRA over the class of nearly vanishing matrices and would never detect its failure to approximate the only entry of the matrix $W$ close to 1.

**Example B.2.** $\delta$-matrices are sparse, but subtract the rank-1 matrix filled with 1/2 from every $\delta$-matrix and obtain a family of dense matrices of rank 2 that are not close to sparse matrices, but similarly to Example B.1 LRA of this family computed by any superfast algorithm is at best within 1/2, that is, no better than by the trivial matrix filled with zeros.

**Acknowledgements:** Our work has been supported by NSF Grants CCF–1116736, CCF–1563942 and CCF–1733834 and PSC CUNY Award 69813 00 48.
References

[BW17] C. Boutsidis, D. Woodruff, Optimal CUR Matrix Decompositions, *SIAM Journal on Computing*, **46**, 2, 543–589, 2017, DOI:10.1137/140977898.

[CW09] K. L. Clarkson, D. P. Woodruff, Numerical linear algebra in the streaming model, in *Proc. 41st ACM Symposium on Theory of Computing (STOC 2009)*, pages 205–214, ACM Press, New York, 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.

[GE96] M. Gu, S.C. Eisenstat, An Efficient Algorithm for Computing a Strong Rank Revealing QR Factorization, *SIAM J. Sci. Comput.*, **17**, 848–869, 1996.

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

[GTZ97] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A Theory of Pseudo-skeleton Approximations, *Linear Algebra and Its Applications*, **261**, 1–21, 1997.

[H02] N. J. Higham, *Accuracy and Stability in Numerical Analysis*, SIAM, Philadelphia, 2002. (second 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. Preprint: [arXiv:1104.5557](https://arxiv.org/abs/1104.5557) (2011) (Abridged version in: *Advances in Machine Learning and Data Mining for Astronomy*, edited by M. J. Way et al., pp. 647–672, 2012.)

[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.

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

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

[S98] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decompositions*, SIAM, Philadelphia, 1998.
[TYUC17] J. A. Tropp, A. Yurtsever, M. Udell, V. Cevher, Practical Sketching Algorithms for Low-rank Matrix Approximation, *SIAM J. Matrix Anal. Appl.*, **38**, 4, 1454–1485, 2017. Also see [arXiv:1609.00048](http://arxiv.org/abs/1609.00048) January 2018.