

2014

TR-2014008: Supporting GENP and Low-Rank Approximation with Random Multipliers

Victor Y. Pan

Guoliang Qian

Xiaodong Yan

Follow this and additional works at: http://academicworks.cuny.edu/gc_cs_tr

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Pan, Victor Y.; Qian, Guoliang; and Yan, Xiaodong, "TR-2014008: Supporting GENP and Low-Rank Approximation with Random Multipliers" (2014). *CUNY Academic Works*.
http://academicworks.cuny.edu/gc_cs_tr/399

This Technical Report is brought to you by CUNY Academic Works. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of CUNY Academic Works. For more information, please contact AcademicWorks@gc.cuny.edu.

Supporting GENP and Low-rank Approximation with Random Multipliers *

Victor Y. Pan^{[1,2],[a]}, Guoliang Qian^{[2],[b]} and Xiaodong Yan^{[2],[c]}

^[1] Department of Mathematics and Computer Science
Lehman College of the City University of New York
Bronx, NY 10468 USA

^[2] Ph.D. Programs in Mathematics and Computer Science
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] gqian@gc.cuny.edu

^[c] xyannyc@yahoo.com

May 1, 2014

Abstract

We prove that standard Gaussian random multipliers are expected to stabilize numerically both Gaussian elimination with no pivoting and block Gaussian elimination and that they also are expected to support the celebrated randomized algorithm for low-rank approximation of a matrix even without customary oversampling. Our tests show similar results where we apply random circulant and Toeplitz multipliers instead of standard Gaussian ones.

2000 Math. Subject Classification: 15A52, 15A12, 15A06, 65F22, 65F05

Key Words: Random matrices, Random multipliers, GENP, Low-rank approximation, SRFT matrices, Random circulant matrices

1 Introduction

1.1 Overview

A standard Gaussian random matrix (hereafter we call it just *Gaussian matrix*) is likely to be well conditioned [D88], [E88], [ES05], [CD05], [SST06], and motivated by this well known property we apply multiplication by such matrices in order to

- stabilize numerically Gaussian elimination and block Gaussian elimination by using no pivoting, orthogonalization or symmetrization

*Some results of this paper have been presented at the ACM-SIGSAM International Symposium on Symbolic and Algebraic Computation (ISSAC '2011), in San Jose, CA, June 8–11, 2011, the 3rd International Conference on Matrix Methods in Mathematics and Applications (MMMA 2011), in Moscow, Russia, June 22–25, 2011, the 7th International Congress on Industrial and Applied Mathematics (ICIAM 2011), in Vancouver, British Columbia, Canada, July 18–22, 2011, the SIAM International Conference on Linear Algebra, in Valencia, Spain, June 18–22, 2012, the Conference on Structured Linear and Multilinear Algebra Problems (SLA2012), in Leuven, Belgium, September 10–14, 2012, and the 18th Conference of the International Linear Algebra Society (ILAS 2013), in Providence, RI, USA, June 3–7, 2013

- approximate leading and trailing singular spaces of an ill conditioned matrix, associated with its largest and smallest singular values, respectively, and
- approximate this matrix by a low-rank matrix with no customary oversampling

We support these applications both formally and experimentally. To yield formal support we first apply deterministic Matrix Analysis (see Sections 3 and 4) and then invoke a basic lemma about the products of Gaussian and square unitary matrices and apply the known probabilistic estimates for the norms of Gaussian matrices and their generalized inverses (see Section 5). Our new insights and techniques should motivate further work on these subjects.

1.2 Numerical Gaussian elimination with no pivoting

It is well known that even for a nonsingular and well conditioned input matrix, Gaussian elimination fails in numerical computations with rounding errors as soon as it encounters a vanishing or nearly vanishing leading (that is northwestern) entry. It is not always easy to see a priori which inputs are bad in this respect. For example, the unitary matrix of discrete Fourier transform of a large size turns out to be bad (see Corollary 5.12). In practice the users avoid such encounters by applying GEPP, which stands for Gaussian elimination with partial pivoting and has some limited formal and ample empirical support. Partial pivoting, that is appropriate row interchange, takes its toll, however: "pivoting usually degrades the performance" [GL96, page 119]. It interrupts the stream of arithmetic operations with foreign operations of comparison, involves book-keeping, compromises data locality, increases communication overhead and data dependence, and tends to destroy matrix structure. Avoiding pivoting in numerical solution of Toeplitz and Toeplitz-like linear systems of equations was the central subject of the highly recognized papers [GKO95] and [G98], which first reduced the task to the case of Cauchy-like inputs by specializing the techniques of the transformation of matrix structures from [P90] (cf. [Pa]) and then applied a fast Cauchy-like variant of GEPP.

The users apply *Gaussian elimination with no pivoting* (hereafter we refer to it as *GENP*) or its variants wherever this application is numerically safe, in particular where a well conditioned input matrix is positive definite, diagonally dominant, or totally positive. We prove that this class can be generalized to all nonsingular and well conditioned matrices if we allow to *precondition* GENP with Gaussian multipliers (see Section 5.2). In our tests of this recipe the output accuracy of GENP with preprocessing was a little lower than in the case of the customary GEPP, but we readily fixed the discrepancy by applying one or a few steps of iterative refinement, performed at a dominated computational cost (see Figures 2 and 3 and Table D.3). All our results on GENP can be extended to *block Gaussian elimination* (see Section 2.6).

1.3 Low-rank approximation of a matrix

Random multipliers are known to be highly efficient for low-rank approximations of an $m \times n$ matrix A having a small numerical rank r . As the basic step one computes the product AH where H is a random $n \times (r + p)$ multipliers and p is a positive *oversampling integer*. The resulting randomized algorithms have been studied extensively, both formally and experimentally [HMT11], [M11]. They are numerically stable, run at a low computational cost, allow low-cost improvement of the output accuracy by means of the Power Method, and have important applications to matrix computations, data mining, statistics, PDEs and integral equations. By extending our analysis of preprocessed GENP, we prove and confirm empirically that even for an $n \times r$ Gaussian multiplier, that is *without customary oversampling*, the algorithms are expected to output some quite close low-rank approximations of the input matrix (see Section 5.3, Figures 6 and 7, and Table D.10).

1.4 Computations with random structured multipliers

The SRFT $n \times q$ multipliers H involve only n random parameters versus nq parameters of Gaussian multipliers, accelerate the computation of the product AH by a factor of $q/\log(q)$ versus $n \times q$ Gaussian multipliers H , and are expected to support rank- r approximation provided that an

oversampling integer $p = q - r$ has order $r \log(r)$ [HMT11, Section 11]. We readily extend the result to prove it also in the case where $n \times q$ products of random $n \times n$ circulant and random $n \times q$ permutation matrices are used as multipliers instead of SRFT matrices (see Remark 5.1). It is known that empirically SRFT and a number of other structured multipliers support low-rank approximation of the same quality as Gaussian multipliers and work already where the oversampling parameter p is a reasonable constant [HMT11, page 279], [M11]. We observe such empirical behavior also in the cases where we preprocess GENP or compute low-rank approximation with no oversampling by applying random circulant or Toeplitz multipliers, respectively (see Figures 2, 3, 6 and 7 and Tables D.4, D.5, D.11, and D.12). Success with circulant multipliers would support numerical application of the MBA celebrated superfast algorithm to Toeplitz linear systems [P01, Chapter 5], but no formal support of this technique is available. Moreover we prove that at least in the special case where the input is the matrix of the discrete Fourier transform of a large size the application of GENP with a Gaussian circulant multiplier is expected to fail (see Example 5.4). At the end of Sections 5.4 and 7 we comment on some research challenges motivated by this proof.

1.5 Related works

Preconditioning of linear systems of equations is a classical subject [A94], [B02], [G97]. On early work on randomized multiplicative preprocessing as a means of countering degeneracy of matrices see Section 2.13 “Regularization of a Matrix via Preconditioning with Randomization” in [BP94] and the bibliography therein. On specialization of such techniques to Gaussian elimination see [PP95]. Randomized multiplicative preconditioning for numerical stabilization of GENP was proposed in [PGMQ, Section 12.2] and [PQZ13], although only weaker theorems on the formal support of this approach were stated and their proofs were omitted. The paper [BBD12] and the bibliography therein cover the heuristic application of PRBMs (that is Partial Random Butterfly Multipliers), providing empirical support for GENP with preprocessing. On low-rank approximation we refer the reader to the surveys [HMT11] and [M11], which were the springboard for our study in Section 4. We cite these and other related works throughout the paper and refer the reader to [PQZa, Section 11] on further bibliography. The estimates of our Corollary 3.1 are close to the ones of [PQ10, Theorem 3.8], which were the basis for our algorithms in [PQ10], [PQ12], and [PQZC]. Unlike the latter papers, however, we state these basic estimates in a simpler form, refine them by following [CD05] rather than [SST06], and include their detailed proofs. On the related subject of estimating the norms and condition numbers of Gaussian matrices and random structured matrices see [D88], [E88], [ES05], [CD05], [SST06], [HMT11], [T11], and [PQa]. For a natural extension of our present work, one can combine randomized matrix multiplication with randomized augmentation and additive preprocessing of [PQ10], [PQ12], and [PQZC].

1.6 Organization of the paper

In the next section we recall some definitions and basic results. In Section 3 we first estimate the singular values of matrix products and then relate these results to preprocessing GENP. In Section 4 we first recall an algorithm from [HMT11] for low-rank approximation of a matrix that has a small numerical rank, and then analyze some variants of this algorithm using no oversampling and no randomization. In Section 5 we incorporate Gaussian randomization in the algorithms of the two previous sections, then complete their analysis, show some modifications, and comment on using non-Gaussian, and in particular random unitary circulant and Toeplitz multipliers. In Section 6 we cover numerical tests (the contribution of the second and third authors). Section 7 contains a brief summary. In Appendix A we estimate the dimension of the variety of the matrices of smaller ranks. In Appendix B we recall the known probabilistic estimates for the error norms of randomized low-rank approximations. In Appendix C we estimate the probability that a random matrix has full rank under the uniform probability distribution. In Appendix D we display tables with our test results, which are more detailed than the data given by the plots in Section 6. Some readers may be only interested in the part of our paper on GENP. They can skip Sections 2.3 (except for Theorem 2.1), 2.4, 4, 5.3, and 6.2.

2 Some definitions and basic results

Except for using unitary circulant matrices in Sections 5.4 and 6.2, we assume computations in the field \mathbb{R} of real numbers, but the extension to the case of the complex field \mathbb{C} is quite straightforward. Hereafter “flop” stands for “arithmetic operation”, “i.i.d.” stands for “independent identically distributed”, and “Gaussian matrix” stands for “standard Gaussian random matrix” (cf. Definition 5.1). The concepts “large”, “small”, “near”, “closely approximate”, “ill conditioned” and “well conditioned” are quantified in the context. By saying “expect” and “likely” we mean “with probability 1 or close to 1”. (We do not use the concept of the expected value, except for some equations of Theorem 5.4, Corollary 5.3, and Appendix B.) Next we recall and extend some customary definitions of matrix computations [GL96], [S98].

2.1 Some basic definitions of matrix computations

$\mathbb{R}^{m \times n}$ is the class of real $m \times n$ matrices $A = (a_{i,j})_{i,j}^{m,n}$. $(B_1 \mid \dots \mid B_k)$ is a $1 \times k$ block matrix with the blocks B_1, \dots, B_k . $\text{diag}(B_1, \dots, B_k) = \text{diag}(B_j)_{j=1}^k$ is a $k \times k$ block diagonal matrix with the diagonal blocks B_1, \dots, B_k . In both cases the blocks B_j can be rectangular. \mathbf{e}_i is the i th coordinate vector of dimension n for $i = 1, \dots, n$. These vectors define the $n \times n$ identity matrix $I_n = (\mathbf{e}_1 \mid \dots \mid \mathbf{e}_n)$. $O_{k,l}$ is the $k \times l$ matrix filled with zeros. We write I and O where the matrix size is defined by context. $\mathcal{R}(A)$ denotes the range of an $m \times n$ matrix A , that is the linear space $\{\mathbf{z} : \mathbf{z} = \mathbf{A}\mathbf{x}\}$ generated by its columns. Any matrix having full column rank is a *matrix basis* for its range. $\text{rank}(A) = \dim \mathcal{R}(A)$ denotes its rank. $A_{k,l}$ denotes the leading, that is northwestern $k \times l$ block submatrix of a matrix A , and in Section 2.6 we also write $A^{(k)} = A_{k,k}$. A^T is the transpose of a matrix A . A_s^T denotes the transpose $(A_s)^T$ of a matrix A_s , e.g., $A_{k,l}^T$ stands for $(A_{k,l})^T$. A matrix of a rank ρ has *generic rank profile* if all its leading $i \times i$ blocks are nonsingular for $i = 1, \dots, \rho$. If such a matrix is nonsingular itself, then it is called *strongly nonsingular*.

Preprocessing $A \rightarrow FAH$ for a pair of nonsingular matrices F and H , one of which can be the identity matrix I , reduces the inversion of a matrix A to the inversion of the product FAH , and similarly for the solution of a linear system of equations.

Fact 2.1. *Assume three nonsingular matrices F , A , and H and a vector \mathbf{b} . Then $A^{-1} = H(AH)^{-1}$, $A^{-1} = (FA)^{-1}F$, $A^{-1} = H(FAH)^{-1}F$. Furthermore, if $\mathbf{A}\mathbf{x} = \mathbf{b}$, then $AH\mathbf{y} = \mathbf{b}$, $FA\mathbf{x} = F\mathbf{b}$, and $FAH\mathbf{y} = F\mathbf{b}$, $\mathbf{x} = H\mathbf{y}$.*

2.2 Matrix norms, orthogonality, SVD, and pseudo-inverse

$\|A\| = \|A\|_2 = \sup_{\mathbf{v}, \tau_{\mathbf{v}}=1} \|A\mathbf{v}\|$ is the spectral norm of a matrix $A = (a_{i,j})_{i,j=1}^{m,n}$, whereas $\|A\|_F = \sqrt{\sum_{i,j=1}^{m,n} |a_{i,j}|^2}$ is its Frobenius norm. Then $\|A\|_F \leq \sqrt{n}\|A\|$ and $\|AB\| \leq \|A\| \|B\|$ where the matrices A and B have the same number of columns. A real matrix Q is called *orthogonal* if $Q^T Q = I$ or $Q Q^T = I$. $(Q, R) = (Q(A), R(A))$ for an $m \times n$ matrix A of rank n denotes a unique pair of orthogonal $m \times n$ and upper triangular $n \times n$ matrices such that $A = QR$ and all diagonal entries of the matrix R are positive [GL96, Theorem 5.2.2]. We recall that

$$\|U\| = \|U\|_F = 1, \quad \|UAV\| \leq \|A\|, \quad \text{and} \quad \|UAV\|_F \leq \|A\|_F \quad \text{for orthogonal matrices } U \text{ and } V. \quad (2.1)$$

An *SVD* or *full SVD* of an $m \times n$ matrix A of a rank ρ is a factorization

$$A = S_A \Sigma_A T_A^T \quad (2.2)$$

where $S_A = (\mathbf{s}_i)_{i=1}^m$ and $T_A = (\mathbf{t}_j)_{j=1}^n$ are square orthogonal matrices, $\Sigma_A = \text{diag}(\widehat{\Sigma}_A, O_{m-\rho, n-\rho})$, $\widehat{\Sigma}_A = \text{diag}(\sigma_j(A))_{j=1}^\rho$, $\sigma_j = \sigma_j(A) = \sigma_j(A^T)$ is the j th largest singular value of a matrix A , and $\sigma_j = 0$ for $j > \rho$. These values have the minimax property

$$\sigma_j = \max_{\dim(\mathbb{S})=j} \min_{\mathbf{x} \in \mathbb{S}, \|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|$$

for $j = 1, \dots, \rho$ and linear spaces \mathbb{S} (see [GL96, Theorem 8.6.1]). It follows that $\sigma_\rho > 0$, $\sigma_1 = \max_{\|x\|=1} \|Ax\| = \|A\|$, $\|A\|_F^2 = \sum_{j=1}^\rho \sigma_j^2$, and

$$\min_{\text{rank}(B) \leq s-1} \|A - B\| = \sigma_s(A), \quad s = 1, 2, \dots \quad (2.3)$$

Fact 2.2. *If A_0 is a submatrix of a matrix A , then $\sigma_j(A) \geq \sigma_j(A_0)$ for all j .*

Proof. [GL96, Corollary 8.6.3] implies the claimed bound where A_0 is a block column of the matrix A . Transposition of a matrix and permutations of its rows and columns do not change singular values, and thus we can extend the bounds to all submatrices A_0 . \square

Fact 2.3. *(Cf. [GL96, Corollary 8.6.3].) Suppose $r + l \leq n \leq m$, $l \geq 0$, $1 \leq k \leq r$, $A \in \mathbb{R}^{m \times n}$, and $A_{m,r}$ is the leftmost $m \times r$ block of the matrix A . Then $\sigma_k(A_{m,r}) \geq \sigma_{k+l}(A_{m,r+l})$.*

$A^+ = T_A \text{diag}(\widehat{\Sigma}_A^{-1}, O_{n-\rho, m-\rho}) S_A^T$ is the Moore–Penrose pseudo-inverse of the matrix A of (2.2). If a matrix A has full column rank ρ , then $A^+ = (A^T A)^{-1} A^T$ and

$$\|A^+\| = 1/\sigma_\rho(A). \quad (2.4)$$

A^{+T} stands for $(A^+)^T = (A^T)^+$, A_s^T for $(A_s)^T$, and A_s^+ for $(A_s)^+$ where s can denote a scalar, a matrix, or a pair of such objects.

Corollary 2.1. *Assume that $\text{rank}(A_{m,r}) = r$ and $\text{rank}(A_{m,r+l}) = r + l$ for the matrices $A_{m,r}$ and $A_{m,r+l}$ of Fact 2.3. Then $\|A_{m,r}^+\| \leq \|A_{m,r+l}^+\|$.*

Proof. Combine Fact 2.3 for $k = r$ with equation (2.4). \square

2.3 Perturbation of matrix inverse, Q-factor and orthogonal projector

Theorem 2.1. [S98, Corollary 1.4.19]. *Assume a pair of square matrices A (nonsingular) and E such that $\|A^{-1}E\| \leq 1$. Then $\|(A+E)^{-1}\| \leq \frac{\|A^{-1}\|}{1-\|A^{-1}E\|}$ and moreover $\frac{\|(A+E)^{-1}-A^{-1}\|}{\|A^{-1}\|} \leq \frac{\|A^{-1}\|}{1-\|A^{-1}E\|}$.*

Theorem 2.2. [S95, Theorem 5.1]. *Assume a pair of $m \times n$ matrices A and $A + E$, and let the norm $\|E\|$ be small. Then $\|Q(A+E) - Q(A)\| \leq \sqrt{2}\|A^+\| \|E\|_F + O(\|E\|_F^2)$.*

P_A denotes the orthogonal projector on the range of a matrix A having full column rank,

$$P_A = A(A^T A)^{-1} A^T = AA^+ = QQ^T \text{ for } Q = Q(A). \quad (2.5)$$

Corollary 2.2. *Suppose $m \times n$ matrices A and $A + E$ have full rank. Then*

$$\|P_{A+E} - P_A\| \leq 2\|Q(A+E) - Q(A)\| \leq 2\sqrt{2} \|A^+\| \|E\|_F + O(\|E\|_F^2).$$

Proof. Clearly $P_{A+E} - P_A = Q(A+E)Q(A+E)^T - Q(A)Q(A)^T =$

$$(Q(A+E) - Q(A))Q(A+E)^T + Q(A)(Q(A+E)^T - Q(A)^T).$$

Consequently

$$\|P_{A+E} - P_A\| \leq \|Q(A+E) - Q(A)\| \|Q(A+E)^T\| + \|Q(A)\| \|Q(A+E)^T - Q(A)^T\|.$$

Substitute $\|Q(A)\| = \|Q(A+E)^T\| = 1$ and $\|Q(A+E)^T - Q(A)^T\| = \|Q(A+E) - Q(A)\|$ and obtain that $\|P_{A+E} - P_A\| \leq 2\|Q(A+E) - Q(A)\|$. Substitute the bound of Theorem 2.2. \square

2.4 Truncation of SVD. Leading singular spaces

Suppose $\rho = \text{rank}(A)$, write $S_{\rho,A} = (S_A)_{m,\rho}$, $T_{\rho,A} = (T_A)_{n,\rho}$, and $\Sigma_{\rho,A} = (\Sigma_A)_{\rho,\rho} = \text{diag}(\sigma_j)_{j=1}^\rho$, and obtain the *thin* or *compact SVD*

$$A = S_{\rho,A} \Sigma_{\rho,A} T_{\rho,A}^T. \quad (2.6)$$

Now for every integer r in the range $1 \leq r \leq \rho = \text{rank}(A)$, write $\Sigma_{\rho,A} = \text{diag}(\Sigma_{r,A}, \bar{\Sigma}_{A,r})$ and partition the matrices $S_{\rho,A}$ and $T_{\rho,A}$ into block columns, $S_{\rho,A} = (S_{r,A} \mid \bar{S}_{A,r})$, and $T_{\rho,A} = (T_{r,A} \mid \bar{T}_{A,r})$ where $\Sigma_{r,A} = (\Sigma_A)_{r,r} = \text{diag}(\sigma_j)_{j=1}^r$, $S_{r,A} = (S_A)_{m,r}$, and $T_{r,A} = (T_A)_{n,r}$. Then partition the thin SVD as follows,

$$A_r = S_{r,A} \Sigma_{r,A} T_{r,A}^T, \quad \bar{A}_r = \bar{S}_{A,r} \bar{\Sigma}_{A,r} \bar{T}_{A,r}^T, \quad A = A_r + \bar{A}_r \text{ for } 1 \leq r \leq \rho = \text{rank}(A), \quad (2.7)$$

and call the above decomposition of the matrix A_r the r -truncation of the thin SVD (2.6). Note that \bar{A}_ρ is an empty matrix. Equation (2.3) implies that

$$\|A - A_r\| = \sigma_{r+1}(A). \quad (2.8)$$

Write $\mathbb{S}_{r,A} = \mathcal{R}(S_{r,A})$ and $\mathbb{T}_{r,A} = \mathcal{R}(T_{r,A})$. If $\sigma_r > \sigma_{r+1}$, then $\mathbb{S}_{r,A}$ and $\mathbb{T}_{r,A}$ are the left and right *leading singular spaces*, respectively, associated with the r largest singular values of the matrix A . The left singular spaces of a matrix A are the right singular spaces of its transpose A^T and vice versa. All matrix bases for the singular spaces $\mathbb{S}_{r,A}$ and $\mathbb{T}_{r,A}$ are given by the matrices $S_{r,A}X$ and $T_{r,A}Y$, respectively, for nonsingular $r \times r$ matrices X and Y . The bases are orthogonal where the matrices X and Y are orthogonal.

2.5 Condition number, numerical rank and generic conditioning profile

$\kappa(A) = \frac{\sigma_1(A)}{\sigma_\rho(A)} = \|A\| \|A^+\|$ is the condition number of an $m \times n$ matrix A of a rank ρ . Such matrix is *ill conditioned* if the ratio $\sigma_1(A)/\sigma_\rho(A)$ is large. If the ratio is reasonably bounded, then the matrix is *well conditioned*. See [GL96, Sections 2.3.2, 2.3.3, 3.5.4, 12.5], [H02, Chapter 15], [KL94], and [S98, Section 5.3] on the estimation of matrix norms and condition numbers. An $m \times n$ matrix A has a *numerical rank* $r = \text{nrank}(A) \leq \rho = \text{rank}(A)$ if the ratios $\sigma_j(A)/\|A\|$ are small for $j > r$ but not for $j \leq r$.

Remark 2.1. One can specify the adjective “small” above as “smaller than a fixed positive tolerance” and similarly specify “closely” and “well conditioned”. The specification can be a challenge, e.g., for the matrix $\text{diag}(1.1^{-j})_{j=0}^{999}$.

If a well conditioned $m \times n$ matrix A has a rank $\rho < l = \min\{m, n\}$, then all its close neighbors have numerical rank ρ and almost all of them have rank l . Conversely, if a matrix A has a positive numerical rank $r = \text{nrank}(A)$, then the r -truncation A_r is a well conditioned rank- r approximation to the matrix A within the error norm bound $\sigma_{r+1}(A)$ (cf. (2.8)). It follows that a matrix is ill conditioned if and only if it is close to a matrix having a smaller rank and that a matrix has a numerical rank r if and only if it can be closely approximated by a well conditioned matrix having rank r . Rank-revealing factorizations of a matrix A that has a small numerical rank r , but possibly has a large rank ρ , produce its rank- r approximation at a lower computational cost [GE96], [HP92], [P00a], whereas the randomized algorithms of [HMT11] decrease the computational cost further.

An $m \times n$ matrix has *generic conditioning profile* if it has a numerical rank r and if its leading $i \times i$ blocks are nonsingular and well conditioned for $i = 1, \dots, r$. We call such a matrix *strongly well conditioned* if it has full numerical rank $r = \min\{m, n\}$.

2.6 Block Gaussian elimination and GENP

For a nonsingular 2×2 block matrix $A = \begin{pmatrix} B & C \\ D & E \end{pmatrix}$ with a nonsingular *pivot block* $B = A^{(k)}$, define $S = S(A^{(k)}, A) = E - DB^{-1}C$, the *Schur complement* of $A^{(k)}$ in A , and the block factorization,

$$A = \begin{pmatrix} I_k & O_{k,r} \\ DB^{-1} & I_r \end{pmatrix} \begin{pmatrix} B & O_{k,r} \\ O_{r,k} & S \end{pmatrix} \begin{pmatrix} I_k & B^{-1}C \\ O_{k,r} & I_r \end{pmatrix}. \quad (2.9)$$

Apply this factorization recursively to the pivot block B and its Schur complement S and arrive at the block Gaussian elimination process, completely defined by the sizes of the pivot blocks. The

recursive process either fails, where its pivot block turns out to be singular, in particular where it is a vanishing pivot entry of GENP, or can continue until all pivot blocks become nonzero scalars. When this occurs we arrive at GENP. Factorization (2.9) defines the block elimination of the first k columns of the matrix A , whereas $S = S(A^{(k)}, A)$ is the matrix computed at this elimination step. Now assume that the pivot dimensions d_1, \dots, d_r and $\bar{d}_1, \dots, \bar{d}_r$ of two block elimination processes sum to the same integer k , that is $k = d_1 + \dots + d_r = \bar{d}_1 + \dots + \bar{d}_r$. Then verify that both processes produce the same Schur complement $S = S(A^{(k)}, A)$.

Theorem 2.3. *In every step of the recursive block factorization process based on (2.9) every diagonal block of a block diagonal factor is either a leading block of the input matrix A or the Schur complement $S(A^{(h)}, A^{(k)})$ for some integers h and k such that $0 < h < k \leq n$ and $S(A^{(h)}, A^{(k)}) = (S(A^{(h)}, A))^{(h)}$.*

Corollary 2.3. *The recursive block factorization process based on equation (2.9) can be completed by involving no vanishing pivot elements and no singular pivot blocks if and only if the input matrix A has generic rank profile.*

Proof. Combine Theorem 2.3 with the equation $\det A = (\det B) \det S$, implied by (2.9). \square

The following theorem bounds the norms of all pivot blocks and their inverses. Consequently it bounds the condition numbers of these blocks, which are precisely the quantities responsible for safe numerical performance of block Gaussian elimination and GENP. The theorem expresses these bounds in terms of the norms of the leading blocks of the input matrix, and it remains to bound these norms, which we are going to do in Sections 3 and 5.2.

Theorem 2.4. *(Cf. [PQZ13, Theorem 5.1].) Assume GENP or block Gaussian elimination applied to an $n \times n$ matrix A and write $N = \|A\|$ and $N_- = \max_{j=1}^n \|(A^{(j)})^{-1}\|$, and so $N_- N \geq \|A\| \|A^{-1}\| \geq 1$. Then the absolute values of all pivot elements of GENP and the norms of all pivot blocks of block Gaussian elimination do not exceed $N_+ = N + N_- N^2$, whereas the absolute values of the reciprocals of these elements and the norms of the inverses of the blocks do not exceed N_- .*

Proof. Observe that the inverse S^{-1} of the Schur complement S in (2.9) is the southeastern block of the inverse A^{-1} and obtain $\|B\| \leq N$, $\|B^{-1}\| \leq N_-$, and $\|S^{-1}\| \leq \|A^{-1}\| \leq N_-$. Moreover $\|S\| \leq N + N_- N^2$, due to (2.9). Now the claimed bound follows from Theorem 2.3. \square

We can invert equation (2.9) to obtain $A^{-1} = \begin{pmatrix} I_k & -B^{-1}C \\ O_{k,r} & I_r \end{pmatrix} \begin{pmatrix} B^{-1} & O_{k,r} \\ O_{r,k} & S^{-1} \end{pmatrix} \begin{pmatrix} I_k & O_{k,r} \\ -DB^{-1} & I_r \end{pmatrix}$ and can extend this factorization to recursive block factorization of the inverse matrix A^{-1} provided that the matrix A and all pivot blocks of this factorization are nonsingular.

Remark 2.2. *For a strongly nonsingular input matrix A block factorization (2.9) can be extended to computing the complete recursive factorization, which defines GENP. By virtue of Theorem 2.4 the norms of the inverses of all pivot blocks involved in this computation are at most N_- . If the matrix A is also strongly well conditioned, then we have a reasonable upper bound on N_- , and so in view of Theorem 2.1 the inversion of all pivot blocks is numerically safe. In this case we say that GENP is locally safe for the matrix A . Locally safe recursive factorization involves neither divisions by small pivot entries (avoiding them is the purpose of pivoting) nor inversions of ill conditioned pivot blocks. Let us also compare the magnification of the perturbation norm bound of Theorem 2.1 in GEPP and in the process of recursive factorization, which defines GENP and block Gaussian elimination. We observe immediately that in the recursive factorization only the factors of the leading blocks and the Schur complements can contribute to this magnification, namely at most $\log_2(n)$ such factors can contribute to the norm of each of the output triangular or block triangular factors L and U . This implies the upper bound $(N_+ N_-)^{\log_2(n)}$ on their norms, which can be compared favorably to the sharp upper bound 2^{n-1} on the growth factor for GEPP (cf. [GL96, page 119] and [S98, Theorem 3.4.12]).*

3 Singular values of the matrix products (deterministic estimates) and GENP with preprocessing

Fact 2.1 reduces the tasks of inverting a nonsingular and well conditioned matrix A and solving a linear system $A\mathbf{x} = \mathbf{b}$ to similar tasks for the matrix FAH and multipliers F and H of our choice. Remark 2.2 motivates the choice for which the matrix FAH is strongly nonsingular and strongly well conditioned. In Section 5.2 we prove that this is likely to occur already where one of the multipliers F and H is the identity matrix I and another one is a Gaussian matrix, and therefore also where both F and H are independent Gaussian matrices. In this section we prepare background for that proof by estimating the norms of the inverses of the matrices $(FA)_{k,k} = F_{k,m}A_{m,k}$ and $(AH)_{k,k} = A_{k,n}H_{n,k}$ for general (possibly nonrandom) multipliers F and H . (Clearly the norms of the matrices themselves are bounded as follows, $\|(FA)_{k,k}\| \leq \|F_{k,m}\| \|A_{m,k}\|$ and $\|(AH)_{k,k}\| \leq \|A_{k,n}\| \|H_{n,k}\|$.) We will keep writing M_s^T and M_s^+ for $(M_s)^T$ and $(M_s)^+$, respectively, where, say, M can stand for A , F , or H and s can stand for k , A , or a pair (k, l) or (A, r) . We begin with two simple lemmas.

Lemma 3.1. *If S and T are square orthogonal matrices, then $\sigma_j(SA) = \sigma_j(AT) = \sigma_j(A)$ for all j .*

Lemma 3.2. *Suppose $\Sigma = \text{diag}(\sigma_i)_{i=1}^n$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$, $F \in \mathbb{R}^{r \times n}$, and $H \in \mathbb{R}^{n \times r}$. Then $\sigma_j(F\Sigma) \geq \sigma_j(F)\sigma_n$, $\sigma_j(\Sigma H) \geq \sigma_j(H)\sigma_n$ for all j . If also $\sigma_n > 0$, then $\text{rank}(F\Sigma) = \text{rank}(F)$, whereas $\text{rank}(\Sigma H) = \text{rank}(H)$.*

The following theorem bounds the norms $\|(FA)^+\|$ and $\|(AH)^+\|$ for three matrices A , F and H .

Theorem 3.1. *Suppose $A \in \mathbb{R}^{m \times n}$, $F \in \mathbb{R}^{r \times m}$, $H \in \mathbb{R}^{n \times r}$, $r \leq \rho$ for $\rho = \text{rank}(A)$, $A = S_A \Sigma_A T_A^T$ (cf. (2.2)), $\hat{F} = FS_A$, and $\hat{H} = T_A^T H$. Then*

$$\sigma_j(FA) \geq \sigma_k(A) \sigma_j(\hat{F}_{r,k}) \text{ for all } k \leq m \text{ and all } j, \quad (3.1)$$

$$\sigma_j(AH) \geq \sigma_l(A) \sigma_j(\hat{H}_{l,r}) \text{ for all } l \leq n \text{ and all } j. \quad (3.2)$$

Proof. Note that $AH = S_A \Sigma_A T_A^T H$, and so $\sigma_j(AH) = \sigma_j(\Sigma_A T_A^T H) = \sigma_j(\Sigma_A \hat{H})$ for all j by virtue of Lemma 3.1, because S_A is a square orthogonal matrix. Furthermore it follows from Fact 2.2 that $\sigma_j(\Sigma_A \hat{H}) \geq \sigma_j(\Sigma_{l,A} \hat{H}_{l,r})$ for all $l \leq n$. Combine this bound with the latter equations, then apply Lemma 3.2, and obtain bound (3.2). Similarly deduce bound (3.1). \square

Corollary 3.1. *Keep the assumptions of Theorem 3.1. Then (i) $\sigma_r(AH) \geq \sigma_\rho(A) \sigma_r(\hat{H}_{\rho,r}) = \sigma_r(\hat{H}_{\rho,r})/\|A^+\|$, (ii) $\|(AH)^+\| \leq \|A^+\| \|\hat{H}_{\rho,r}^+\|$ if $\text{rank}(AH) = \text{rank}(\hat{H}_{\rho,r}) = r$, (iii) $\sigma_r(FA) \geq \sigma_\rho(A) \sigma_r(\hat{F}_{r,\rho}) = \sigma_r(\hat{F}_{r,\rho})/\|A^+\|$, (iv) $\|(FA)^+\| \leq \|A^+\| \|\hat{F}_{r,\rho}^+\|$ if $\text{rank}(FA) = \text{rank}(\hat{F}_{r,\rho}) = r$.*

Proof. Substitute $j = r$ and $l = \rho$ into bound (3.2), recall (2.4), and obtain part (i). If $\text{rank}(AH) = \text{rank}(\hat{H}_{l,r}) = r$, then apply (2.4) to obtain that $\sigma_r(AH) = 1/\|(AH)^+\|$ and $\sigma_r(\hat{H}_{l,r}) = 1/\|\hat{H}_{l,r}^+\|$. Substitute these equations into part (i) and obtain part (ii). Similarly prove parts (iii) and (iv). \square

Let us extend Theorem 3.1 to the leading blocks of the matrix products.

Corollary 3.2. *Keep the assumptions of Theorem 3.1 and fix two positive integers k and l such that $k \leq m$, $l \leq n$. Then (i) $\|(FA)_{k,l}^+\| \leq \|\hat{F}_{k,m}^+\| \|A_{m,l}^+\| \leq \|\hat{F}_{k,m}^+\| \|A^+\|$ if $m \geq n = \rho$ and if the matrices $(FA)_{k,l}$ and $\hat{F}_{k,m}$ have full rank, whereas (ii) $\|(AH)_{k,l}^+\| \leq \|\hat{H}_{n,l}^+\| \|A_{k,n}^+\| \leq \|\hat{H}_{n,l}^+\| \|A^+\|$ if $m = \rho \leq n$ and if the matrices $(AH)_{k,l}$ and $\hat{H}_{n,l}$ have full rank.*

Proof. Recall that $(FA)_{k,l} = F_{k,m}A_{m,l}$ and the matrix $A_{m,l}$ has full rank if $m \geq n = \rho$. Apply Corollary 3.1 for A and F replaced by $A_{m,l}$ and $F_{k,m}$, respectively, and obtain that $\|(FA)_{k,l}^+\| \leq \|\hat{F}_{k,m}^+\| \|A_{m,l}^+\|$. Combine (2.4) and Corollary 2.1 and deduce that $\|A_{m,l}^+\| \leq \|A^+\|$. Combine the two latter inequalities to complete the proof of part (i). Similarly prove part (ii). \square

The following definition formalizes the assumptions of Corollaries 3.1 and 3.2.

Definition 3.1. Assume the matrices A , F , \widehat{F} , H , and \widehat{H} of Theorem 3.1. Then the matrix pair (A, H) (resp. (F, A)) has full rank if the matrices AH , $\widehat{H}_{\rho,r}$ (resp. FA , $\widehat{F}_{r,\rho}$) and A have full rank. This pair has full rank and is well conditioned if in addition the matrices $\widehat{H}_{\rho,r}$ (resp. $\widehat{F}_{r,\rho}$) and A are well conditioned, whereas it has generic rank profile if $\text{rank}(A) = \rho$ and $\text{rank}((AH)_{k,k}) = \text{rank}(\widehat{H}_{\rho,k}) = k$ (resp. $\text{rank}((FA)_{k,k}) = \text{rank}(\widehat{F}_{k,\rho}) = k$) for $k = 1, \dots, r$. The pair has generic rank profile and is strongly well conditioned if in addition the matrices $\widehat{H}_{\rho,k}$ (resp. $\widehat{F}_{k,\rho}$) for $k = 1, \dots, r$ are well conditioned.

Remark 3.1. Fact 2.1, Corollary 2.3 and Theorem 2.4 together imply the following guiding rule. Suppose that $A \in \mathbb{R}^{m \times n}$, $F \in \mathbb{R}^{r \times m}$, $H \in \mathbb{R}^{n \times r}$, $r \leq \text{rank}(A)$, and the matrix pair (A, H) for $m \leq n$ or (F, A) for $m \geq n$ has generic rank profile and is strongly well conditioned. Then GENP is locally safe for the matrix products AH or FA , respectively (see Remark 2.2 on the concept “locally safe”).

4 Low-rank approximation

4.1 The basic algorithm

Suppose we seek a rank- r approximation of a matrix A that has a small numerical rank r . One can solve this problem by computing the SVD of the matrix A or its rank-revealing factorization [GE96], [HP92], [P00a], but substantial benefits of using random matrix multipliers instead have been shown in the paper [HMT11]. Assume an $m \times n$ matrix A that has a small numerical rank r and also assume a Gaussian $n \times r$ multiplier H . Then according to [HMT11, Theorem 4.1] the column span of the matrices AH and $Q(AH)$ is likely to approximate the leading singular space $\mathbb{S}_{r,A}$ of the matrix A , and if it does, then it follows that the rank- r matrix $QQ^T A$ approximates the matrix A . In this section we recall the algorithm supporting this theorem, but keep the multiplier H unspecified. In the next subsection we estimate the approximation errors of that algorithm assuming no randomization and no oversampling, suggested in [HMT11]. In the next section we extend our study to the case where H is a Gaussian multiplier.

Algorithm 4.1. Low-rank approximation of a matrix. (Cf. Remarks 4.1–4.3.)

INPUT: A matrix $A \in \mathbb{R}^{m \times n}$, its numerical rank r , and an integer $p \geq 2$ such that $r+p \leq \min\{m, n\}$.

OUTPUT: an orthogonal matrix $Q \in \mathbb{R}^{m \times (r+p)}$ such that the matrix $QQ^T A \in \mathbb{R}^{m \times n}$ has a rank at most $r+p$ and approximates the matrix A .

INITIALIZATION: Generate an $n \times (r+p)$ matrix H .

COMPUTATIONS:

1. Compute an $n \times (r+p)$ orthogonal matrix $Q = Q(AH)$, sharing its range with the matrix AH .
2. Compute and output the matrix $R_{AH}A = QQ^T A$ and stop.

This basic algorithm from [HMT11] uses $O((r+p)mn)$ flops overall.

Remark 4.1. One can devise a dual variation of Algorithm 4.1, which computes the orthogonal $(r+p) \times n$ matrix $Q = Q(FA)$ for a proper $(r+p) \times m$ pre-multiplier F and which approximates an orthogonal basis for the leading singular space $\mathbb{T}_{r,A}$. In this case the matrix $(P_{FA}A^T)^T = AQ^T Q$ of the rank $r+p$ approximates the matrix A .

4.2 Analysis of the basic algorithm assuming no randomization and no oversampling

In Corollaries 4.1 and 4.2 of this subsection we estimate the error norms for the approximations computed by Algorithm 4.1 whose oversampling parameter p is set to 0, namely for the approximation of an orthogonal matrix basis for the leading singular space $\mathbb{S}_{r,A}$ (by the matrix Q of the algorithm) and of a rank- r approximation of the matrix A . We first recall the following results.

Theorem 4.1. (Cf. (2.5).) Suppose A is an $m \times n$ matrix, $S_A \Sigma_A T_A^T$ is its SVD, r is an integer, $0 < r \leq l = \min\{m, n\}$, and $Q = Q_{r,A}$ is an orthogonal matrix basis for the space $\mathbb{S}_{r,A}$. Then $\|A - QQ^T A\| = \sigma_{r+1}(A)$.

Theorem 4.2. Assume two matrices $A \in \mathbb{R}^{m \times n}$ and $H \in \mathbb{R}^{n \times r}$ and the two matrices A_r and \bar{A}_r of (2.7). Then (i) $AH = A_r H + \bar{A}_r H$ where $A_r H = S_{r,A} U$, $U = \Sigma_{r,A} T_{r,A}^T H$ and (ii) the columns of the matrix $A_r H$ span the space $\mathbb{S}_{r,A}$ if $\text{rank}(A_r H) = r$.

These results together imply that the columns of the matrix $Q(AH)$ form an approximate orthogonal basis of the linear space \mathbb{S}_A , and next we estimate the error norms of this approximations.

Theorem 4.3. Keep the assumptions of Theorem 4.2. Then (i) $\|\bar{A}_r H\|_F \leq \sigma_{r+1}(A) \|H\|_F$. (ii) Furthermore if the matrix $T_{r,A}^T H$ is nonsingular, then $\|(A_r H)^+\| \leq \|(T_{r,A}^T H)^{-1}\| / \sigma_r(A)$.

Proof. Note that $\|\bar{A}_r H\|_F = \|\bar{S}_{A,r} \bar{\Sigma}_{A,r} \bar{T}_{A,r}^T H\|_F \leq \|\bar{\Sigma}_{A,r} \bar{T}_{A,r}^T H\|_F$ by virtue of bound (2.1). Combine this bound with Lemma 3.2 and obtain that $\|\bar{A}_r H\|_F \leq \sigma_{r+1}(A) \|\bar{T}_{A,r}^T H\|_F$, which is not greater than $\sigma_{r+1}(A) \|H\|_F$ by virtue of bound (2.1). This proves part (i). Part (ii) follows because $(A_r H)^+ = (S_{r,A} \Sigma_{r,A} T_{r,A}^T H)^{-1} = (T_{r,A}^T H)^{-1} \Sigma_{r,A}^{-1} S_{r,A}^T$ if the matrix $T_{r,A}^T H$ is nonsingular and because $\|S_{r,A}\| = 1$, whereas $\|\Sigma_{r,A}^{-1}\| = 1/\sigma_r(A)$. \square

Combine Theorems 2.2, 4.2, and 4.3 to obtain the following estimates.

Corollary 4.1. Keep the assumptions of Theorem 4.2, let the matrix $T_{r,A}^T H$ be nonsingular, and write $\|E\|_F = \sigma_{r+1}(A) \|H\|_F$ and $\Delta_+ = \sqrt{2} \|E\|_F \|(T_{r,A}^T H)^{-1}\| / \sigma_r(A)$ and obtain that $\Delta = \|Q(A_r H)^T - Q(AH)^T\| \leq \Delta_+ + O(\|E\|_F^2)$ for $\Delta_+ = \sqrt{2} \|H\|_F \|(T_{r,A}^T H)^{-1}\| \sigma_{r+1}(A) / \sigma_r(A)$.

Next we combine Corollary 2.2 with Theorem 4.1 and employ the orthogonal projection $P_{AH} = Q(AH)Q(AH)^T$ (cf. (2.5)) to extend the latter estimate to bound the error norm of low-rank approximation of a matrix A by means of Algorithm 4.1.

Corollary 4.2. Keep the assumptions of Corollary 4.1 and write $\Delta'_+ = \sigma_{r+1}(A) + 2\Delta_+ \|A\|$. Then $\Delta' = \|A - P_{AH} A\| \leq \Delta'_+ + O(\|E\|_F^2 \|A\|)$.

Proof. Note that $\|A - P_{AH} A\| \leq \|A - P_M A\| + \|(P_M - P_{AH})A\|$ for any $m \times r$ matrix M . Write $M = A_r H$, apply Theorem 4.1 and obtain $\|A - P_M A\| = \sigma_{r+1}(A)$. Corollaries 2.2 and 4.1 together imply that $\|(P_M - P_{AH})A\| \leq \|A\| \|P_{A_r H} - P_{AH}\| \leq 2\Delta_+ \|A\|$. Combine the above relationships. \square

Remark 4.2. Write $B_i = (A^T A)^i A$ and recall that $\sigma_j(B_i) = (\sigma_j(A))^{2i+1}$ for all positive integers i and j . Therefore one can apply the power transforms $A \rightarrow B_i$ for $i = 1, 2, \dots$ to increase the ratio $\sigma_r(A) / \sigma_{r+1}(A)$, which shows the gap between the two singular values. Consequently the bound Δ_+ on the error norm of the approximation of an orthogonal basis of the leading singular space $\mathbb{S}_{r,A}$ by $Q(B_i H)$ is expected to decrease as i increases (cf. [HMT11, equation (4.5)]). We use the matrix $AH = B_0 H$ in Algorithm 4.1, but suppose we replace it with the matrices $B_i H$ for small positive integer i , or even for $i = 1$, which would amount just to symmetrization. Then it would follow that we would obtain low-rank approximation with the optimum error $\sigma_{r+1}(A)$ up to the terms of higher order in $\sigma_{r+1}(A) / \sigma_r(A)$ as long as the value $\|H\|_F \|(T_{r,A}^T H)^{-1}\|$ is reasonably bounded from above. The power transform $A = B_0 \rightarrow B_i$ requires to increase by a factor of $2i + 1$ the number of matrix-by-vector multiplications involved, but for small positive integers i the additional computational cost is still dominated by the costs of computing SVD and rank-revealing factorizations.

Remark 4.3. Let us summarize our analysis. Suppose that the ratio $\sigma_r(A) / \sigma_{r+1}(A)$ is large, whereas the matrix product $P = T_{r,A}^T H$ has full rank r and is well conditioned. We can restate these assumptions in terms of Definition 3.1 by saying that the matrix pair (A_r, H) has full rank and is well conditioned. Now set to 0 the oversampling integer parameter p of Algorithm 4.1. Then this algorithm outputs (i) close approximation $Q(AH)$ to an orthogonal bases for the leading singular space $\mathbb{S}_{r,A}$ of the input matrix A and (ii) a rank- r approximation to this matrix. Up to the terms of higher order, the error norm of the latter approximation is within a factor of $1 + \|H\|_F \|(T_{r,A}^T H)^{-1}\| / \sigma_r(A)$ from the optimal bound $\sigma_{r+1}(A)$. By applying the above power transform of the input matrix A at the dominated computational cost we can decrease the error norm even below the value $\sigma_{r+1}(A)$.

5 Benefits of using random matrix multipliers

In Section 5.1 we define Gaussian matrices and recall their basic properties. In Sections 5.2 and 5.3 we prove that the pairs (F, A) or (H, A) for assumed input matrices A and Gaussian matrices F and H are expected to satisfy the assumptions of Remarks 3.1 and 4.3, which implies randomized support of locally safe GENP and low-rank approximation, respectively. In Section 5.4 we comment on using non-Gaussian random multipliers.

5.1 A Gaussian matrix, its rank, norm and condition estimates

Definition 5.1. *A matrix is said to be standard Gaussian random (hereafter we say just Gaussian) if it is filled with i.i.d. Gaussian random variables having mean 0 and variance 1.*

Fact 5.1. *A Gaussian matrix is rank deficient with probability 0.*

Proof. Assume a rank deficient $m \times n$ matrix of a rank ρ . Then the determinants of all its $n \times n$ submatrices vanish. This implies $(m - \rho)(n - \rho)$ polynomial equations on the entries, that is the rank deficient matrices form an algebraic variety of a lower dimension $mn - (m - \rho)(n - \rho) = (m + n - \rho)\rho$ in the linear space $\mathbb{R}^{m \times n}$ (cf. Fact A.1). (V is an algebraic variety of a dimension $d \leq N$ in the space \mathbb{R}^N if it is defined by $N - d$ polynomial equations and cannot be defined by fewer equations.) Clearly such a variety has Lebesgue (uniform) and Gaussian measure 0, both being absolutely continuous with respect to one another. \square

Corollary 5.1. *A Gaussian matrix has generic rank profile with probability 1.*

Definition 5.2. $\nu_{j,m,n}$ denotes the random variables $\sigma_j(G)$ for a Gaussian $m \times n$ matrix G and all j , whereas $\nu_{m,n}$, $\nu_{F,m,n}$, $\nu_{m,n}^+$, and $\kappa_{m,n}$ denote the random variables $\|G\|$, $\|G\|_F$, $\|G^+\|$, and $\kappa(G) = \|G\| \|G^+\|$, respectively. Note that $\nu_{j,n,m} = \nu_{j,m,n}$, $\nu_{n,m} = \nu_{m,n}$, $\nu_{n,m}^+ = \nu_{m,n}^+$, and $\kappa_{n,m} = \kappa_{m,n}$.

Theorem 5.1. (Cf. [DS01, Theorem II.7].) *Suppose $h = \max\{m, n\}$, $t \geq 0$, and $z \geq 2\sqrt{h}$. Then $\text{Probability}\{\nu_{m,n} > z\} \leq \exp(-(z - 2\sqrt{h})^2/2)$ and $\text{Probability}\{\nu_{m,n} > t + \sqrt{m} + \sqrt{n}\} \leq \exp(-t^2/2)$.*

Theorem 5.2. *Suppose $m \geq n$, and $x > 0$ and write $\Gamma(x) = \int_0^\infty \exp(-t)t^{x-1}dt$ and $\zeta(t) = t^{m-1}m^{m/2}2^{(2-m)/2} \exp(-mt^2/2)/\Gamma(m/2)$. Then*

- (i) *Probability $\{\nu_{m,n}^+ \geq m/x^2\} < \frac{x^{m-n+1}}{\Gamma(m-n+2)}$ for $n \geq 2$ and*
- (ii) *Probability $\{\nu_{m,1}^+ \geq x\} \leq (m/2)^{(m-2)/2}/(\Gamma(m/2)x^m)$.*

Proof. (i) See [CD05, Proof of Lemma 4.1]. (ii) $G \in \mathbb{R}^{m \times 1}$ is a vector of length m . So, with probability 1, $G \neq 0$, $\text{rank}(G) = 1$, $\|G^+\| = 1/\|G\|$, and consequently $\text{Probability}\{\|G^+\| \geq x\} = \text{Probability}\{\|G\| \leq 1/x\} = \int_0^{1/x} \zeta(t)dt$. Note that $\exp(-mt^2/2) \leq 1$, and so $\int_0^{1/x} \zeta(t)dt < c_m \int_0^{1/x} t^{m-1}dt = c_m/(mx^m)$ where $c_m = m^{m/2}2^{(2-m)/2}/\Gamma(m/2)$. \square

The following condition estimates from [CD05, Theorem 4.5] are quite tight for large values x , but for $n \geq 2$ even tighter estimates (although more involved) can be found in [ES05]. (See [D88] and [E88] on the early study.)

Theorem 5.3. *If $m \geq n \geq 2$, then $\text{Probability}\{\kappa_{m,n}m/(m - n + 1) > x\} \leq \frac{1}{2\pi}(6.414/x)^{m-n+1}$ for $x \geq m - n + 1$, whereas $\kappa_{m,1} = 1$ with probability 1.*

Corollary 5.2. *A Gaussian matrix has generic rank profile with probability 1 and is expected to be well conditioned.*

Proof. Combine Corollary 5.1 and Theorem 5.3. \square

Now recall that “actual outcome” of Algorithm 4.1 “is very close to the typical outcome because of the measure concentration effect” [HMT11, page 226]. This motivates us to reproduce next some known bounds for the expected values of the norms and condition numbers of random matrices (cf. Appendix B).

Theorem 5.4. *It holds that (i) $\mathbb{E}(\nu_{n,n}) \leq 2\sqrt{n}$, whereas (ii) $\mathbb{E}(\log(k_{m,n})) \leq \log \frac{n}{m-n+1} 2.258$ for $m \geq n \geq 2$.*

Proof. See [S91] on part (i) and see [CD05, Theorem 6.1] on part (ii). \square

The bounds of part (i) of the theorem are quite tight (cf. Theorem 5.1). The bounds of part (ii) imply the following more specific estimates.

Corollary 5.3. $\mathbb{E}(\log(k_{n,n})) \leq \log(n) + 2.258$, whereas $\mathbb{E}(k_{m,n}) \leq 5(1 - 1/k)$ for $k + 1 = \frac{m}{n-1}$ and $m \gg n \gg 1$.

5.2 Supporting GENP with Gaussian multipliers

Lemma 5.1. *Suppose H is a Gaussian matrix, S and T are orthogonal matrices, $H \in \mathbb{R}^{m \times n}$, $S \in \mathbb{R}^{k \times m}$, and $T \in \mathbb{R}^{n \times k}$ for some k, m , and n . Then SH and HT are Gaussian matrices.*

Theorem 5.5. *Suppose $A \in \mathbb{R}^{m \times n}$, $F \in \mathbb{R}^{r \times m}$, $H \in \mathbb{R}^{m \times r}$, F and H are Gaussian matrices, and $\text{rank}(A) = \rho$. Then $\text{rank}(FA) = \text{rank}(AH) = \min\{r, \rho\}$ with probability 1.*

Proof. Suppose $A = S_A \Sigma_A T_A^T$ is SVD of (2.2). Then $FA = FS_A \Sigma_A T_A^T = G \Sigma_A T_A^T$ where $G = FS_A$ is a Gaussian $r \times m$ matrix by virtue of Lemma 5.1. Clearly $\text{rank}(FA) = \text{rank}(G \Sigma_A T_A^T) = \text{rank}(G \Sigma_A)$ because T_A is a square orthogonal matrix. Moreover $\text{rank}(G \Sigma_A) = \text{rank}(GD_\rho)$ where $D_\rho = \text{diag}(I_\rho, O_{m-\rho, n-\rho})$, and so GD_ρ is a Gaussian $r \times \rho$ matrix because it is a submatrix of the Gaussian matrix G . Therefore $\text{rank}(FA) = \text{rank}(GD_\rho)$ is equal to $\min\{r, \rho\}$ with probability 1 by virtue of Fact 5.1. Similarly obtain that $\text{rank}(AH) = \min\{r, \rho\}$ with probability 1. \square

Corollary 5.4. *Keep the assumptions of Theorem 3.1. Suppose the matrix A has full rank $\rho = \min\{m, n\}$, $k \leq r \leq \rho$, and F and H are Gaussian matrices. Then (i) so are the matrices $\widehat{F} = FS_A$, $\widehat{H} = T_A^T H$ and all their submatrices, in particular $\widehat{F}_{k,\rho}$ and $\widehat{H}_{\rho,k}$, (ii) with probability 1, $\text{rank}((AH)_{k,k}) = k$ if $m \leq n$, $\text{rank}((FA)_{k,k}) = k$ if $m \leq n$, and $\text{rank}(\widehat{H}_{\rho,k}) = \text{rank}(\widehat{F}_{k,\rho}) = k$, and (iii) $\|(AH)_{k,k}^+\| \leq \nu_{n,k}^+ \|A^+\|$ and $\|(FA)_{k,k}^+\| \leq \nu_{k,m}^+ \|A^+\|$ for all k and for the values $\nu_{g,h}^+$ of Definition 5.2.*

Proof. If H and F are Gaussian matrices, then so are the matrices \widehat{H} and \widehat{F} by virtue of Lemma 5.1. Consequently so are all their submatrices. This proves parts (i) and by virtue of Fact 5.1 also implies the equations $\text{rank}(\widehat{H}_{\rho,k}) = \text{rank}(\widehat{F}_{k,\rho}) = k$ of part (ii). Now recall that $(AH)_{k,k} = A_{k,n} H_{n,k}$ and consequently $\text{rank}((AH)_{k,k}) = \text{rank}(A_{k,n} H_{n,k})$, which is equal to $\text{rank}(A_{k,n})$ with probability 1 by virtue of Theorem 5.5 because $H_{n,k}$ is a Gaussian matrix and because $k \leq \rho \leq n$. Finally obtain that $\text{rank}(A_{k,n}) = k$ for $k \leq \rho = m$, and so $\text{rank}((AH)_{k,k}) = k$. Similarly prove that $\text{rank}((FA)_{k,k}) = k$ for $k \leq \rho = n$. Combine part (i) with Corollary 3.2 to obtain part (iii). \square

Corollary 5.5. *The choice of Gaussian multipliers F where $m \leq n$ or H where $m \geq n$ is expected to satisfy the assumptions of Remark 3.1 (thus supporting application of GENP to the matrix FA where $m \leq n$ or AH where $m \geq n$) provided that the $m \times n$ matrix A is nonsingular and well conditioned.*

Proof. Combine Corollaries 5.2 and 5.4. \square

5.3 Supporting low-rank approximation with Gaussian multipliers

Corollary 5.6. *Suppose $A \in \mathbb{R}^{m \times n}$, $A = S_A \Sigma_A T_A^T$ is its SVD of (2.2), $H = \mathbb{R}^{n \times r}$ is a Gaussian matrix, and $\text{rank}(A) = \rho \geq r$. Then the matrix $T_{r,A}^T H$ is Gaussian.*

Proof. $T_A^T H$ is a Gaussian matrix by virtue of Lemma 5.1. Therefore so is its square submatrix $T_{r,A}^T H$ as well. \square

Corollary 5.7. *Under the assumptions of Corollaries 4.1, 4.2, and 5.6, it holds that $\Delta = \|Q(A_r H)^T - Q(AH)^T\| \leq \Delta_+ + O(\|E\|_F^2)$ and $\Delta' = \|A - P_{AH}A\| \leq \Delta'_+ + O(\|E\|_F^2 \|A\|)$ for $\Delta_+ = \sqrt{2} \nu_{F,n,r} \nu_{r,r}^+ \sigma_{r+1}(A) / \sigma_r(A)$, $\Delta'_+ = \sigma_{r+1}(A) + 2\Delta_+ \|A\|$, and $\nu_{F,n,r}$ and $\nu_{r,r}^+$ of Definition 5.2.*

Corollary 5.8. *The choice of a Gaussian multiplier H is expected to satisfy the assumptions of Remark 4.3, thus supporting the application of Algorithms 4.1 where its oversampling integer parameter p is set to 0.*

Proof. Combine Corollaries 5.2 and 5.7. □

5.4 Random structured multipliers

This subsection involves complex matrices. A complex matrix M is unitary if $M^H M = I$ or $MM^H = I$ where M^H denotes its Hermitian transpose and where $M^H = M^T$ for a real matrix M .

What can motivate us for using non-Gaussian random multipliers? Given matrices $A \in \mathbb{R}^{m \times n}$ and $H \in \mathbb{R}^{n \times r}$, we compute the product AH by using $2mnr - mr$ flops (which means $2n^3 - n^2$ flops for $m = n = r$), but if H is a Toeplitz or circulant matrix, then we can compute such products by using order of $mn \log(r)$ flops (cf. [P01]), which means order of $n^2 \log(n)$ flops for $m = n = r$, and similarly for the customary random structured multipliers listed in [HMT11], that is the ones of subsampled random Fourier transform (SRFT), subsampled random Hadamard transform (SRHT), the chains of Givens rotations (CGR) of [HMT11, Section 11], and the chains of Householder reflections (CHR) of [PQZ13]. Similarly to CGR the latter matrix can be compressed by using the DFR multipliers of [HMT11, equation (4.6)]. Furthermore we need just n random parameters to define a Gaussian circulant $n \times n$ matrix $C = (c_{i-j \bmod n})_{i,j=0}^{n-1}$ or its leading $n \times r$ and $r \times n$ Toeplitz blocks $C_{n,r}$ and $C_{r,n}$, and similarly for the four other classes of structured matrices listed above.

At the end of this subsection we discuss some additional benefits of using circulant multipliers for solving Toeplitz linear systems of equations. Generally Gaussian circulant matrices are not unitary, but are expected to be very well conditioned (see sharp estimates in [PQa]), whereas $\kappa(T) \leq \kappa(C)$ for the leading $n \times r$ Toeplitz block $C_{n,r}$ of an $n \times n$ circulant matrix C by virtue of Fact 2.3, and so application of such well conditioned rectangular Toeplitz multipliers in Algorithm 4.1 is promising. Moreover we can alternatively employ complex random unitary $n \times n$ circulant matrix C of Example 5.2 below, as well as its $n \times r$ leading unitary Toeplitz block submatrix $C_{n,r}$ for $r < n$. The real circulant matrices of our alternative Example 5.1 are not unitary, but empirically tend to be well conditioned [PQa].

Hereafter $\omega_q = \exp(\frac{2\pi}{q}\sqrt{-1})$ denotes a q -th primitive root of unity. To simplify our notation we also write ω for ω_n . $\Omega = (\omega^{ij})_{i,j=0}^{n-1}$ and $\Omega^{-1} = \frac{1}{n}\Omega^H$ denote the discrete Fourier transform (hereafter we use the acronym *DFT*) at n points and its inverse, respectively.

Theorem 5.6. (Cf. [CPW74].) *Let C denote a circulant $n \times n$ matrix defined by its first column \mathbf{c} and write $\mathbf{u} = (u_i)_{i=1}^n = \Omega \mathbf{c}$. Then $C = \Omega^{-1} \text{diag}(u_j)_{j=1}^n \Omega$.*

Corollary 5.9. (Cf. [PQa].) *Assume a nonsingular circulant matrix C with the first column $\mathbf{c} = C\mathbf{e}_1$ and let $\mathbf{u} = \Omega \mathbf{c}$, as in Theorem 5.6. Then (i) $\|C\| = \|\text{diag}(\mathbf{u})\| = \max_{j=1}^n |u_j|$, $\|C^{-1}\| = \|(\text{diag}(\mathbf{u}))^{-1}\| = 1/\min_{j=1}^n |u_j|$, and so $\kappa(C) = \|C\| \|C^{-1}\| = \max_{i,j=1}^n |u_i/u_j|$. (ii) If \mathbf{c} is a Gaussian vector, then so is the vector \mathbf{u}/\sqrt{n} .*

Example 5.1. *Generation of random real circulant matrices. Generate the vector \mathbf{c} of n i.i.d. random real variables in the range $[-1, 1]$ under the uniform probability distribution on this range. Define an $n \times n$ circulant matrix C with the first column $\mathbf{c} = C\mathbf{e}_1$.*

Example 5.2. *Generation of random unitary circulant matrices.*

(i) *Generate a vector $\mathbf{u} = (u_j)_{j=1}^n$ where $u_j = \exp(2\pi\phi_j\sqrt{-1})$ (and so $|u_j| = 1$ for all i) and where ϕ_1, \dots, ϕ_n are n independent random real variables, e.g., Gaussian variables or variables uniformly distributed in the range $[0, 1]$.*

(ii) *Generate the matrices $\Omega = (\omega^{ij})_{i,j=0}^{n-1}$ and $\Omega^{-1} = \frac{1}{n}\Omega^H$ of the discrete Fourier transform at n points and its inverse, respectively, where $\omega = \exp(\frac{2\pi}{n}\sqrt{-1})$ is a primitive root of 1.*

(iii) Compute the vector $\mathbf{c} = \Omega^{-1}\mathbf{u}$ and output the unitary circulant matrix C defined by its first column $C\mathbf{e}_1 = \mathbf{c}$.

Example 5.3. For two fixed integers l and n , $1 < l < n$, SRFT (that is semisample random Fourier transform) $n \times l$ matrices are the matrices of the form $S = \sqrt{n/l} D\Omega R$ where D is a random $n \times n$ diagonal matrix whose diagonal entries are i.i.d. variables uniformly distributed on the unit circle $C(0, 1) = \{x : |x| = 1\}$, Ω is the DFT matrix, and R is a random $n \times l$ permutation matrix defined by random choice of l columns under the uniform probability distribution on the set of the n columns of the identity matrix I_n (cf. [HMT11, Section 11]).

Theorem 5.6 implies the following fact.

Corollary 5.10. Assume an $n \times l$ SRFT matrix S . Then $\sqrt{l/n} \Omega^{-1}S$ is an $n \times l$ submatrix of a unitary circulant $n \times n$ matrix.

Can we extend our results to non-Gaussian random multipliers? Fact 5.1 can be immediately extended if the assumed probability distribution is absolutely continuous with respect to the Lebesgue/Gaussian measures. In particular this is the case under the restrictions imposed by the Toeplitz, circulant, SRFT, SRHT, CGR and CHR matrix structures, and so we can define the random multipliers F and H accordingly and still satisfy with probability 1 the full rank and generic rank profile assumptions of Remarks 3.1 and 4.3. Moreover we satisfy these assumptions with probability close to 1 even where we fill the multipliers F and H with i.i.d. random variables defined under the uniform probability distribution over a fixed sufficiently large finite set (see Appendix C).

The assumptions of the two remarks about the conditioning of the matrices involved into the computations, however, fail if we choose a Gaussian random multiplier with a mean μ and a standard deviation σ such that $\mu \gg \sigma$ (say $\mu > 10 \log(n)\sigma$). Indeed in this case the matrices F and H are expected to be closely approximated by the rank-1 matrix $\mu\mathbf{e}\mathbf{e}^T$ where $\mathbf{e}^T = (1, 1, \dots, 1)$. Moreover our proofs supporting the conditioning assumptions of Remarks 3.1 and 4.3 rely on using Lemma 5.1, and we cannot extend this lemma to the case of non-Gaussian matrices. Nevertheless by allowing substantial oversampling, one can prove that SRFT multipliers are expected to support efficient low-rank approximation of a matrix having a small numerical rank, similarly to Gaussian multipliers applied with no oversampling.

Theorem 5.7. (Cf. [HMT11, Theorem 11.2]: Error bounds for low-rank approximation with SRFT). Fix four integers l, m, n , and r such that $4[\sqrt{r} + \sqrt{8 \log(rn)n}]^2 \log(r) \leq l \leq n$. Assume an $m \times n$ matrix A with singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots$, an $n \times l$ SRFT matrix S of Example 5.3, and $Y = AS$. Then $\|(I - P_Y)A\| \leq \sqrt{1 + 7n/l} \sigma_{r+1}$ and $\|(I - P_Y)A\|_F \leq \sqrt{1 + 7n/l} (\sum_{j>r} \sigma_j^2)^{1/2}$ with a probability $1 - O(1/r)$.

Remark 5.1. Clearly the theorem still holds if we replace the matrix S by the matrix US for a unitary matrix $U = (1/\sqrt{n})\Omega^{-1}$. In this case $US = CP$ for the matrix P of Example 5.3 and the matrix $C = \Omega^{-1}D\Omega$, which is circulant by virtue of Theorem 5.6. By virtue of Theorem 5.7 we can expect that Algorithm 4.1 would produce a rank- r approximation for an SRFT $n \times l$ multiplier H and consequently for H denoting the $n \times l$ submatrix CP of $n \times n$ a random unitary circulant matrix C made up of its l a randomly selected columns where the selection is defined by the matrix P of Example 5.3 for a sufficiently large integer l of order $r \log(r)$. Recall that multiplication of an $n \times n$ Toeplitz matrix by an $n \times l$ matrix $US = CP$ involves $O(nl \log(n))$ flops [P01], versus $O(n^2l)$ in the straightforward algorithm.

According to extensive tests by many researchers various random structured $n \times l$ multipliers (such as SRFT, SRHT, CGR and CHR matrices) support low-rank approximation already where the oversampling parameter $p = l - r$ is a reasonable constant (see [HMT11] and [M11]). In particular SRFT with oversampling by 20 is adequate in almost all applications of low-rank approximations [HMT11, page 279]. Likewise, in our extensive tests covered in Section 6.2, Toeplitz multipliers defined as the $n \times r$ leading blocks of $n \times n$ random circulant matrices of Examples 5.1 and 5.2 consistently supported low-rank approximation without oversampling. Example 5.4 below shows,

however, that a straightforward extension of Theorem 5.7 to supporting GENP with Gaussian circulant multipliers fails. We are going to use some auxiliary results of independent interest. Recall that $\omega_q = \exp(\frac{2\pi}{q}\sqrt{-1})$ denotes a q -th primitive root of unity.

Theorem 5.8. *Assume three integers h, k and n such that $2 < 2h < k, 2k^2 < n$. Write $k' = k + 1 - 2h, f_n = \omega_{2n}, g_{k',n} = |1 - \omega_n^{k'}|, s_n = \omega_n^i$ for $i = 1, \dots, k-1, t_j = f_n \omega_n^j$ for $j = h, \dots, k-h, c = \omega_{2n}^k, \theta = \max_{i=0, j=h}^{k-1, k-h} |c - s_i|/|c - t_j|, \delta = \min_{j=h}^{k-h} |t_j - c|$, and $\bar{\Omega}_{k,k'} = (\omega_n^{ij})_{i=0, j=h}^{k-1, k-h}$ to denote the $k \times k'$ submatrix of the matrix $\Omega = (\omega_n^{ij})_{i,j=0}^{n-1}$ of DFT at the n knots $1, \omega_n, \dots, \omega_n^{n-1}$. Assume that $\theta < 1$. Then $\|\bar{\Omega}_{k,k'}\| \leq \sqrt{kk'} \theta^{k'} / ((1 - \theta)\delta g_{k,n})$.*

Proof. Note that $\bar{\Omega}_{k',k}$ is a Vandermonde matrix $(s_i^j)_{i=h, j=0}^{k-h, k-1}$ with the knots $s_i = \omega_n^i$ for $i = h, \dots, k-h$ and apply the following well known expression (see, e.g., [Pb, equation (3.9)] for $V_s = \Omega_{k',k}, m = k, n = k',$ and $f = f_n$), $\bar{\Omega}_{k,k'} = \frac{f_n^{1-k'}}{\sqrt{k'}} \text{diag}(s_i^k - f_n^k)_{i=0}^{k-1} C_{s,t} \text{diag}(\omega_{k'}^j)_{j=h}^{k-h} \Omega_{k',k'} \text{diag}(f_n^j)_{j=h}^{k-h}$ where $C_{s,t} = (\frac{1}{s_i - t_j})_{i=0, j=h}^{k, k-h}$ is a Cauchy matrix. Write $c_+ = \max_{i=0, j=h}^{k-1, k-h} |\frac{1}{s_i - t_j}|$, note that $\|C_{s,t}\| \leq \sqrt{kk'} c_+, |f_n| = 1, \min_{i=0}^k |s_i^k - f_n^k| = g_{k',n}$, and the matrices $\text{diag}(\omega_{k'}^j)_{j=0}^{k-1}, \text{diag}(f_n^j)_{j=h}^{k-h}$, and $\frac{1}{\sqrt{k'}} \Omega_{k',k'}$ are unitary, and conclude that $\|\bar{\Omega}_{k,k'}\| \leq \frac{1}{g_{k',n}} \|C_{s,t}\|$. Apply [Pb, Corollary 7.1] to the transposed matrix $C_{s,t}^T$ (cf. [MRT05], [CGS07, Section 2.2]) and obtain that $c_+ \leq \theta^{k'} / ((1 - \theta)\delta)$. Combine the above bounds and obtain the theorem. \square

Corollary 5.11. *Under the assumptions of Theorem 5.8 suppose that n is a large integer. Then $\|\bar{\Omega}_{k,k'}\| \leq \epsilon = k\sqrt{k/k'} n^{2^{1-k'}} / (\pi^2 h)$.*

Proof. Note that $\theta < 1/2, g_{k',n} > \pi k' / n$, and $\delta > \pi h / k$ for large integers n . Substitute these bounds into the bound of Theorem 5.8 and obtain the corollary. \square

Corollary 5.12. *Under the assumptions of Corollary 5.11 the matrix $\Omega_{k,k}$ has ϵ -rank at most $2h$ where the value $\epsilon = k\sqrt{k/k'} n^{2^{1-k'}} / (\pi^2 h)$ converges to 0 as the integer k' grows large, and consequently the matrix $\Omega_{k,k}$ is ill conditioned.*

The corollary implies that the unitary DFT matrix Ω of a large size has ill conditioned leading blocks, and therefore is hard for GENP. The next example shows that post-multiplication by a Gaussian circulant matrix is not expected to fix the problem.

Example 5.4. *Assume a large integer n and the $n \times n$ matrix $A = \Omega$. Then a Gaussian circulant $n \times n$ post-multiplier $C = \Omega^{-1} D \Omega$ with Gaussian diagonal matrix $D = \text{diag}(g_j)_{j=1}^n$ (having i.i.d. Gaussian diagonal entries g_1, \dots, g_n) is not expected to support GENP. Indeed in this case $AC = D\Omega$, and so $(AC)_{k,k} = D_{k,k} \Omega_{k,k}, (AC)_{k,k}^{-1} = \Omega_{k,k}^{-1} D_{k,k}^{-1}$, and $\|(AC)_{k,k}^{-1}\| \geq \|\Omega_{k,k}^{-1}\| / g_k$. The random value $g_k = \max_{j=1}^k |d_j| \leq \nu_{k,1}$ is not expected to grow fast as k grows large, by virtue of Theorem 5.1, whereas the norm $\|\Omega_{k,k}^{-1}\|$ grows exponentially fast in k by virtue of Corollary 5.12 applied for k and k' satisfying $k^2 < n$ and, say, $2k' \geq k$. Therefore for such values the matrix $(AC)_{k,k}$ is expected to be ill conditioned, and so Gaussian circulant multipliers are not expected to support GENP for the matrix $A = \Omega$.*

The DFT matrix Ω is not a Toeplitz-like matrix, and in spite of having Example 5.4 one may still hope that random circulant multipliers are expected to support numerical application of GENP and block Gaussian elimination, in particular of the MBA celebrated superfast algorithm, which solves a strongly nonsingular Toeplitz or Toeplitz-like linear system of m equations in nearly linear arithmetic time (cf. [P01, Chapter 5], [PQZ11]). While fixing the well known problem with this algorithm, the preprocessing would keep it efficient, because circulant multipliers preserve Toeplitz structure. Besides circulant matrices one can consider using random factor-circulant multipliers, which also preserve Toeplitz structure. They are defined by scalar factors, which we can randomize. Furthermore we can apply simultaneously pre- and post-multipliers F and H associated with two independent random scalar factors f and h , respectively.

6 Numerical Experiments

We performed numerical experiments with random general, circulant and Toeplitz matrices by using MATLAB in the Graduate Center of the City University of New York on a Dell computer with a Intel Core 2 2.50 GHz processor and 4G memory running Windows 7. In particular we generated Gaussian matrices by using the standard normal distribution function `randn` of MATLAB, and we use the MATLAB function `rand` for generating numbers in the range $[0, 1]$ under the uniform probability distribution function for Example 5.1. We display our estimates obtained in terms of the spectral matrix norm but our tests showed similar results where we used the Frobenius norm instead.

6.1 GENP with Gaussian and random circulant multipliers

We tested GENP and preprocessed GENP for $n \times n$ DFT matrices $A = \Omega$ and for the matrices A generate as follows. We fixed $n = 2^s$ and $k = n/2$ for $s = 6, 7, 8, 9, 10$, and first, by following [H02, Section 28.3], generated a $k \times k$ matrix $A_k = U\Sigma V^T$ where we chose $\Sigma = \text{diag}(\sigma_i)_{i=1}^k$ with $\sigma_i = 1$ for $i = 1, \dots, k-4$ and $\sigma_i = 0$ for $i = k-3, \dots, k$ and where U and V were $k \times k$ random orthonormal matrices, computed as the $k \times k$ factors $Q(X)$ in the QR factorization of $k \times k$ random matrices X . Then we generated Gaussian Toeplitz matrices B , C and D such that $\|B\| \approx \|C\| \approx \|D\| \approx \|A_k\| \approx 1$ and defined the $n \times n$ matrix $A = \begin{pmatrix} A_k & B \\ C & D \end{pmatrix}$. For every dimension n , $n = 64, 128, 256, 512, 1024$ we run 1000 numerical tests where we solved the linear system $A\mathbf{x} = \mathbf{b}$ with Gaussian vector \mathbf{b} and output the maximum, minimum and average relative residual norms $\|A\mathbf{y} - \mathbf{b}\|/\|\mathbf{b}\|$ as well as the standard deviation. Figure 1 and Table D.1 show the norms of A^{-1} . They ranged from 2.2×10^1 to 3.8×10^6 in our tests.

At first we describe the results of our tests for the latter class of matrices A . As we expected GEPP has always output accurate solutions to the linear systems $A\mathbf{y} = \mathbf{b}$ in our tests (see Table D.2). GENP, however, was expected to fail for these systems, because the $(n/2) \times (n/2)$ leading principal block A_k of the matrix A was singular, having nullity $k - \text{rank}(A_k) = 4$. Indeed this caused poor performance of GENP in our tests, which have consistently output corrupted solutions, with relative residual norms ranging from 10^{-3} to 10^2 .

In view of Corollary 5.5 we expected to fix this deficiency by means of multiplication by Gaussian matrices, and indeed in all our tests we observed residual norms below 1.3×10^{-6} , and they decreased below 3.6×10^{-12} in a single step of iterative refinement (see Table D.3). Furthermore the tests showed the same power of preconditioning where we used the circulant multipliers of Examples 5.1 and 5.2 (see Tables D.4 and D.5). As can be expected the output accuracy of GENP with preprocessing has deteriorated a little versus GEPP in our tests, but the output residual norms were small enough to support application of the inexpensive iterative refinement, whose single step decreased the average relative residual norm below 10^{-11} for $n = 1024$ in the case of Gaussian multipliers and to about 10^{-13} for $n = 1024$ in the case of circulant multipliers of Examples 5.1 and 5.2. See further details in Figures 2 and 3 and Tables D.3–D.5.

We also applied similar tests to the $n \times n$ DFT matrix $A = \Omega$. The results were in very good accordance with our study in Section 5.4. Of course in this case the solution of a linear system $A\mathbf{x} = \mathbf{b}$ can be computed immediately as $\mathbf{x} = \frac{1}{n}\Omega^H\mathbf{b}$, but our goal was not outputting the solution but testing GENP with and without preprocessing. In these tests the norm $\|A^{-1}\|$ was fixed at $1/\sqrt{n}$, GEPP produced the solution within the relative residual norm between 10^{-15} and 10^{-16} , but GENP failed when we applied it to the inputs Ω with no preprocessing as well as to the inputs Ω preprocessed with random circulant multipliers of Examples 5.1 and 5.2. In these cases relative residual norms of the output approximations ranged between 10^{-2} and 10^4 . In contrast GENP applied to the inputs preprocessed with Gaussian multipliers produced quite reasonable approximations to the solution. Already after a single step of iterative refinement they have at least matched the level of GENP. Table D.6 displays these norms in some detail.

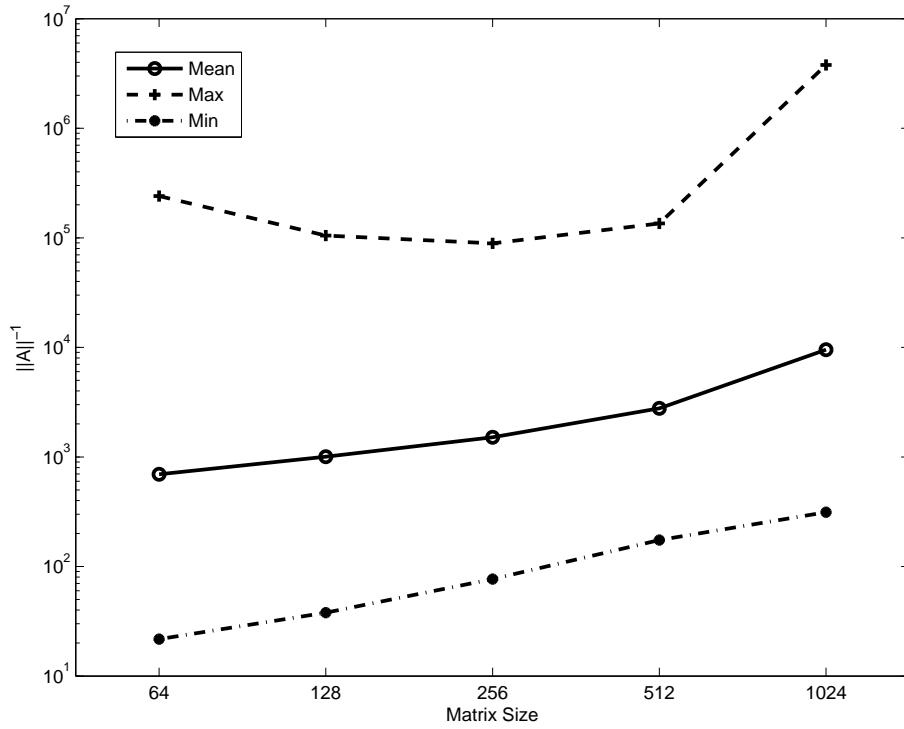


Figure 1: Norm of A^{-1}

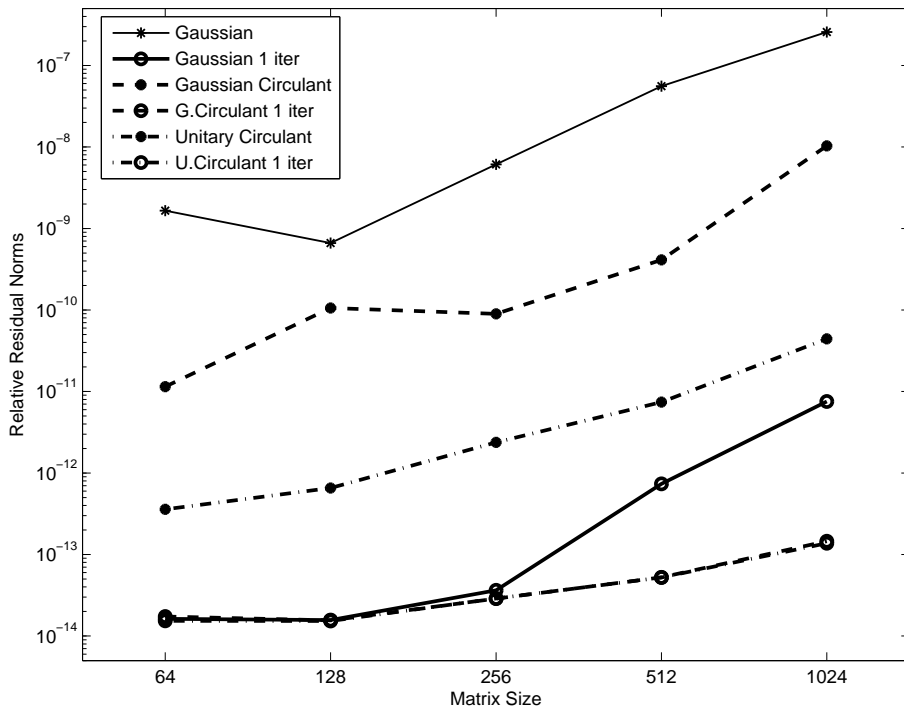


Figure 2: Average relative residual norms for GENP by using random multipliers. The two broken lines representing one iteration of circulant multipliers are overlapping at the bottom of the display

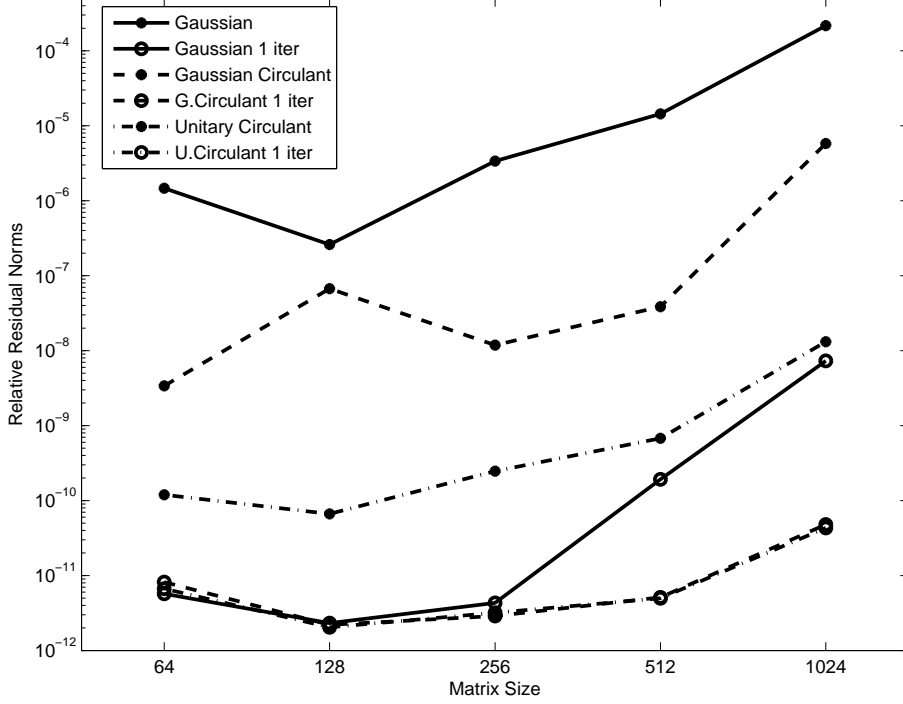


Figure 3: Maximum relative residual norms for GENP by using random multipliers. The two broken lines representing one iteration of circulant multipliers are overlapping at the bottom of the display

6.2 Approximation of the leading singular spaces and low-rank approximation of a matrix

In this section we present our test results for the approximation of the leading singular spaces of $n \times n$ matrices A that have numerical rank r and for the approximation of these matrices with matrices of rank r . For $n = 64, 128, 256, 512, 1024$ and $r = 8, 32$ we generated $n \times n$ random orthogonal matrices S and T and diagonal matrices $\Sigma = \text{diag}(\sigma_j)_{j=1}^n$ such that $\sigma_j = 1/j$, $j = 1, \dots, r$, $\sigma_j = 10^{-10}$, $j = r + 1, \dots, n$ (cf. [H02, Section 28.3]). Then we computed the input matrices $A = S_A \Sigma_A T_A^T$, for which $\|A\| = 1$ and $\kappa(A) = 10^{10}$. Furthermore we generated $n \times r$ random matrices H and computed the matrices $B_{r,A} = AH$, $Q_{r,A} = Q(B_{r,A})$, $S_{r,A}$, $T_{r,A}$, $Y_{r,A} = Q_{r,A}^T S_{r,A}$, and $Q_{r,A} Q_{r,A}^T A$. Figures 4–7 and Tables D.7–D.12 display the resulting data on the residual norms $\text{rn}^{(1)} = \|Q_{r,A} Y_{r,A} - S_{r,A}\|$ and $\text{rn}^{(2)} = \|A - Q_{r,A} Q_{r,A}^T A\|$, obtained in 1000 runs of our tests for every pair of n and r . In these figures and tables $\text{rn}^{(1)}$ denotes the residual norms of the approximations of the matrix bases for the leading singular spaces $\mathbb{S}_{r,A}$, whereas $\text{rn}^{(2)}$ denotes the residual norms of the approximations of the matrix A by the rank- r matrix $Q_{r,A} Q_{r,A}^T A$.

Figures 4 and 5 and Tables D.7–D.9 show the norm $\text{rn}^{(1)}$. The last column of each of the tables displays the ratio of the observed values $\text{rn}^{(1)}$ and its upper bound $\tilde{\Delta}_+ = \sqrt{2} \frac{\sigma_{r+1}(A)}{\sigma_r(A)} \|H\|_F \|(T_{r,A}^T H)^{-1}\|$ estimated up to the higher order terms (cf. Corollary 4.1). In our tests we had $\sigma_r(A) = 1/r$ and $\sigma_{r+1}(A) = 10^{-10}$. Table D.7 covers the case where we generated Gaussian multipliers H , whereas Tables D.8 and D.9 cover the cases where we generated random Toeplitz multipliers H as $n \times r$ submatrices of the $n \times n$ circulant matrices of Examples 5.1 and 5.2, respectively.

Figures 6 and 7 and Tables D.10–D.12 show similar results of our tests for the observed residual norms $\text{rn}^{(2)}$ and their ratios with their upper bounds $\tilde{\Delta}'_+ = \sigma_{r+1}(A) + 2\Delta_+ \|A\|$, estimated up to the higher order terms (cf. Corollary 4.2).

Tables D.13–D.14 show some auxiliary information. Namely, Table D.13 displays the data on the ratios $\|(T_{r,A}^T H)^{-1}\| / \|(H_{r,r})^{-1}\|$, where $H_{r,r}$ denotes the $r \times r$ leading submatrix of the matrix

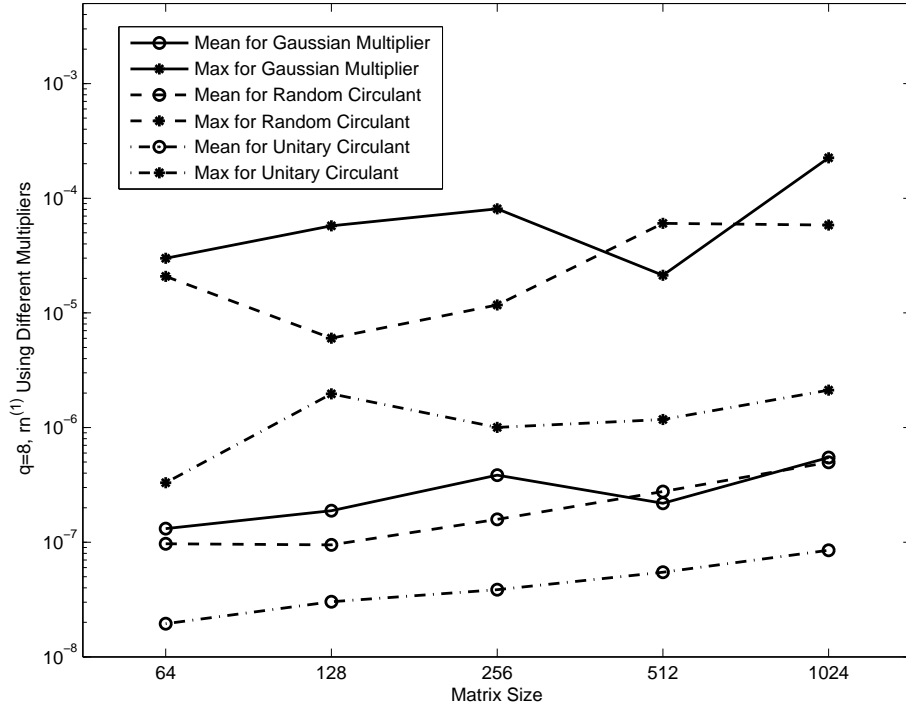


Figure 4: Residual norms $rn^{(1)}$ using different random multipliers, case $q=8$

H. Tables D.14 and D.15 display the average condition numbers of Gaussian $n \times n$ matrices and circulant $n \times n$ matrices C of Example 5.1, respectively.

The test results are in quite good accordance with our theoretical study of Gaussian multipliers and suggest that the power of random circulant and Toeplitz multipliers is similar to the power of Gaussian multipliers, as in the case of various random structured multipliers of [HMT11] and [M11].

7 Conclusions

It is well known that standard Gaussian random matrices (we refer to them as Gaussian for short) tend to be well conditioned, and this property motivated our application of Gaussian multipliers to advancing matrix computations. In particular we preprocess well conditioned nonsingular input matrices with Gaussian multipliers to support the application of GENP and block Gaussian elimination. Both of these algorithms readily fail in practical numerical computations without preprocessing, but we prove that with Gaussian multipliers the algorithms are expected to be locally safe, that is that the absolute values of the reciprocals of all pivot elements of GENP and the norms of the inverses of all pivot blocks of block Gaussian elimination are likely to be reasonably bounded. Our tests were in good accordance with that formal study. We generated matrices that were hard for GENP, but the problems were consistently avoided where we preprocessed the inputs with Gaussian multipliers. Moreover in our tests we observed similar results even with random circulant multipliers. Their choice accelerates multiplication significantly, and is particularly efficient where the input matrix has Toeplitz structure. That choice also limits randomization to n random parameters for an $n \times n$ input.

Our analysis can be naturally extended to the problem of rank- r approximation of an $m \times n$ matrix A that has a numerical rank r . We recalled that with a probability close to 1 the column sets of the matrix $Q(AH)$ for an $n \times r$ Gaussian matrix H approximates some orthogonal basis for the left leading singular space $\mathbb{S}_{r,A}$ associated with the r largest singular values of an $m \times n$ matrix A . Having such approximate basis available we can readily approximate the matrix A by a

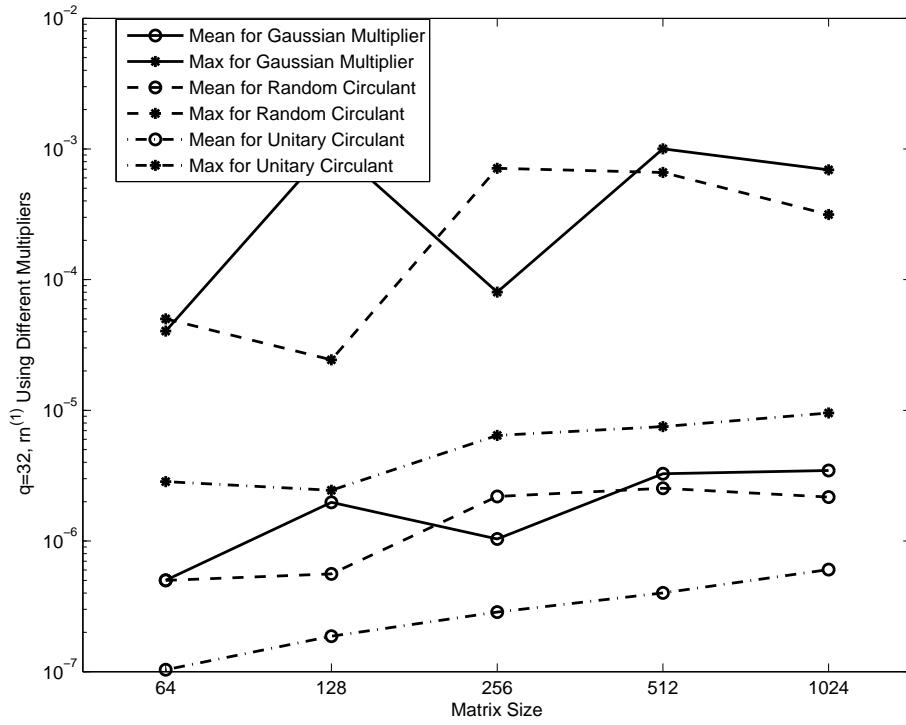


Figure 5: Residual norms $rn^{(1)}$ using different random multipliers, case $q=32$

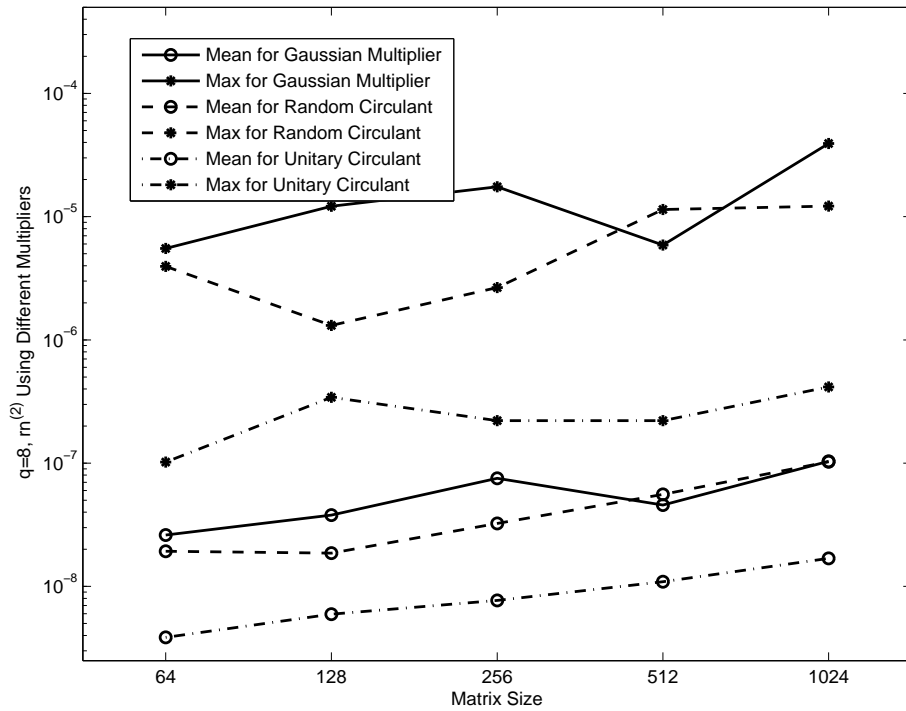


Figure 6: Residual norms $rn^{(2)}$ using different random multipliers, case $q=8$

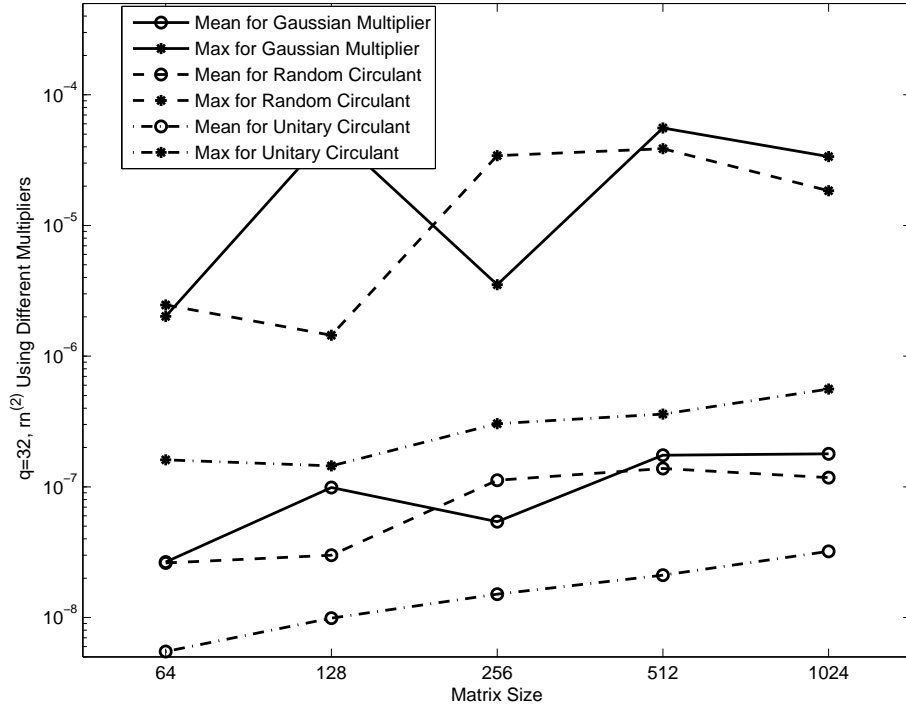


Figure 7: Residual norms $rn^{(2)}$ using different random multipliers, case $q=32$

matrix of rank r . This is an efficient well developed algorithm (see [HMT11]), but we proved that this algorithm was expected to produce reasonable rank- r approximation with Gaussian multipliers even without customary oversampling, recommended in [HMT11]. Then again in our tests the latter techniques were efficient even where instead of Gaussian multipliers we applied random Toeplitz multipliers, thus significantly accelerating the multiplication stage and limiting randomization to n parameters for an $n \times n$ input.

Formal proof of the power of random structured SRFT multipliers with substantial oversampling is known for low-rank approximation [HMT11, Section 11], and can be readily extended to the case where the products of random unitary circulant multipliers and random rectangular permutation matrices are applied instead of the SRFT matrices (see Section 5.4). Proving the power of random circulant multipliers for GENP can be a natural research challenge, but we show a specific input for which such multipliers are expected to fail (see Example 5.4). Would the same multipliers or their proper randomized variations be expected to avoid failure on the average input? Or in the case of Toeplitz and Toeplitz-like input matrices? Positive answer would formally support safe numerical application of the MBA celebrated superfast algorithm preconditioned with such random multipliers to solving Toeplitz and Toeplitz-like linear systems of equations (cf. [PQZ11]). Another natural research challenge is the combination of randomized matrix multiplication with randomized additive preprocessing and augmentation studied in [PQ10], [PQ12], [PQZC], and [PQZ13].

Acknowledgements: Our research has been supported by NSF Grant CCF-1116736 and PSC CUNY Awards 64512-0042 and 65792-0043. We are also grateful to the reviewers and Mr. Jesse Wolf for thoughtful helpful comments.

References

- [A94] O. Axelsson, *Iterative Solution Methods*, Cambridge Univ. Press, England, 1994.

- [B02] M. Benzi, Preconditioning Techniques for Large Linear Systems: a Survey, *J. of Computational Physics*, **182**, 418–477, 2002.
- [BBD12] D. Becker, M. Baboulin, J. Dongarra, Reducing the amount of pivoting in symmetric indefinite systems, *Proceedings of the 9th International Conference on Parallel Processing and Applied Mathematics, PPAM 2011, Lecture Notes in Computer Science*, **7203**, 133–142, Springer-Verlag (2012). Also *INRIA Research Report 7621 (05/2011)*, *University of Tennessee Technical Report ICL-UT-11-06*.
- [BP94] D. Bini, V. Y. Pan, *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms*, Birkhäuser, Boston, 1994.
- [CD05] Z. Chen, J. J. Dongarra, Condition Numbers of Gaussian Random Matrices, *SIAM J. on Matrix Analysis and Applications*, **27**, 603–620, 2005.
- [CGS07] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, J. Zhu, A Superfast Algorithm for Toeplitz Systems of Linear Equations, *SIAM J. on Matrix Analysis and Applications*, **29**, **4**, 1247–1266, 2007.
- [CPW74] R. E. Cline, R. J. Plemmons, and G. Worm, Generalized Inverses of Certain Toeplitz Matrices, *Linear Algebra and Its Applications*, **8**, 25–33, 1974.
- [D88] J. Demmel, The Probability That a Numerical Analysis Problem Is Difficult, *Math. of Computation*, **50**, 449–480, 1988.
- [DL78] R. A. Demillo, R. J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Processing Letters*, **7**, **4**, 193–195, 1978.
- [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.
- [G97] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
- [G98] M. Gu, Stable and Efficient Algorithms for Structured Systems of Linear Equations, *SIAM J. on Matrix Analysis and Applications*, **19**, 279–306, 1998.
- [GE96] M. Gu, S. C. Eisenstat, Efficient algorithms for computing a strong rank-revealing QR factorization, *SIAM Journal on Scientific Computing*, **17**, 848–869, 1996.
- [GKO95] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gaussian Elimination with Partial Pivoting for Matrices with Displacement Structure, *Mathematics of Computation*, **64**, 1557–1576, 1995.
- [GL96] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1996 (third addition).
- [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.

- [HP92] Y. P. Hong, C.-T. Pan, The rank revealing QR decomposition and SVD, *Math. of Computation*, **58**, 213–232, 1992.
- [KL94] C. S. Kenney, A. J. Laub, Small-Sample Statistical Condition Estimates for General Matrix Functions, *SIAM J. on Scientific and Statistical Computing*, **15**, 36–61, 1994.
- [M11] M. W. Mahoney, Randomized Algorithms for Matrices and Data, *Foundations and Trends in Machine Learning*, NOW Publishers, **3, 2**, 2011. (Abridged version in: *Advances in Machine Learning and Data Mining for Astronomy*, edited by M. J. Way, et al., pp. 647-672, 2012.)
- [MRT05] P. G. Martinsson, V. Rokhlin, M. Tygert, A Fast Algorithm for the Inversion of Toeplitz Matrices, *Comput. Math. Appl.*, **50**, 741–752, 2005.
- [MRT11] P.-G. Martinsson, V. Rokhlin, M. Tygert, A randomized algorithm for the decomposition of matrices, *Applied and Computational Harmonic Analysis*, **30**, 47–68, 2011.
- [P90] V. Y. Pan, On Computations with Dense Structured Matrices, *Math. of Computation*, **55**, **191**, 179–190, 1990. Also in *Proc. Intern. Symposium on Symbolic and Algebraic Computation (ISSAC'89)*, 34–42, ACM Press, New York, 1989.
- [P00a] C.-T. Pan, On the Existence and Computation of Rank-revealing LU Factorization, *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, Transformations of Matrix Structures Work Again, *Linear Algebra and Its Applications*, accepted. Available at arxiv:1311.3729[math.NA]
- [Pb] V. Y. Pan, Fast Approximation Algorithms for Computations with Cauchy Matrices, Polynomials and Rational Functions, *Proc. of the Ninth International Computer Science Symposium in Russia (CSR'2014)*, (E. A. Hirsch et al., editors), Moscow, Russia, June 2014, *Lecture Notes in Computer Science (LNCS)*, **8476**, pp. 287-300 Springer International Publishing, Switzerland, 2014. Also Tech. Report TR 2014005, *PhD Program in Comp. Sci., Graduate Center, CUNY*, 2014.
available at <http://www.cs.gc.cuny.edu/tr/techreport.php?id=469>
- [PGMQ] V. Y. Pan, D. Grady, B. Murphy, G. Qian, R. E. Rosholt, A. Ruslanov, Schur Aggregation for Linear Systems and Determinants, *Theoretical Computer Science, Special Issue on Symbolic-Numerical Algorithms* (D. A. Bini, V. Y. Pan, and J. Verschelde editors), **409, 2**, 255–268, 2008.
- [PIM10] V. Y. Pan, D. Ivolgin, B. Murphy, R. E. Rosholt, Y. Tang, X. Yan, Additive Preconditioning for Matrix Computations, *Linear Algebra and Its Applications*, **432**, 1070–1089, 2010.
- [PP95] D. S. Parker and B. Pierce, The randomizing FFT: An alternative to pivoting in Gaussian elimination, Tech. Report CSD 950037, *Computer Science Dept., Univ. California at Los Angeles*, 1995.
- [PQ10] V. Y. Pan, G. Qian, Randomized Preprocessing of Homogeneous Linear Systems of Equations, *Linear Algebra and Its Applications*, **432**, 3272–3318, 2010.
- [PQ12] V. Y. Pan, G. Qian, Solving Linear Systems of Equations with Randomization, Augmentation and Aggregation, *Linear Algebra and Its Applications*, **437**, 2851–1876, 2012.

- [PQa] V. Y. Pan, G. Qian, Estimating the Norms of Circulant and Toeplitz Random Matrices and Their Inverses, *Linear Algebra and Its Applications*, accepted. Available at arxiv:1311.3730[math.NA]
- [PQYa] V. Y. Pan, G. Qian, X. Yan, Supporting GENP with random multipliers, Tech. Report TR 2013016, *PhD Program in Comp. Sci., Graduate Center, CUNY*, 2013, available at <http://www.cs.gc.cuny.edu/tr/techreport.php?id=463>
- [PQZ11] V. Y. Pan, G. Qian, A. Zheng, Randomized Preconditioning of the MBA Algorithm, in *Proc. International Symp. on Symbolic and Algebraic Computation (ISSAC'2011)*, San Jose, California, June 2011 (edited by Anton Leykin), 281–288, ACM Press, New York (2011).
- [PQZ13] V. Y. Pan, G. Qian, A. Zheng, Randomized Preprocessing versus Pivoting, *Linear Algebra and Its Applications*, **438**, 4, 1883–1899, 2013.
- [PQZa] V. Y. Pan, G. Qian, A. Zheng, Randomized Matrix Computations, Tech. Report TR 2012009, *PhD Program in Comp. Sci., Graduate Center, CUNY* Available at <http://www.cs.gc.cuny.edu/tr/techreport.php?id=438> and <http://arxiv.org/abs/1210.7476>
- [PQZC] V. Y. Pan, G. Qian, A. Zheng, Z. Chen, Matrix Computations and Polynomial Root-finding with Preprocessing, *Linear Algebra and Its Applications*, **434**, 854–879, 2011.
- [S80] J. T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of ACM*, **27**, 4, 701–717, 1980.
- [S91] S. J. Szarek, Condition numbers of random matrices, *Journal of Complexity*, **7**, 2, 131–149, 1991.
- [S95] J.-G. Sun, On Perturbation Bounds for QR Factorization, *Linear Algebra and Its Applications*, **215**, 95–111, 1995.
- [S98] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decompositions*, SIAM, 1998.
- [S03] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, 2003.
- [SST06] A. Sankar, D. Spielman, S.-H. Teng, Smoothed Analysis of the Condition Numbers and Growth Factors of Matrices, *SIAM J. on Matrix Analysis*, **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.
- [Z79] R. E. Zippel, Probabilistic Algorithms for Sparse Polynomials, *Proceedings of EURO-SAM'79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.

Appendix

A On the algebraic variety of low-rank matrices

The following simple result (not used in this paper) shows that the $m \times n$ matrices of a rank ρ form an algebraic variety of the dimension $d_\rho = (m + n - \rho)\rho$ in the space $\mathbb{R}^{m \times n}$, and clearly $d_\rho < mn$ for $\rho < \min\{m, n\}$.

Fact A.1. *The set \mathbb{A} of $m \times n$ matrices of rank ρ is an algebraic variety of dimension $(m + n - \rho)\rho$.*

Proof. Let A be an $m \times n$ matrix of a rank ρ with a nonsingular leading $\rho \times \rho$ block B and write $A = \begin{pmatrix} B & C \\ D & E \end{pmatrix}$. Then the $(m - \rho) \times (n - \rho)$ Schur complement $E - DB^{-1}C$ must vanish, which imposes $(m - \rho)(n - \rho)$ algebraic equations on the entries of the matrix A . Similar argument can be applied where any $\rho \times \rho$ submatrix of the matrix A (among $\binom{m}{\rho} \binom{n}{\rho}$ such submatrices) is nonsingular. Therefore $\dim \mathbb{A} = mn - (m - \rho)(n - \rho) = (m + n - \rho)\rho$. \square

B Some error norm bounds for low-rank approximation

The paper [HMT11] proposed using Algorithm 4.1 with the positive oversampling integer parameter p (see [HMT11, Algorithm 4.1 and Theorems 10.1 and 10.6]). This choice relied on the following bounds of [HMT11, Theorems 10.5 and 10.6] on the expected value $\mathbb{E}(\|A - P_{AH}A\|)$ of the output error norm of the algorithm for $P_{AH} = QQ^T$,

$$\mathbb{E}(\|A - P_{AH}A\|_F) \leq \left(1 + \frac{r}{p-1} \sum_{j>r} \sigma_j(A)^2\right)^{1/2}, \quad (\text{B.1})$$

$$\mathbb{E}(\|A - P_{AH}A\|) \leq \left(1 + \frac{r}{p-1} \sigma_{r+1}(A)^2\right)^{1/2} + \frac{e\sqrt{r+p}}{p} \sum_{j>r} \sigma_j(A)^2)^{1/2}. \quad (\text{B.2})$$

Here is a simplified variant of the latter estimate from [HMT11, equation (1.8)],

$$\mathbb{E}(\|A - P_{AH}A\|) \leq \left(1 + \frac{4\sqrt{r+p}}{p-1} \sqrt{\min\{m, n\}}\right) \sigma_{r+1}(A). \quad (\text{B.3})$$

Quite typically the values $\sigma_j(A)$ for $j > r$ are not known, but one can adapt the parameter l by using a posteriori error estimation. One can simplify this estimation by recalling from [HMT11, equation (4.3)] that

$$\|A - P_{AH}A\| \leq 10\sqrt{2/\pi} \max_{j=1, \dots, r} (A - P_{AH}A) \mathbf{g}_j \quad (\text{B.4})$$

with a probability at least $1 - 10^{-r}$. Here \mathbf{g}_j is the j th column of $n \times r$ Gaussian matrix, that is $\mathbf{g}_1, \dots, \mathbf{g}_r$ are r independent Gaussian vectors of length n , and r is an integer parameter (see our Remark 4.2 on improving this approximation). Here is an alternative simplified expression from [HMT11, equation (1.9)],

$$\text{Probability}(\|A - P_{AH}A\| \leq (1 + 9\sqrt{r+p}\sqrt{\min\{m, n\}})\sigma_{r+1}(A)) \geq 1 - 3/p^p \quad (\text{B.5})$$

under some mild assumptions on the positive oversampling integer p . The above bounds show that low-rank approximations of high quality can be obtained by using a reasonably small oversampling integer parameter p , say $p = 20$, but they do not apply where $p \leq 1$. Our analysis of the basic algorithms relies on Corollary 5.4 and provides some reasonable formal support even where $p = 0$.

C Uniform random sampling and nonsingularity of random matrices

Uniform random sampling of elements from a finite set Δ is their selection from this set at random, independently of each other and under the uniform probability distribution on the set Δ .

The total degree of a multivariate monomial is the sum of its degrees in all its variables. The total degree of a polynomial is the maximal total degree of its monomials.

Lemma C.1. [DL78], [S80], [Z79]. *For a set Δ of a cardinality $|\Delta|$ in any fixed ring let a polynomial in m variables have a total degree d and let it not vanish identically on the set Δ^m . Then the polynomial vanishes in at most $d|\Delta|^{m-1}$ points of this set.*

Theorem C.1. *Under the assumptions of Lemma C.1 let the values of the variables of the polynomial be randomly and uniformly sampled from a finite set Δ . Then the polynomial vanishes with a probability at most $\frac{d}{|\Delta|}$.*

Corollary C.1. *Let the entries of a general or Toeplitz $m \times n$ matrix have been randomly and uniformly sampled from a finite set Δ of cardinality $|\Delta|$ (in any fixed ring). Let $l = \min\{m, n\}$. Then (a) every $k \times k$ submatrix M for $k \leq l$ is nonsingular with a probability at least $1 - \frac{k}{|\Delta|}$ and (b) is strongly nonsingular with a probability at least $1 - \sum_{i=1}^k \frac{i}{|\Delta|} = 1 - \frac{(k+1)k}{2|\Delta|}$.*

Proof. The claimed properties of nonsingularity and nonvanishing hold for generic matrices. The singularity of a $k \times k$ matrix means that its determinant vanishes, but the determinant is a polynomial of total degree k in the entries. Therefore Theorem C.1 implies parts (a) and consequently (b). Part (c) follows because a fixed entry of the inverse vanishes if and only if the respective entry of the adjoint vanishes, but up to the sign the latter entry is the determinant of a $(k-1) \times (k-1)$ submatrix of the input matrix M , and so it is a polynomial of degree $k-1$ in its entries. \square

D Tables

Table D.1: The norms $\|A\|^{-1}$ of the input matrices A

dimension	mean	max	min	std
64	6.95×10^2	2.41×10^5	2.18×10^1	7.87×10^3
128	1.00×10^3	1.05×10^5	3.78×10^1	5.81×10^3
256	1.51×10^3	8.90×10^4	7.68×10^1	6.06×10^3
512	2.78×10^3	1.35×10^5	1.74×10^2	8.64×10^3
1024	9.54×10^3	3.79×10^6	3.13×10^2	1.21×10^5

Table D.2: Relative residual norms of GEPP

dimension	mean	max	min	std
64	4.91×10^{-14}	2.06×10^{-11}	1.75×10^{-15}	6.64×10^{-13}
128	6.86×10^{-14}	7.58×10^{-12}	3.97×10^{-15}	3.02×10^{-13}
256	2.00×10^{-13}	1.95×10^{-11}	1.05×10^{-14}	8.93×10^{-13}
512	6.08×10^{-13}	5.76×10^{-11}	3.55×10^{-14}	2.65×10^{-12}
1024	2.67×10^{-12}	8.02×10^{-10}	1.13×10^{-13}	2.65×10^{-11}

Table D.3: Relative residual norms: GENP with Gaussian multipliers

dimension	iterations	mean	max	min	std
64	0	1.66×10^{-9}	1.47×10^{-6}	4.47×10^{-14}	4.67×10^{-8}
64	1	1.63×10^{-14}	5.71×10^{-12}	5.57×10^{-16}	1.91×10^{-13}
128	0	6.62×10^{-10}	2.61×10^{-7}	3.98×10^{-13}	8.66×10^{-9}
128	1	1.57×10^{-14}	2.31×10^{-12}	9.49×10^{-16}	8.23×10^{-14}
256	0	6.13×10^{-9}	3.39×10^{-6}	2.47×10^{-12}	1.15×10^{-7}
256	1	3.64×10^{-14}	4.32×10^{-12}	1.91×10^{-15}	2.17×10^{-13}
512	0	5.57×10^{-8}	1.44×10^{-5}	1.29×10^{-11}	7.59×10^{-7}
512	1	7.36×10^{-13}	1.92×10^{-10}	3.32×10^{-15}	1.07×10^{-11}
1024	0	2.58×10^{-7}	2.17×10^{-4}	4.66×10^{-11}	6.86×10^{-6}
1024	1	7.53×10^{-12}	7.31×10^{-9}	6.75×10^{-15}	2.31×10^{-10}

Table D.4: Relative residual norms: GENP with real circulant Gaussian multipliers of Example 5.1

dimension	iterations	mean	max	min	std
64	0	1.15×10^{-11}	3.39×10^{-9}	2.15×10^{-14}	1.18×10^{-10}
64	1	1.73×10^{-14}	8.18×10^{-12}	5.95×10^{-16}	2.62×10^{-13}
128	0	1.06×10^{-10}	6.71×10^{-8}	1.73×10^{-13}	2.15×10^{-9}
128	1	1.56×10^{-14}	2.20×10^{-12}	8.96×10^{-16}	7.91×10^{-14}
256	0	8.97×10^{-11}	1.19×10^{-8}	6.23×10^{-13}	4.85×10^{-10}
256	1	2.88×10^{-14}	2.89×10^{-12}	1.89×10^{-15}	1.32×10^{-13}
512	0	4.12×10^{-10}	3.85×10^{-8}	2.37×10^{-12}	2.27×10^{-9}
512	1	5.24×10^{-14}	5.12×10^{-12}	2.95×10^{-15}	2.32×10^{-13}
1024	0	1.03×10^{-8}	5.80×10^{-6}	1.09×10^{-11}	1.93×10^{-7}
1024	1	1.46×10^{-13}	4.80×10^{-11}	6.94×10^{-15}	1.60×10^{-12}

Table D.5: Relative residual norms: GENP with unitary circulant multipliers of Example 5.2

dimension	iterations	mean	max	min	std
64	0	3.59×10^{-13}	1.19×10^{-10}	6.14×10^{-15}	3.95×10^{-12}
64	1	1.53×10^{-14}	6.69×10^{-12}	5.74×10^{-16}	2.14×10^{-13}
128	0	6.54×10^{-13}	6.64×10^{-11}	2.68×10^{-14}	2.67×10^{-12}
128	1	1.53×10^{-14}	2.04×10^{-12}	9.31×10^{-16}	7.45×10^{-14}
256	0	2.37×10^{-12}	2.47×10^{-10}	9.41×10^{-14}	1.06×10^{-11}
256	1	2.88×10^{-14}	3.18×10^{-12}	1.83×10^{-15}	1.36×10^{-13}
512	0	7.42×10^{-12}	6.77×10^{-10}	3.35×10^{-13}	3.04×10^{-11}
512	1	5.22×10^{-14}	4.97×10^{-12}	3.19×10^{-15}	2.29×10^{-13}
1024	0	4.43×10^{-11}	1.31×10^{-8}	1.28×10^{-12}	4.36×10^{-10}
1024	1	1.37×10^{-13}	4.33×10^{-11}	6.67×10^{-15}	1.41×10^{-12}

Table D.6: Relative residual norms: GENP with Gaussian multipliers

dimension	iterations	mean	max	min	std
64	0	3.41×10^{-13}	1.84×10^{-11}	1.73×10^{-14}	1.84×10^{-12}
64	1	5.10×10^{-16}	8.30×10^{-16}	4.02×10^{-16}	6.86×10^{-17}
128	0	5.48×10^{-13}	7.21×10^{-12}	6.02×10^{-14}	9.05×10^{-13}
128	1	7.41×10^{-16}	9.62×10^{-16}	6.11×10^{-16}	6.82×10^{-17}
256	0	2.26×10^{-12}	4.23×10^{-11}	2.83×10^{-13}	4.92×10^{-12}
256	1	1.05×10^{-15}	1.26×10^{-15}	9.14×10^{-16}	6.76×10^{-17}
512	0	1.11×10^{-11}	6.23×10^{-10}	6.72×10^{-13}	6.22×10^{-11}
512	1	1.50×10^{-15}	1.69×10^{-15}	1.33×10^{-15}	6.82×10^{-17}
1024	0	7.57×10^{-10}	7.25×10^{-8}	1.89×10^{-12}	7.25×10^{-9}
1024	1	2.13×10^{-15}	2.29×10^{-15}	1.96×10^{-15}	7.15×10^{-17}

Table D.7: Residual norms $\text{rn}^{(1)}$ and the mean ratios of them and their upper bounds $\tilde{\delta}_+$, in the case of using Gaussian multipliers

q	n	mean	max	mean of ratio $\text{rn}^{(1)}/\tilde{\Delta}_+$
8	64	1.31×10^{-7}	3.00×10^{-5}	1.48×10^{-1}
8	128	1.88×10^{-7}	5.75×10^{-5}	1.52×10^{-1}
8	256	3.84×10^{-7}	8.09×10^{-5}	1.54×10^{-1}
8	512	2.18×10^{-7}	2.13×10^{-5}	1.57×10^{-1}
8	1024	5.47×10^{-7}	2.25×10^{-4}	1.58×10^{-1}
32	64	5.00×10^{-7}	4.05×10^{-5}	5.23×10^{-2}
32	128	1.98×10^{-6}	1.08×10^{-3}	6.44×10^{-2}
32	256	1.04×10^{-6}	8.03×10^{-5}	6.90×10^{-2}
32	512	3.27×10^{-6}	1.00×10^{-3}	7.11×10^{-2}
32	1024	3.46×10^{-6}	6.92×10^{-4}	7.30×10^{-2}

Table D.8: Residual norms $\text{rn}^{(1)}$ and the mean ratios of them and their upper bounds $\tilde{\delta}_+$, in the case of using Toeplitz random multipliers and Example 5.1

q	n	mean	max	mean of ratio $\text{rn}^{(1)}/\tilde{\Delta}_+$
8	64	9.70×10^{-8}	2.01×10^{-5}	1.50×10^{-1}
8	128	9.48×10^{-8}	6.03×10^{-6}	1.54×10^{-1}
8	256	1.58×10^{-7}	1.17×10^{-5}	1.57×10^{-1}
8	512	2.77×10^{-7}	6.04×10^{-5}	1.57×10^{-1}
8	1024	4.97×10^{-7}	5.83×10^{-5}	1.58×10^{-1}
32	64	4.99×10^{-7}	5.01×10^{-5}	5.73×10^{-2}
32	128	5.61×10^{-7}	2.43×10^{-5}	6.54×10^{-2}
32	256	2.19×10^{-6}	7.11×10^{-4}	6.98×10^{-2}
32	512	2.53×10^{-6}	6.62×10^{-4}	7.20×10^{-2}
32	1024	2.17×10^{-6}	3.15×10^{-4}	7.25×10^{-2}

Table D.9: Residual norms $\text{rn}^{(1)}$ and the mean ratios of them and their upper bounds $\tilde{\delta}_+$, in the case of using Toeplitz random multipliers and Example 5.2

q	n	mean	max	mean of ratio $\text{rn}^{(1)}/\tilde{\Delta}_+$
8	64	1.94×10^{-8}	3.30×10^{-7}	1.59×10^{-1}
8	128	3.03×10^{-8}	1.97×10^{-6}	1.59×10^{-1}
8	256	3.85×10^{-8}	1.00×10^{-6}	1.59×10^{-1}
8	512	5.47×10^{-8}	1.18×10^{-6}	1.59×10^{-1}
8	1024	8.51×10^{-8}	2.12×10^{-6}	1.59×10^{-1}
32	64	1.03×10^{-7}	2.84×10^{-6}	7.37×10^{-2}
32	128	1.87×10^{-7}	2.44×10^{-6}	7.39×10^{-2}
32	256	2.86×10^{-7}	6.43×10^{-6}	7.39×10^{-2}
32	512	4.00×10^{-7}	7.50×10^{-6}	7.38×10^{-2}
32	1024	6.05×10^{-7}	9.54×10^{-6}	7.43×10^{-2}

Table D.10: Residual norms $\text{rn}^{(2)}$ and the mean ratio of them and their upper bounds, in the case of using Gaussian random multipliers

q	n	mean	max	mean of ratio $\text{rn}^{(2)}/\tilde{\Delta}_+$
8	64	2.61×10^{-8}	5.52×10^{-6}	1.46×10^{-2}
8	128	3.79×10^{-8}	1.21×10^{-5}	1.52×10^{-2}
8	256	7.54×10^{-8}	1.75×10^{-5}	1.54×10^{-2}
8	512	4.57×10^{-8}	5.88×10^{-6}	1.55×10^{-2}
8	1024	1.03×10^{-7}	3.93×10^{-5}	1.56×10^{-2}
32	64	2.66×10^{-8}	2.02×10^{-6}	1.38×10^{-3}
32	128	9.87×10^{-8}	5.22×10^{-5}	1.70×10^{-3}
32	256	5.41×10^{-8}	3.52×10^{-6}	1.83×10^{-3}
32	512	1.75×10^{-7}	5.57×10^{-5}	1.89×10^{-3}
32	1024	1.79×10^{-7}	3.36×10^{-5}	1.92×10^{-3}

Table D.11: Residual norms $\text{rn}^{(2)}$ and the mean ratio of them and their upper bounds, in the case of using Toeplitz random multipliers and Example 5.1

q	n	mean	max	mean of ratio $\text{rn}^{(2)}/\tilde{\Delta}_+$
8	64	1.93×10^{-8}	3.95×10^{-6}	1.48×10^{-2}
8	128	1.86×10^{-8}	1.31×10^{-6}	1.52×10^{-2}
8	256	3.24×10^{-8}	2.66×10^{-6}	1.55×10^{-2}
8	512	5.58×10^{-8}	1.14×10^{-5}	1.55×10^{-2}
8	1024	1.03×10^{-7}	1.22×10^{-5}	1.56×10^{-2}
32	64	2.62×10^{-8}	2.47×10^{-6}	1.52×10^{-3}
32	128	3.00×10^{-8}	1.44×10^{-6}	1.73×10^{-3}
32	256	1.12×10^{-7}	3.42×10^{-5}	1.84×10^{-3}
32	512	1.38×10^{-7}	3.87×10^{-5}	1.30×10^{-3}
32	1024	1.18×10^{-7}	1.84×10^{-5}	1.92×10^{-3}

Table D.12: Residual norms $\bar{r}n^{(2)}$ and the mean ratio of them and their upper bounds, in the case of using Toeplitz random multipliers and Example 5.2

q	n	mean	max	mean of ratio $\bar{r}n^{(2)}/\bar{\Delta}_+$
8	64	3.86×10^{-9}	1.02×10^{-7}	1.56×10^{-2}
8	128	5.96×10^{-9}	3.42×10^{-7}	1.56×10^{-2}
8	256	7.70×10^{-9}	2.21×10^{-7}	1.56×10^{-2}
8	512	1.10×10^{-8}	2.21×10^{-7}	1.56×10^{-2}
8	1024	1.69×10^{-8}	4.15×10^{-7}	1.56×10^{-2}
32	64	5.49×10^{-9}	1.61×10^{-7}	1.95×10^{-3}
32	128	9.90×10^{-9}	1.45×10^{-7}	1.95×10^{-3}
32	256	1.51×10^{-8}	3.05×10^{-7}	1.95×10^{-3}
32	512	2.11×10^{-8}	3.60×10^{-7}	1.95×10^{-3}
32	1024	3.21×10^{-8}	5.61×10^{-7}	1.95×10^{-3}

Table D.13: Mean ratios of the norms of the inverses of the matrices $T_{r,A}^T H$ and $H_{r,r}$

n	$q = 8$	$q = 32$
64	7.93	7.19
128	5.74	2.12
256	1.26	3.67
512	5.72	1.44
1024	5.12	7.86

Table D.14: Condition numbers of Gaussian matrices

n	mean	max	min	std
64	1.83	2.47	1.40	0.16
128	1.51	1.77	1.30	0.08
256	1.34	1.55	1.20	0.05
512	1.23	1.38	1.11	0.03
1024	1.15	1.23	1.08	0.02

Table D.15: Condition numbers of circulant matrices of Example 5.1

n	mean	max	min	std
64	$4.65 \times 10^{+1}$	$6.66 \times 10^{+3}$	$4.11 \times 10^{+0}$	$2.91 \times 10^{+2}$
128	$4.91 \times 10^{+1}$	$3.93 \times 10^{+3}$	$5.92 \times 10^{+0}$	$1.65 \times 10^{+2}$
256	$1.40 \times 10^{+2}$	$7.31 \times 10^{+4}$	$8.50 \times 10^{+0}$	$2.32 \times 10^{+3}$
512	$1.01 \times 10^{+2}$	$1.06 \times 10^{+4}$	$1.33 \times 10^{+1}$	$4.69 \times 10^{+2}$
1024	$1.16 \times 10^{+2}$	$3.48 \times 10^{+3}$	$1.97 \times 10^{+1}$	$1.79 \times 10^{+2}$