2015

Random Multipliers Numerically Stabilize Gaussian and Block Gaussian Elimination: Proofs and an Extension to Low-rank Approximation

Victor Pan
Lehman College and Graduate Center of CUNY

Xiaodong Yan

Random Multipliers Numerically Stabilize Gaussian and Block Gaussian Elimination: Proofs and an Extension to Low-rank Approximation [*]

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

December 14, 2014

Abstract

We study two applications of standard Gaussian random multipliers. At first we prove that with a probability close to 1 such a multiplier is expected to numerically stabilize Gaussian elimination with no pivoting as well as block Gaussian elimination. Then, by extending our analysis, we prove that such a multiplier is also expected to support low-rank approximation of a matrix without customary oversampling. Our test results are in good accordance with this formal study. The results remain similar when we replace Gaussian multipliers with random circulant or Toeplitz multipliers, which involve fewer random parameters and enable faster multiplication. We formally support the observed efficiency of random structured multipliers applied to approximation, but not to elimination. Moreover, we prove that with a probability close to 1 Gaussian random circulant multipliers do not fix numerical instability of the elimination algorithms for a specific narrow class of well-conditioned inputs. We know of no such hard input classes for various alternative choices of random structured multipliers, but for none of such multipliers we have a formal proof of its efficiency for numerical Gaussian elimination.

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

Key Words: Random matrices, Random multipliers, Gaussian elimination, Pivoting, Block Gaussian elimination, Low-rank approximation, SRFT matrices, Random circulant matrices

---

1 Introduction

1.1 Overview

We call a standard Gaussian random matrix just *Gaussian* and apply Gaussian multipliers to

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

- approximate the leading singular spaces of an ill-conditioned matrix, associated with its largest singular values and

- approximate this matrix by a low-rank matrix.

Ample empirical evidence shows efficiency of all these applications [HMT11], [M11], [PQZ13], but formal proofs are only known in the case of approximation under the assumption of small oversampling, that is, a minor increase of the multiplier size [HMT11]. We provide formal support for randomized stabilization of Gaussian and block Gaussian elimination and for randomized approximation without oversampling.

Our test results are in good accordance with our analysis and also show similar power of random circulant multipliers or their leading (that is, northeastern) Toeplitz blocks for the elimination and approximation, respectively. Simple extension of the known estimates enables formal support for this empirical observation in the case of low-rank approximation, but not in the case of elimination. Moreover, we prove that with a high probability Gaussian circulant multipliers cannot fix numerical instability of the elimination algorithms for a specific narrow class of inputs. The issue remains open for the elimination preprocessed with other random structured multipliers.

In the next subsections we specify further the computational tasks and our results.

1.2 Numerical Gaussian elimination with no pivoting and block Gaussian elimination

Gaussian elimination, applied numerically, with rounding errors, can fail even in the case of a nonsingular well-conditioned input matrix unless this matrix is also positive definite, diagonally dominant, or totally positive. For example (see part (i) of Theorem 6.2), numerical Gaussian elimination fails when it is applied to the matrices of discrete Fourier transform of large sizes, even though they are unitary up to scaling. In practice the user avoids the problems by applying Gaussian elimination with partial pivoting, that is, appropriate row interchange, which has some limited formal and ample empirical support. Our alternative is Gaussian elimination with no pivoting. Hereafter we use the acronyms *GEPP* and *GENP*.

In Section 5.2 we prove that GENP is safe numerically with a probability close to 1 when the input matrix is nonsingular, well-conditioned and preprocessed with a Gaussian multiplier.

At the first glance our preprocessing may seem to be more costly than partial pivoting. The latter only involves order of n^2 comparisons for an $n \times n$ input matrix A, versus $2n^3 - n^2$ arithmetic operations for its multiplication by a Gaussian multiplier. Generation of n^2 independent Gaussian entries of the $n \times n$ multiplier is an additional charge, even though one can discount partly it because this preprocessing stage is independent of the input matrix. More careful comparison, however, shows that partial pivoting takes quite a heavy toll. It interrupts the stream of arithmetic operations with foreign operations of comparison, involves book-keeping, compromises data locality, and increases communication overhead and data dependence.

Choosing between GEPP and GENP with randomized preprocessing the user may also consider various other factors, some of them dynamic in time. For example, here is a relevant citation from [BCD14]: "The traditional metric for the efficiency of a numerical algorithm has been the number of arithmetic operations it performs. Technological trends have long been reducing the time to perform an arithmetic operation, so it is no longer the bottleneck in many algorithms; rather, communication, or moving data, is the bottleneck".

Our test results with Gaussian input matrices are in good accordance with our formal estimates (see Figures 1–5 and Tables D.1–D.6). In our tests the output accuracy of GENP with preprocessing was a little lower than in the case of the customary GEPP, as can be expected, but a single step of iterative refinement, performed at a dominated computational cost, has always fixed this discrepancy (see Figures 2 and 3 and Table D.3). The tests show similar results when we applied Gaussian circulant rather than general Gaussian multipliers, and in this case the cost of our preprocessing decreases dramatically (see more on that in Section 1.4).

Finally, all our study of GENP, including formal probabilistic support of its numerical performance, is immediately extended to *block Gaussian elimination* (see Section 3), whereas pivoting cannot amend this valuable algorithm, and so without our preprocessing, it is numerically unsafe to use it, unless an input matrix is nonsingular, well-conditioned, and positive definite, diagonally dominant, or totally positive.

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 $n \times l$ multiplier, for $l = r + p$ and a positive *oversampling integer p*. 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 that even for an $n \times r$ Gaussian multiplier, that is, *without oversampling*, the algorithms output rank-r approximations of the input matrix with a probability close to 1. Then again our test results are in good accordance with our formal estimates (see Figures 6 and 7 and Tables D.10–D.12).

The decrease of p to 0 should be theoretically interesting, although it has only minor practical promise, limited to the cases where the numerical rank r is small and the user knows it. Indeed, according to [HMT11, Section 4.2], "it is adequate to choose ... $p = 5$ or $p = 10$".

1.4 Computations with random structured multipliers

The SRFT $n \times l$ multipliers H (SRFT is the acronym for Subsample Random Fourier Transform) involve only n random parameters versus nl parameters of Gaussian multipliers and accelerate the computation of the product AH by a factor of $l/\log(l)$ versus $n \times l$ Gaussian multipliers H. It has been proved that they are expected to support rank-r approximation assuming oversampling integers $p = l - r$ of order $r \log(r)$, and empirically this has been observed for reasonably small constants p, usually being not more than 20 [HMT11, Section 11], [M11]. We readily extend the latter results to the case where $n \times l$ products of random $n \times n$ circulant and random $n \times l$ permutation matrices are used as multipliers instead of SRFT matrices (see Remark 7.3), and we observe such empirical behavior also in the cases when we preprocess GENP by applying random circulant multipliers (see Figures 2, 3, 6 and 7 and Tables D.4, D.5, D.11, and D.12). In the latter case we only need n random parameters versus n^2 for a Gaussian multiplier and accelerate Gaussian preprocessing for GENP by a factor of $n/\log(n)$.

This acceleration factor grows to $n^2/\log(n)$ if an input matrix is Toeplitz or Toeplitz-like and if we can apply the MBA celebrated algorithm, which is just recursive block Gaussian elimination adjusted to a Toeplitz-like input [P01, Chapter 5] and which is superfast, that is, runs in nearly linear arithmetic time. Numerical stability problems, however, are well known for this algorithm [B85], and fixing them was the central subject of the highly recognized papers [GKO95] and [G98]. Pivoting could not be applied here because it destroys Toeplitz structure, thus increasing the solution cost to cubic. So the authors first reduced the task to the case of Cauchy-like inputs by specializing the techniques of the transformation of matrix structures from [P90] (also see [P15]) and then applied a fast Cauchy-like variant of GEPP using quadratic arithmetic time.

3

Stabilization of the MBA superfast algorithm by means of random circulant multipliers would mean randomized acceleration by order of magnitude versus [GKO95] and [G98] because circulant multipliers preserve Toeplitz-like structure of an input matrix and thus keep the MBA algorithm superfast. Empirically, Gaussian circulant multipliers do stabilize GENP numerically, but no formal support is known for this observed behavior. Moreover we even prove that the application of GENP fails for a specific narrow class of matrices, and with probability near 1 their preprocessing with Gaussian circulant multipliers does not help (see Remark 7.3). We know of no such hard inputs for some variations of using Gaussian circulant multipliers for GENP and MBA, such as using products of random circulant and skew-circulant multipliers and using random circulant pre-multipliers and post-multipliers simultaneously. Both variations preserve Toeplitz-like input structure and allow superfast performance of the MBA algorithm, but we have no formal support for the efficiency of such multipliers.

1.5 Related works

Preconditioning of linear systems of equations is a classical subject [A94], [B02], [G97]. For 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 the most recent advance in this direction see [PZa]. On the early 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 some 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 7. We cite these and other related works throughout the paper and refer the reader to [PQZa, Section 11] for further bibliography. The estimates of our Corollary 4.1 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 [PSZa]. 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 show that GENP and block Gaussian elimination are numerically safe for a matrix whose all leading blocks are nonsingular and well conditioned. In Section 4 we estimate the impact of preprocessing with general nonrandom multipliers on these properties of the leading blocks. In Sections 5 and 6 we extend our analysis from Section 4 in order to cover the impact of Gaussian and random structured multipliers, respectively. In Section 7 we recall an algorithm from [HMT11] for low-rank approximation and prove that this randomized algorithm is expected to work even with no oversampling. In Section 8 we cover numerical tests (the contribution of the last two authors). Section 9 contains a brief summary. In Appendix A we recall the known probabilistic estimates for the error norms of randomized low-rank approximations. In Appendix B we estimate the probability that a random matrix has full rank under the uniform probability distribution. In Appendix C we estimate the perturbation errors of matrix inversion. In Appendix D we display tables with our test results, which are more detailed than the data given by the plots in Section 8. Some readers may be only interested in the part of our paper on GENP. They can skip Sections 7 and 8.2.

2 Some definitions

Except for using unitary circulant matrices in Sections 6 and 8.2, 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 only use the concept of the expected value in Theorem A.1, Corollary A.1, and Appendix A.)

Next we recall and extend some customary definitions of matrix computations [GL13], [S98].

$\mathbb{R}^{m \times n}$ is the class of real $m \times n$ matrices $A = (a_{i,j})_{i,j}^{m,n}$.

$\mathrm{diag}(B_1, \ldots, B_k) = \mathrm{diag}(B_j)_{j=1}^k$ is a $k \times k$ block diagonal matrix with the diagonal blocks $B_1, \ldots, B_k$. $(B_1)_1^k$, $(B_1 \mid \ldots \mid B_k)$, and $(B_1, \ldots, B_k)$ denote a $1 \times k$ block matrix with the blocks $B_1, \ldots, B_k$. In both cases the blocks B_j can be rectangular.

I_n is the $n \times n$ identity matrix. $O_{k,l}$ is the $k \times l$ matrix filled with zeros. We write I and O if the matrix size is defined by context. A^T is the transpose of a matrix A.

$A_{k,l}$ denotes its leading, that is, northwestern $k \times l$ block submatrix, and we also write $A^{(k)} = A_{k,k}$.

$||A|| = ||A||_2$ is the spectral norm of a matrix A. $||A||_F$ is its Frobenius norm.

A real matrix Q is *orthogonal* if $Q^T Q = I$ or $QQ^T = I$. $(Q, R) = (Q(A), R(A))$ 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 [GL13, Theorem 5.2.3].

A^+ denotes the Moore–Penrose pseudo-inverse of an $m \times n$ matrix A, and

$$A = S_A \Sigma_A T_A^T \tag{2.1}$$

denotes its SVD where $S_A^T S_A = S_A S_A^T = I_m$, $T_A^T T_A = T_A T_A^T = I_n$, $\Sigma_A = \mathrm{diag}(\sigma_j(A))_j$, and $\sigma_j = \sigma_j(A)$ is the jth largest singular value of A. If a matrix A has full column rank ρ, then

$$||A^+|| = 1/\sigma_\rho(A). \tag{2.2}$$

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, e.g., $A_{k,l}^T$ stands for $(A_{k,l})^T$.

$\kappa(A) = \frac{\sigma_1(A)}{\sigma_\rho(A)} = ||A||\, ||A^+||$ 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*. An $m \times n$ matrix A has a *numerical rank* $r = \mathrm{nrank}(A) \le \rho = \mathrm{rank}(A)$ if the ratios $\sigma_j(A)/||A||$ are small for $j > r$ but not for $j \le r$.

The following concepts cover all rectangular matrices, but we need them just in the case of square matrices, whose sets of leading blocks include the matrices themselves. A matrix is *strongly nonsingular* if all its leading blocks are nonsingular. Such a matrix is *strongly well-conditioned* if all its leading blocks are well-conditioned.

We recall further relevant definitions and basic results of matrix computations in the beginning of Section 7 and in the Appendix.

3 Block Gaussian elimination and GENP

For a nonsingular 2×2 block matrix $A = \begin{pmatrix} B & C \\ D & E \end{pmatrix}$ of size $n \times n$ with nonsingular $k \times k$ *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 factorizations,

$$A = \begin{pmatrix} I_k & O_{k,r} \\ DB^{-1} & I_r \end{pmatrix} \begin{pmatrix} B & O_{k,r} \\ O_{r,k} & S \end{pmatrix} \begin{pmatrix} I_k & B^{-1}C \\ O_{k,r} & I_r \end{pmatrix} \tag{3.1}$$

and

$$A^{-1} = \begin{pmatrix} I_k & -B^{-1}C \\ O_{k,r} & I_r \end{pmatrix} \begin{pmatrix} B^{-1} & O_{k,r} \\ O_{r,k} & S^{-1} \end{pmatrix} \begin{pmatrix} I_k & O_{k,r} \\ -DB^{-1} & I_r \end{pmatrix}. \tag{3.2}$$

We verify readily that S^{-1} is the $(n-k) \times (n-k)$ trailing (that is, southeastern) block of the inverse matrix A^{-1}, and so the Schur complement S is nonsingular since the matrix A is nonsingular.

Factorization (3.2) reduces the inversion of the matrix A to the inversion of the leading block B and its Schur complement S, and we can recursively reduce the task to the case of the leading blocks and Schur complements of decreasing sizes as long as the leading blocks are nonsingular. After sufficiently many recursive steps of this process of block Gaussian elimination, we only need to invert matrices of small sizes, and then we can stop the process and apply a selected black box inversion algorithm.

In $\lceil \log_2(n) \rceil$ recursive steps all pivot blocks and all other matrices involved into the resulting factorization turn into scalars, all matrix multiplications and inversions turn into scalar multiplications and divisions, and we arrive at a *complete recursive factorization* of the matrix A. If $k = 1$ at all recursive steps, then the complete recursive factorization (3.2) defines GENP and can be applied to computing the inverse A^{-1} or the solution $\mathbf{y} = A^{-1}\mathbf{b}$ to a linear system $A\mathbf{y} = \mathbf{b}$.

Actually, however, any complete recursive factorizations turns into GENP up to the order in which we consider its steps. This follows because at most $n - 1$ distinct Schur complements $S = S(A^{(k)}, A)$ for $k = 1, \ldots, n - 1$ are involved in all recursive block factorization processes for $n \times n$ matrices A, and so we arrive at the same Schur complement in a fixed position via GENP and via any other recursive block factorization (3.1). Hence we can interpret factorization step (3.1) as the block elimination of the first k columns of the matrix A, which produces the matrix $S = S(A^{(k)}, A)$. If the dimensions $d_1, \ldots, d_r$ and $\bar{d}_1, \ldots, \bar{d}_{\bar{r}}$ of the pivot blocks in two block elimination processes sum to the same integer k, that is, if $k = d_1 + \cdots + d_r = \bar{d}_1 + \cdots + \bar{d}_{\bar{r}}$, then both processes produce the same Schur complement $S = S(A^{(k)}, A)$. The following results extend this observation.

Theorem 3.1. *In the recursive block factorization process based on (3.1), every diagonal block of every 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 $0 < h < k \le n$ and $S(A^{(h)}, A^{(k)}) = (S(A^{(h)}, A))^{(h)}$.*

Corollary 3.1. *The recursive block factorization process based on equation (3.1) can be completed by involving no singular pivot blocks (and in particular no pivot elements vanish) if and only if the input matrix A is strongly nonsingular.*

Proof. Combine Theorem 3.1 with the equation $\det A = (\det B) \det S$, implied by (3.1). $\qquad\square$

The following theorem bounds the norms of all pivot blocks and their inverses and hence bounds the condition numbers of the blocks, that is, precisely the quantities responsible for safe numerical performance of block Gaussian elimination and GENP.

Theorem 3.2. *(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 $N_-N \ge ||A|| \, ||A^{-1}|| \ge 1$. Then the absolute values of all pivot elements of GENP and the norms of all pivot blocks of block Gaussian elimination do not exceed $N_+ = N + N_-N^2$, while 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 (3.1) is the southeastern block of the inverse A^{-1} and obtain $||B|| \le N$, $||B^{-1}|| \le N_-$, and $||S^{-1}|| \le ||A^{-1}|| \le N_-$. Moreover $||S|| \le N + N_-N^2$, due to (3.1). Now the claimed bound follows from Theorem 3.1. $\qquad\square$

Remark 3.1. *By virtue of Theorem 3.2 the norms of the inverses of all pivot blocks involved into a complete (and hence also into any incomplete) recursive factorization of a strongly nonsingular matrix A are at most N_-. We have a reasonable upper bound on N_- if the matrix A is strongly well-conditioned as well. Then in view of Theorem C.1 the inversion of all pivot blocks is numerically safe, and we say that GENP is locally safe for the matrix A.*

Remark 3.2. *In the recursive factorizations above only the factors of the leading blocks and the Schur complements can contribute to the magnification of any input perturbation. Namely at most $\lceil \log_2(n) \rceil$ such factors can contribute to the norm of each of the output triangular or block triangular factors L and U. This implies the moderately large worst case upper bound $(N_+ N_-)^{\log_2(n)}$ on their norms, which is overly pessimistic according to our tests.*

Remark 3.3. *Our study in this and the next two sections can be extended readily to the cases of GENP and block Gaussian elimination applied to rectangular and possibly rank deficient matrices and to under- and over-determined and possibly rank deficient linear systems of equations. Recursive factorization and elimination can be completed and are numerically safe when they are applied to any strongly nonsingular and strongly well-conditioned leading block of the input matrix, in particular to the input matrix itself if it is strongly nonsingular and strongly well-conditioned.*

4 Singular values of the matrix products (deterministic estimates) and GENP and block Gaussian elimination with preprocessing

Preprocessing $A \to FAH$ 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 a the product FAH, and similarly for the solution of a linear system of equations.

Fact 4.1. *Assume three nonsingular matrices F, A, and H and a vector $\mathbf{b}$. Then $A^{-1} = H(AH)^{-1}$, $A^{-1} = (FA)^{-1}F$, $A^{-1} = H(FAH)^{-1}F$. Moreover, if $A\mathbf{x} = \mathbf{b}$, then $AH\mathbf{y} = \mathbf{b}$, $FA\mathbf{x} = F\mathbf{b}$, and $FAH\mathbf{y} = F\mathbf{b}$, $\mathbf{x} = H\mathbf{y}$.*

Remark 3.1 motivates the choice of the multipliers F and H for which the matrix FAH is strongly nonsingular and strongly well-conditioned. This is likely to occur already if one of the multipliers F and H is the identity matrix and another one is a Gaussian random matrix. The studies of premultiplication by F and post-multiplication by H are similar, and so we only prove the latter claim in the case of post-multiplication. We complete our proof in Section 5.2. It involves the norms of the inverses of the matrices $(AH)_{k,k} = A_{k,n}H_{n,k}$ for $k = 1, \ldots, r$, which we estimate in this section assuming nonrandom multipliers H. We begin with two simple lemmas.

Lemma 4.1. *If S and T are square orthogonal matrices, then $\sigma_j(SA) = \sigma_j(AT) = \sigma_j(A)$ for all j.*

Lemma 4.2. *Suppose $\Sigma = \operatorname{diag}(\sigma_i)_{i=1}^n$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, and $H \in \mathbb{R}^{n \times r}$. Then*

$$\sigma_j(\Sigma H) \geq \sigma_j(H)\sigma_n \text{for all } j.$$

If also $\sigma_n > 0$, then

$$\operatorname{rank}(\Sigma H) = \operatorname{rank}(H).$$

We also need the following basic results (cf. [GL13, Corollary 8.6.3]).

Theorem 4.1. *If A_0 is a submatrix of a matrix A, then $\sigma_j(A) \geq \sigma_j(A_0)$ for all j.*

Theorem 4.2. *Suppose $r + l \leq n \leq m$, $l \geq 0$, $A \in \mathbb{R}^{m \times n}$, $\operatorname{rank}(A_{m,r}) = r$ and $\operatorname{rank}(A_{m,r+l}) = r + l$. Then $||A_{m,r}^+|| \leq ||A_{m,r+l}^+||$.*

The following theorem will enable us to estimate the norm $||(AH)^+||$.

Theorem 4.3. *Suppose $A \in \mathbb{R}^{n \times n}$, $H \in \mathbb{R}^{n \times r}$, $\operatorname{rank}(A) = n \geq r$, $A = S_A \Sigma_A T_A^T$ is SVD (cf. (2.1)), and $\widehat{H} = T_A^T H$. Then*

$$\sigma_j(AH) \geq \sigma_l(A) \, \sigma_j(\widehat{H}_{l,r}) \text{ for all } l \leq n \text{ and all } j. \tag{4.1}$$

Proof. Note that $AH = S_A \Sigma_A T_A^T H$, and so $\sigma_j(AH) = \sigma_j(\Sigma_A T_A^T H) = \sigma_j(\Sigma_A \widehat{H})$ for all j by virtue of Lemma 4.1, because S_A is a square orthogonal matrix. Moreover it follows from Theorem 4.1 that $\sigma_j(\Sigma_A \widehat{H}) \geq \sigma_j(\Sigma_{l,A} \widehat{H}_{l,r})$ for all $l \leq n$. Combine this bound with the latter equations and apply Lemma 4.2. $\qquad \square$

Corollary 4.1. *Keep the assumptions of Theorem 4.3. Then*
(i) $\sigma_r(AH) \geq \sigma_\rho(A)\sigma_r(\widehat{H}_{n,r}) = \sigma_r(\widehat{H}_{n,r})/||A^+||$,
(ii) $||(AH)^+|| \leq ||A^+|| \, ||\widehat{H}_{n,r}^+||$ if $\mathrm{rank}(AH) = \mathrm{rank}(\widehat{H}_{n,r}) = r$.

Proof. Substitute $j = r$ and $l = n$ into bound (4.1), recall (2.2), and obtain part (i). If $\mathrm{rank}(AH) = \mathrm{rank}(\widehat{H}_{l,r}) = r$, then apply (2.2) to obtain that $\sigma_r(AH) = 1/||(AH)^+||$ and $\sigma_r(\widehat{H}_{l,r}) = 1/||\widehat{H}_{l,r}^+||$. Substitute these equations into part (i) and obtain part (ii). $\qquad \square$

Let us extend the estimates of Theorem 4.3 to the leading blocks of a matrix product.

Corollary 4.2. *Keep the assumptions of Theorem 4.3 and also suppose that the matrices $(AH)_{k,k}$ and $\widehat{H}_{n,k}$ have full rank k for a positive integer $k \leq n$. Then*

$$||(AH)_{k,k}^+|| \leq ||\widehat{H}_{n,k}^+|| \, ||A_{k,n}^+|| \leq ||\widehat{H}_{n,k}^+|| \, ||A^+||.$$

Proof. Note that $(AH)_{k,k} = A_{k,n}H_{n,k}$ and that the matrix $A_{k,n}$ has full rank. Apply Corollary 4.1 for A and H replaced by $A_{k,n}$ and $H_{n,k}$, respectively, and obtain that $||(AH)_{k,k}^+|| \leq ||\widehat{H}_{k,n}^+|| \, ||A_{n,k}^+||$. Combine (2.2) and Theorem 4.2 and deduce that $||A_{n,k}^+|| \leq ||A^+||$. Combine the two latter inequalities to complete the proof of part (i). Similarly prove part (ii). $\qquad \square$

Fact 4.1, Corollary 3.1 and Theorem 3.2 together imply the following result.

Corollary 4.3. *Suppose that $A \in \mathbb{R}^{n \times n}$, $H \in \mathbb{R}^{n \times r}$, $r \leq n = \mathrm{rank}(A)$, and the matrices $(AH)_{k,k}$ are strongly nonsingular and strongly well-conditioned for $k = 1, \ldots, r$. Then GENP and block Gaussian elimination are locally safe for the matrix product AH (see Remark 3.1 on the concept "locally safe").*

5 Benefits of using Gaussian multipliers for GENP and block Gaussian elimination

In Section 5.1 we recall the norm and condition estimates for Gaussian matrices and deduce that these matrices are strongly nonsingular with probability 1 and are expected to be strongly well-conditioned. In Section 5.2 we prove that the pair (H, A) for a nonsingular and well conditioned matrix A and a Gaussian matrix H is expected to satisfy the assumptions of Corollary 4.3, implying that the application of GENP and block Gaussian elimination to the product AH is numerically safe.

Remark 5.1. *The above results do not hold if the mean greatly exceeds standard deviation of the i.i.d. entries of a multipliers H. Its power for achieving numerically safe GENP and block Gaussian elimination is usually lost in this case. Indeed assume a mean μ and a standard deviation σ such that $\mu \gg \sigma$ (already $\mu > 10\sigma \log(n)$ is sufficient). In this case the matrix H is expected to be closely approximated by the rank-1 matrix $\mu \mathbf{e}\mathbf{e}^T$ where $\mathbf{e}^T = (1, 1, \ldots, 1)$.*

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.*

Theorem 5.1. *A Gaussian matrix G is strongly nonsingular with probability 1.*

Proof. Assume that the $j \times j$ leading submatrix $G^{(j)}$ of a $k \times l$ Gaussian matrix G is singular for some positive integer $j \le h = \min\{k, l\}$, that is, $\det(G^{(j)}) = 0$. Since $\det(G^{(j)})$ is a polynomial in the entries of the Gaussian matrix $G^{(j)}$, such matrices form an algebraic variety of a lower dimension in the linear space $\mathbb{R}^{j^2}$. (V is an algebraic variety of a dimension $d \le N$ in the space $\mathbb{R}^N$ if it is defined by $N - d$ polynomial equations and cannot be defined by fewer equations.) Clearly, Lebesgue (uniform) and Gaussian measures of such a variety equal 0, being absolutely continuous with respect to one another. Hence these measures of the union of h such matrices are also 0. $\qquad \square$

Theorem 5.2. *Assume a nonsingular $n \times n$ matrix A and an $n \times k$ Gaussian matrix $H_{n,k}$. Then the product $A_{k,n}H_{n,k}$ is nonsingular with probability 1.*

Proof. $\det(A_{k,n}H_{n,k})$ is a polynomials in the entries of the Gaussian matrix $H_{n,k}$. Such a polynomial vanishes with probability 0 unless it vanishes identically in $H_{n,k}$, but the matrix $A_{k,n}A_{k,n}^T$ is positive definite, and so $\det(A_{k,n}H_{n,k}) > 0$ for $H_{n,k} = A_{k,n}^T$. $\qquad \square$

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, while $\nu_{m,n}$, $\nu_{F,m,n}$, $\nu_{m,n}^+$, 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 5.3. *(Cf. [DS01, Theorem II.7].) Suppose $h = \max\{m, n\}$, $t \ge 0$, and $z \ge 2\sqrt{h}$. Then* Probability $\{\nu_{m,n} > z\} \le \exp(-(z - 2\sqrt{h})^2/2)$ *and* Probability $\{\nu_{m,n} > t + \sqrt{m} + \sqrt{n}\} \le \exp(-t^2/2)$.

Theorem 5.4. *(Cf. [CD05, Proof of Lemma 4.1].) Suppose $m \ge n \ge 2$, 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* Probability $\{\nu_{m,n}^+ \ge m/x^2\} < \frac{x^{m-n+1}}{\Gamma(m-n+2)}$.

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

Theorem 5.5. *If $m \ge n \ge 2$, then*

$$\text{Probability } \{\kappa_{m,n}m/(m - n + 1) > x\} \le \frac{1}{2\pi}(6.414/x)^{m-n+1}$$

for $x \ge m - n + 1$, while $\kappa_{m,1} = 1$ with probability 1.

Corollary 5.1. *A Gaussian matrix is expected to be strongly well-conditioned.*

5.2 Supporting GENP with Gaussian multipliers

The main result of this section is Corollary 5.3, which supports application of GENP and block Gaussian elimination to the product AH of a nonsingular matrix A and a Gaussian matrix H.

We need the following simple basic lemma.

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.*

Corollary 5.2. *Suppose A is a nonsingular $n \times n$ matrix, H is an $n \times k$ Gaussian matrix, for $0 < k \le n$, and $\nu_{g,h}$ and $\nu_{g,h}^+$ are the random values of Definition 5.2. Then*
(i) the matrix $(AH)_{k,k}$ is nonsingular with probability 1,
(ii) $||(AH)_{k,k}|| \le \nu_{n,k}||A_{k,n}|| \le \nu_{n,k}||A||$, and
(iii) $||(AH)_{k,k}^+|| \le \nu_{n,k}^+||A^+||$.

Proof. Part (i) restates Theorem 5.2. Part (ii) follows because $(AH)_{k,k} = A_{k,n}H_{n,k}$, $H_{n,k}$ is a Gaussian matrix, and $||A_{k,n}|| \le ||A||$. Part (iii) follows from Corollary 4.2 because $\widehat{H}_{n,k}$ is a Gaussian matrix by virtue of Lemma 5.1. $\qquad \square$

Corollary 5.3. *Suppose that the $n \times n$ matrix A is nonsingular and well-conditioned. Then the choice of Gaussian multiplier H is expected to satisfy the assumptions of Corollary 4.3.*

Proof. Recall that $(AH)_{k,k} = A_{k,n}H_{n,k}$ and hence $||(AH)_{k,k}|| = ||A_{k,n}||\nu_{n,k}$. Then combine Theorems 5.2, 5.3, and 5.4 and Corollary 5.2. $\qquad\square$

6 Random structured multipliers for GENP and block Gaussian elimination

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, so that $M^H = M^T$ for a real matrix M.

Hereafter $\omega = \omega_n = \exp(\frac{2\pi}{n}\sqrt{-1})$ denotes an nth primitive root of unity, $\Omega = (\omega^{ij})_{i,j=0}^{n-1}$ is the matrix of the discrete Fourier transform at n points (we use the acronym *DFT*), and $\Omega^{-1} = \frac{1}{n}\Omega^H$.

An $n \times n$ circulant matrix $C = (c_{i-j \mod n})_{i,j=0}^{n-1}$ is defined by its first column $\mathbf{c} = (c_i)_{i=0}^{n-1}$.

Example 6.1. 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 $\mathbf{c}$.*

The following theorem links the matrices Ω and Ω^{-1} to the class of circulant matrices.

Theorem 6.1. *(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 $C = \Omega^{-1} \operatorname{diag}(u_j)_{j=1}^n \Omega$. Furthermore $C^{-1} = \Omega^{-1} \operatorname{diag}(1/u_j)_{j=1}^n \Omega$ if the matrix C is nonsingular.*

By using FFT, one can multiply the matrices Ω and $\Omega^H = \Omega^{-1}$ by a vector by using $O(n\log(n))$ flops for any n (cf., e.g., [P01, page 29]), and Theorem 6.1 extends this complexity bound to multiplication of an $n \times n$ circulant matrix and its inverses by a vector.

We need $2n^3 - n^2$ flops in order to compute the product AH of the pair of $n \times n$ matrices A and H. If, however, H is a circulant matrix, then we can compute AH by using order of $n^2 \log(n)$ flops. For a Toeplitz-like matrix A defined by its displacement generator of bounded length l, we use $O(ln\log(n))$ flops in order to compute a displacement generator of length l for the matrix AH. (See [P01] for the definition of displacement generators.) In the case of Toeplitz matrices we have $l \leq 2$ and use $O(n\log(n))$ flops. This motivates using Gaussian circulant multipliers H, that is, circulant matrices H whose first column vector is Gaussian. It has been proved in [PSZa] that such matrices are expected to be well-conditioned, which is required for any multiplicative preconditioner.

We can define a unitary circulant matrix by its first column vector $\mathbf{c} = \Omega(\exp(r_i\sqrt{-1}))_{i=0}^{n-1}$ for any set of real values $r_0, \ldots, r_{n-1}$.

Example 6.2. 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 $\phi_1, \ldots, \phi_n$ are n independent random real variables, e.g., Gaussian variables or the variables uniformly distributed in the range $[0,1)$.
(ii) Compute the vector $\mathbf{c} = \Omega^{-1}\mathbf{u}$, where Ω denotes the $n \times n$ DFT matrix. Output the unitary circulant matrix C defined by its first column $\mathbf{c}$.

Our proof that Gaussian multipliers enforce strong nonsingularity of a nonsingular matrix with probability 1 (see Theorem 5.2) has been non-trivially extended in [PZa] to the case of Gaussian circulant multipliers. Furthermore strong nonsingularity holds with probability close to 1 if we fill the first column of a multiplier F or H with i.i.d. random variables defined under the uniform probability distribution over a sufficiently large finite set (see Appendix B and [PSZa]).

In our tests with random input matrices, Gaussian circulant and general Gaussian multipliers have shown the same power of supporting numerically safe GENP (see Section 8.1), but we cannot extend our basic Lemma 5.1 and our Corollary 5.3 to the case of circulant matrices. Moreover our Theorem 6.2 and Remark 6.1 below show that, for a specific narrow class of input matrices A, GENP with these multipliers is expected to fail numerically.

Theorem 6.2. *Assume a large integer n and the $n \times n$ DFT matrix Ω, which is unitary up to scaling by $1/\sqrt{n}$.*

(i) Then application of GENP to this matrix fails numerically and

(ii) a Gaussian circulant $n \times n$ multiplier $C = \Omega^{-1} D \Omega$ with Gaussian diagonal matrix $D = \mathrm{diag}(g_j)_{j=1}^n$ (having i.i.d. Gaussian diagonal entries $g_1, \dots, g_n$) is not expected to fix this problem.

Proof. (i) Subtract the first row of the block $\Omega_{2,2}$ of the matrix Ω and the resulting vector from its second row. Obtain the vector $(0, \omega - 1)$ with the norm $|\omega - 1| = 2\sin(\pi/n)$. Assume that n is large and then observe that $2\sin(\pi/n) \approx 2\pi/n$ and that the variable $2\pi/n$ is expected to be small, implying that $\mathrm{nrank}(\Omega_{2,2}) = 1$ because $||\Omega_{2,2}|| \geq \sqrt{2}$ for large n.

(ii) Note that $\Omega C = D\Omega$. The Gaussian variable g_1 vanishes with probability 0, and so we can assume that $g_1 \neq 0$. Multiply the first row of the block $(D\Omega)_{2,2}$ of the matrix $D\Omega$ by g_2/g_1 and subtract the resulting vector from the second row. Obtain the vector $(0, (\omega - 1)g_2)$ with the norm $|(\omega - 1)g_2| = 2|g_2 \sin(\pi/n)|$ Assume that n is large and then observe that $|(\omega - 1)g_2| \approx 2|g_2|\pi/n$ and that the variable $2|g_2|\pi/n$ is expected to be small. Hence $\mathrm{nrank}((\Omega C)_{2,2}) = \mathrm{nrank}((D\Omega)_{2,2})$ is expected to equal 1 because $||\Omega C)_{2,2}|| \leq ||\Omega_{2,2}|| \max\{g_1, g_2\}$, $||\Omega_{2,2}|| \geq \sqrt{2}$ and the random variable $\max\{|g_1|, |g_2|\}$ is not expected to be close to 0. $\qquad\square$

Remark 6.1. *The same argument shows that Gaussian circulant multipliers C are not expected to support GENP for a bit larger class of matrices, e.g., for $A = M\Omega$ where $M = \mathrm{diag}(D_i)_{i=1}^k$, $D_1 = \mathrm{diag}(d_1, d_2)$, and d_1 and d_2 are two positive constants and the input size $n \times n$ is large as well as where the matrix M is strongly diagonally dominant. The reader is challenged to find out whether GENP with a Gaussian circulant preprocessor is expected to fail numerically for other classes of input matrices, in particular for any subclass of the classes of Toeplitz or Toeplitz-like matrices (cf. [P01] and [P15] on these classes). Another challenge is to choose a distinct random structured preprocessor for which the above problem is avoided. E.g., consider the product $\prod_{i=1}^h C_i$ where h is a small integer exceeding 1, C_{2j} are circulant matrices and C_{2j-1} are skew-circulant (see the definition in [P01]). Toward the same goal we can apply simultaneously random structured pre- and post-multipliers F and H, defined by some i.i.d. random parameters, or the pairs of PRMB multipliers of [BBD12]. In the case of Toeplitz or Toeplitz-like input matrices A, the multiplications FA and AH are much less costly if the multipliers F and H are circulant matrices, skew-circulant matrices, or the products of such matrices.*

7 Low-rank approximation

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 SVD of the matrix A or, at a lower cost, by computing its rank-revealing factorization [GE96], [HP92], [P00a], but using random matrix multipliers instead has some benefits [HMT11]. In this section we study the latter randomized approach. In its first subsection we recall some relevant definitions and auxiliary results.

7.1 Truncation of SVD. Leading and trailing singular spaces

Truncate the square orthogonal matrices S_A and T_A and the square diagonal matrix Σ_A of the SVD of (2.1), write $S_{\rho,A} = (S_A)_{m,\rho}$, $T_{\rho,A} = (T_A)_{n,\rho}$, and $\Sigma_{\rho,A} = (\Sigma_A)_{\rho,\rho} = \mathrm{diag}(\sigma_j)_{j=1}^\rho$, and obtain *thin SVD*

$$A = S_{\rho,A} \Sigma_{\rho,A} T_{\rho,A}^T, \ \rho = \mathrm{rank}(A). \tag{7.1}$$

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

$$A_r = S_{r,A} \Sigma_{r,A} T_{r,A}^T, \ \bar{A}_r = \bar{S}_{A,r} \bar{\Sigma}_{A,r} \bar{T}_{A,r}^T, \ A = A_r + \bar{A}_r \text{ for } 1 \leq r \leq \rho = \mathrm{rank}(A), \tag{7.2}$$

11

and call the above decomposition the *r-truncation of thin SVD* (7.1). Note that $\bar{A}_\rho$ is an empty matrix and recall that

$$||A - A_r|| = \sigma_{r+1}(A). \tag{7.3}$$

Let $\mathbb{S}_{r,A}$ and $\mathbb{T}_{r,A}$ denote the ranges (that is, the column spans) of the matrices $S_{r,A}$ and $T_{A,r}$, respectively. 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 if the matrices X and Y are orthogonal.

7.2 The basic algorithm

Assume an $m \times n$ matrix A having a small numerical rank r and a Gaussian $n \times r$ matrix 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^TA approximates the matrix A.

In this subsection we recall the algorithm supporting this theorem, where temporarily we assume nonrandom multipliers H. In the next subsections we keep it nonrandom and estimate the output approximation errors of the algorithm assuming no oversampling, suggested in [HMT11]. Then we extend our study to the case where H is a Gaussian, and in Section 7.5 cover the results in the case of random structured multipliers.

Algorithm 7.1. Low-rank approximation of a matrix. *(Cf. Remarks 7.1 and 7.2.)*

INPUT: *A matrix $A \in \mathbb{R}^{m \times n}$, its numerical rank r, and two integers $p \geq 2$ and $l = r+p \leq \min\{m,n\}$.*

OUTPUT: *an orthogonal matrix $Q \in \mathbb{R}^{m \times l}$ such that the matrix $QQ^TA \in \mathbb{R}^{m \times n}$ has rank at most l and approximates the matrix A.*

INITIALIZATION: *Generate an $n \times l$ matrix H.*

COMPUTATIONS:

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

This basic algorithm from [HMT11] uses $O(lmn)$ flops overall.

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

In Corollaries 7.1 and 7.2 of this subsection we estimate the error norms for the approximations computed by Algorithm 7.1 whose oversampling parameter p is set to 0, namely for the approximation of an orthogonal basis for the leading singular space $\mathbb{S}_{r,A}$ (by column set of the matrix Q of the algorithm) and for a rank-r approximation of the matrix A. We first recall the following results.

Theorem 7.1. *(Cf. (C.1).) Suppose A is an $m \times n$ matrix, $S_A \Sigma_A T_A^T$ is its SVD, r is an integer, $0 < r \leq l \leq \min\{m,n\}$, and $Q = Q_{r,A}$ is an orthogonal matrix basis for the space $\mathbb{S}_{r,A}$. Then $||A - QQ^TA|| = \sigma_{r+1}(A)$.*

Theorem 7.2. *Assume two matrices $A \in \mathbb{R}^{m \times n}$ and $H \in \mathbb{R}^{n \times r}$ and define the two matrices A_r and $\bar{A}_r$ of (7.2). Then $AH = A_rH + \bar{A}_rH$ where $A_rH = S_{r,A}U$, $U = \Sigma_{r,A}T_{r,A}^TH$. Furthermore the columns of the matrix A_rH span the space $\mathbb{S}_{r,A}$ if $\mathrm{rank}(A_rH) = r$.*

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 7.3. *Keep the assumptions of Theorem 7.2. Then*

(i) $||\bar{A}_r H||_F \le \sigma_{r+1}(A) \, ||H||_F.$

(ii) Furthermore if the matrix $T_{r,A}^T H$ is nonsingular, then $||(A_r H)^+|| \le ||(T_{r,A}^T H)^{-1}||/\sigma_r(A).$

Proof. Recall that

$$||U|| = ||U||_F = 1, \ ||UAV|| \le ||A||, \text{ and } ||UAV||_F \le ||A||_F \text{ for orthogonal matrices } U \text{ and } V. \ (7.4)$$

Then note that $||\bar{A}_r H||_F = ||\bar{S}_{A,r} \bar{\Sigma}_{A,r} \bar{T}_{A,r}^T H||_F \le ||\bar{\Sigma}_{A,r} \bar{T}_{A,r}^T H||_F$ by virtue of bound (7.4).

Combine this bound with Lemma 4.2 and obtain that $||\bar{A}_r H||_F \le \sigma_{r+1}(A) \, ||\bar{T}_{A,r}^T H||_F$, which is not greater than $\sigma_{r+1}(A) \, ||H||_F$ by virtue of bound (7.4). 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 $||S_{r,A}|| = 1$ and $||\Sigma_{r,A}^{-1}|| = 1/\sigma_r(A)$. $\square$

Combine Theorems C.2, 7.2, and 7.3 to obtain the following estimates.

Corollary 7.1. *Keep the assumptions of Theorem 7.2, let the matrix $T_{r,A}^T H$ be nonsingular and write*

$$||E||_F = \sigma_{r+1}(A) \, ||H||_F,$$

$$\Delta_+ = \sqrt{2} \, ||E||_F \, ||(T_{r,A}^T H)^{-1}|| \, /\sigma_r(A) = \Delta_+ = \sqrt{2} \, ||H||_F \, ||(T_{r,A}^T H)^{-1}|| \, \sigma_{r+1}(A)/\sigma_r(A).$$

Then

$$\Delta = ||Q(A_r H)^T - Q(AH)^T|| \le \Delta_+ + O(||E||_F^2).$$

Next combine Corollary C.1 with Theorem 7.1 and employ the orthogonal projection $P_{AH} = Q(AH)Q(AH)^T$ (cf. (C.1)) to extend the latter estimate to bound the error norm of low-rank approximation of a matrix A by means of Algorithm 7.1.

Corollary 7.2. *Keep the assumptions of Corollary 7.1 and write $\Delta'_+ = \sigma_{r+1}(A) + 2\Delta_+||A||$. Then*

$$\Delta' = ||A - P_{AH}A|| \le \Delta'_+ + O(||E||_F^2||A||).$$

Proof. Note that $||A - P_{AH}A|| \le ||A - P_M A|| + ||(P_M - P_{AH})A||$ for any $m \times r$ matrix M. Write $M = A_r H$, apply Theorem 7.1 and obtain $||A - P_M A|| = \sigma_{r+1}(A)$. Corollaries C.1 and 7.1 together imply that $||(P_M - P_{AH})A|| \le ||A|| \, ||P_{A_r H} - P_{AH}|| \le 2\Delta||A||$. Combine the above relationships. $\square$

Remark 7.1. *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 $A \to B_i$ for $i = 1, 2, \ldots$ 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 7.1, 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 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 $A = B_0 \to B_i$ 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 the SVD and rank-revealing factorizations.*

Remark 7.2. *Let us summarize our analysis. Suppose that the ratio $\sigma_r(A)/\sigma_{r+1}(A)$ is large and that the matrix product $P = T_{r,A}^T H$ has full rank r and is well-conditioned. Now set to 0 the oversampling integer parameter p of Algorithm 7.1. Then, by virtue of Theorem 7.3 and Corollaries 7.1 and 7.2, the algorithm outputs a close approximation $Q(AH)$ to an orthogonal bases for the leading singular space $\mathbb{S}_{r,A}$ of the input matrix A and 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 a low computational cost, we can decrease the error norm even below the value $\sigma_{r+1}(A)$.*

7.4 Supporting low-rank approximation with Gaussian multipliers

In this subsection we extend the results of the previous one to support the choice of Gaussian multiplier H in Algorithm 7.1, whose "actual outcome is very close to the typical outcome because of the measure concentration effect" [HMT11, page 226].

Theorem 7.4. *Suppose $A \in \mathbb{R}^{m \times n}$, $A = S_A \Sigma_A T_A^T$ is its SVD of (2.1), $H = \mathbb{R}^{n \times r}$ is a Gaussian matrix, and $\text{rank}(A) = \rho \geq r$.*
(i) Then the matrix $T_{r,A}^T H$ is Gaussian.
(ii) Assume the values Δ_+ and Δ'_+ of Corollaries 7.1 and 7.2 and the values $\nu_{F,n,r}$ and $\nu_{r,r}^+$ of Definition 5.2. Then $\Delta_+ = \sqrt{2} \, \nu_{F,n,r} \nu_{r,r}^+ \sigma_{r+1}(A)/\sigma_r(A)$ and $\Delta'_+ = \sigma_{r+1}(A) + 2\Delta_+ ||A||$.

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. This proves part (i), which implies part (ii). $\square$

Corollary 7.3. *A Gaussian multiplier H is expected to support safe numerical application of Algorithm 7.1 even where the oversampling integer parameter p is set to 0.*

Proof. Combine Theorems 5.1 and 7.4 with Corollary 5.1. $\square$

7.5 Supporting low-rank approximation with random structured multipliers

Multiplication of an $n \times n$ matrix A by a Gaussian matrix H at Stage 1 of Algorithm 7.1 requires $(2r-1)n^2$ flops, but we can save a factor of $r/\log(r)$ flops by applying structured random multipliers H. In particular we can use subsampled random Fourier transforms (SRFTs) of [HMT11, equation (4.6)], subsampled random Hadamard transforms (SRHTs) of [T11], the chains of Givens rotations (CGRs) of [HMT11, Remark 4.5.1], and the leading Toeplitz submatrices $C_{n,r}$ and $C_{r,n}$ of random circulant $n \times n$ matrices C. We need just n random parameters to define a Gaussian circulant $n \times n$ matrix C and its leading Toeplitz blocks $C_{n,r}$ and $C_{r,n}$, and similarly for the other listed classes of structured matrices.

Example 7.1. *For two fixed integers l and n, $1 < l < n$, SRFT $n \times l$ matrices are the matrices of the form $S = \sqrt{n/l} \, D\Omega R$. Here 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, equation (4.6) and Section 11]).*

Theorem 6.1 implies the following fact.

Corollary 7.4. *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.*

According to the 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 8.2, Toeplitz multipliers defined as the $n \times r$ leading blocks of $n \times n$ random circulant matrices consistently supported low-rank approximation without oversampling as efficiently as Gaussian multipliers.

As in the case of our randomized support for GENP and block Gaussian elimination, formal analysis of the impact of random structured multipliers is complicated because we cannot use Lemma 5.1. Nevertheless, by allowing substantial oversampling, one can still prove that SRFT multipliers are expected to support low-rank approximation of a matrix having a small numerical rank.

Theorem 7.5. Error bounds for low-rank approximation with SRFT *(cf. [HMT11, Theorem 11.2]).* *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 $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots$, an $n \times l$ SRFT matrix S of Example 7.1, and $Y = AS$. Then with a probability $1 - O(1/r)$ it holds that*

$$||(I - P_Y)A|| \leq \sqrt{1 + 7n/l} \ \sigma_{r+1} \ \text{and} \ ||(I - P_Y)A||_F \leq \sqrt{1 + 7n/l} \ (\sum_{j>r} \sigma_j^2)^{1/2}.$$

Remark 7.3. *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 $US = CR$ for the matrix R of Example 7.1 and the circulant matrix $C = \Omega^{-1}D\Omega$ (cf. Theorem 6.1). By virtue of Theorem 7.5 we can expect that Algorithm 7.1 would produce a rank-r approximation if we choose a multiplier H being an SRFT $n \times l$ matrix or the $n \times l$ submatrix CP of $n \times n$ random unitary circulant matrix C made up of its l randomly selected columns where the selection is defined by the matrix P of Example 7.1 and where l is an integer 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 $O(n^2 l)$ in the straightforward algorithm.*

8 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 6.1. 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.

8.1 GENP with Gaussian and random circulant multipliers

We applied both GENP and the preprocessed GENP to $n \times n$ DFT matrices $A = \Omega$ and to 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 $i = 1, \ldots, k - 4$ and $\sigma_i = 0$ for $i = k - 3, \ldots, k$ 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 5.3 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 6.1 and 6.2 (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. The output residual norms, however, were small enough to support application of the inexpensive iterative refinement. Already its single
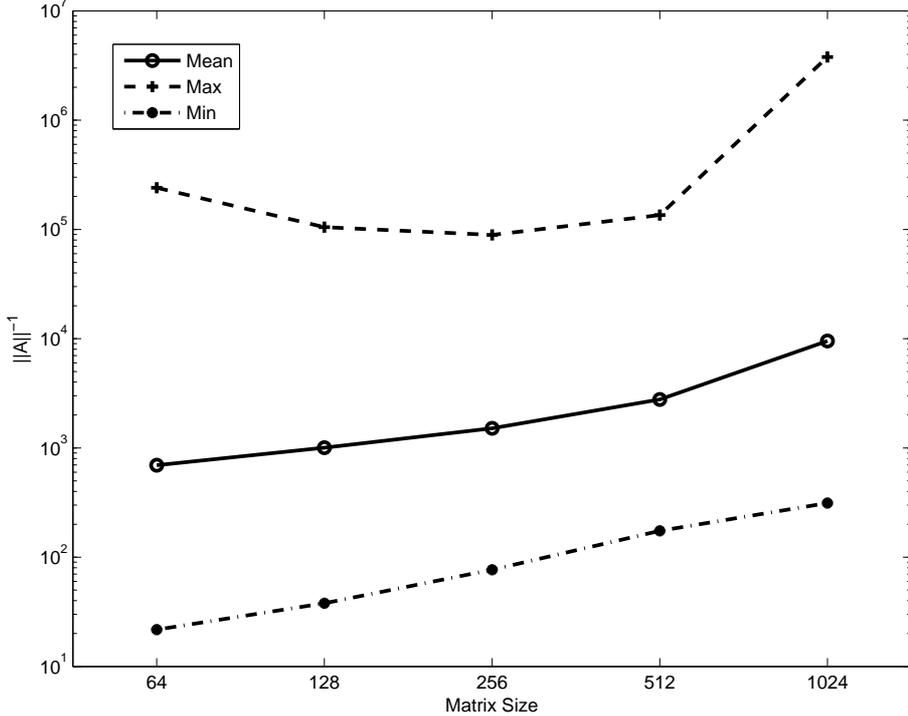
Figure 1: Norm of A^{-1}

step decreased the average relative residual norm below 10^{-11} for $n = 1024$ in all our tests with Gaussian multipliers and to about 10^{-13} for $n = 1024$ in all our tests with circulant multipliers of Examples 6.1 and 6.2. See further details in Figures 2 and 3 and Tables D.3–D.5. This indicates that GENP with preprocessing followed by even a single step of iterative refinement is backward stable, similarly to the celebrated result of [S80].

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 6. 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 we were not seeking the solution, but were trying to compare the performance of 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 on the inputs Ω both when we used no preprocessing and used preprocessing with random circulant multipliers of Examples 6.1 and 6.2. In these cases the 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 GEPP. Table D.6 displays these norms in some detail.

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

We approximated the r-dimensional leading singular spaces of $n \times n$ matrices A that have numerical rank r, and we also approximated 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 $\sigma_j = 1/j$, $j = 1, \ldots, r$, $\sigma_j = 10^{-10}$, $j = r+1, \ldots, n$ (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

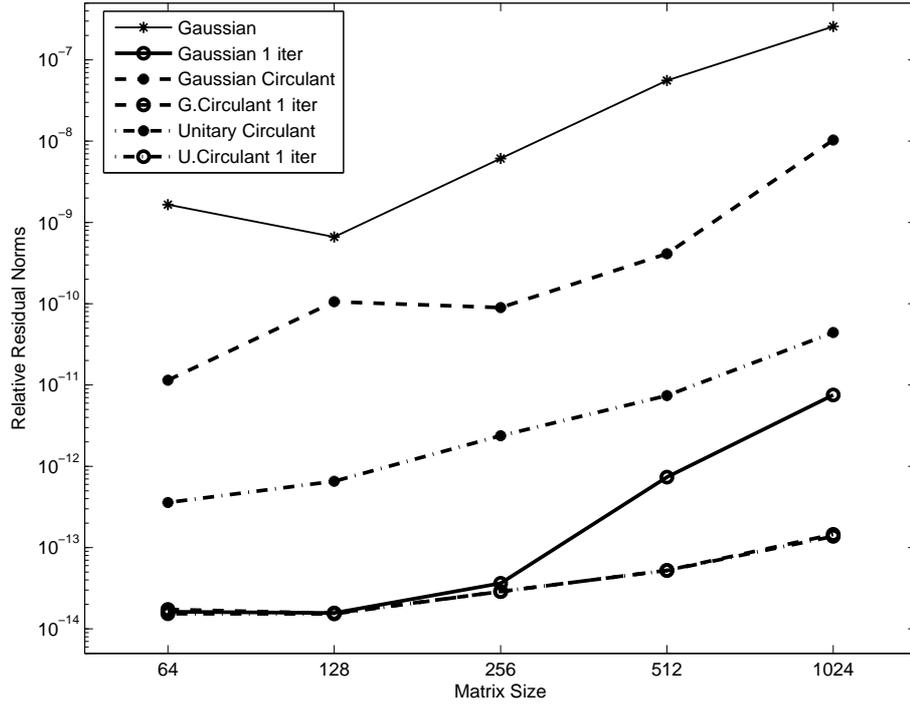Figure 2: Average relative residual norms for GENP by using random multipliers. The two broken lines representing one iteration of circulant multipliers are overlapping at the bottom of the display
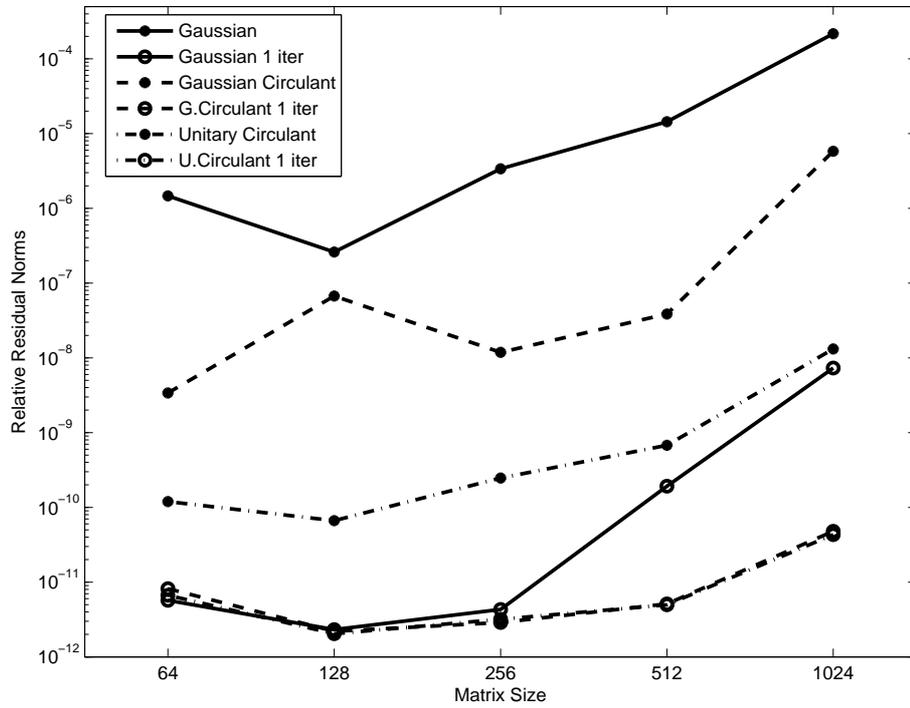

Figure 3: Maximum relative residual norms for GENP by using random multipliers. The two broken lines representing one iteration of circulant multipliers are overlapping at the bottom of the display
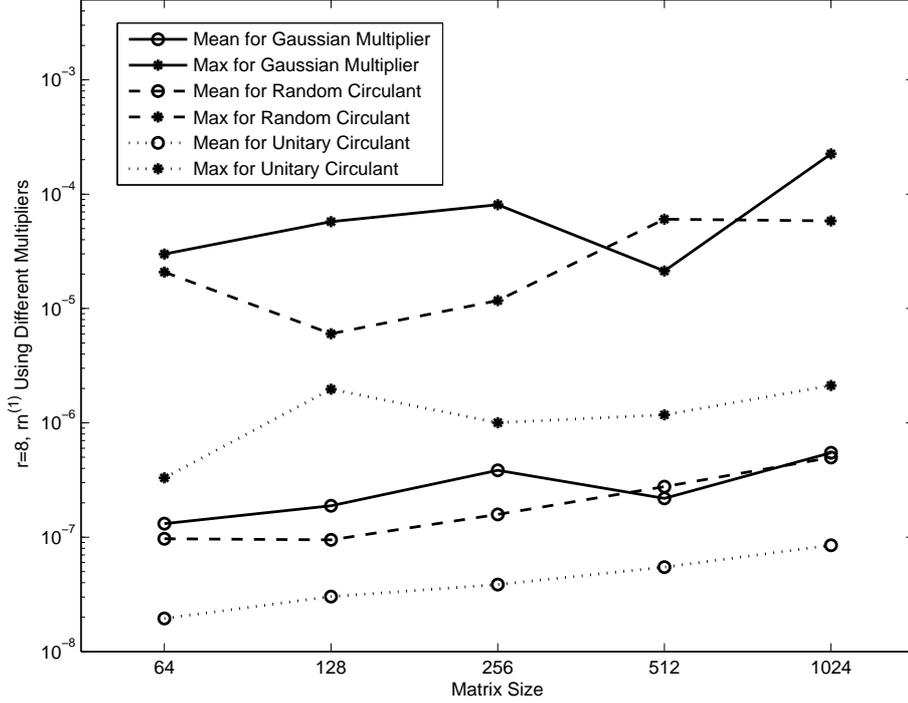
17

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

Tables D.7–D.12 display the resulting data on the residual norms $rn^{(1)} = ||Q_{r,A}Y_{r,A} - S_{r,A}||$ and $rn^{(2)} = ||A - Q_{r,A}Q_{r,A}^T A||$, obtained in 1000 runs of our tests for every pair of n and r. In these figures and tables $rn^{(1)}$ denotes the residual norms of the approximations of the matrix bases for the leading singular spaces $\mathbb{S}_{r,A}$, and $rn^{(2)}$ 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 $rn^{(1)}$. The last column of each of the tables displays the ratio of the observed values $rn^{(1)}$ 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 7.1). 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. Tables D.8 and D.9 cover the cases where we generated random $n \times n$ circulant matrices of Examples 6.1 and 6.2, respectively, and applied their $n \times r$ Toeplitz leading blocks as multipliers H.

Figures 6 and 7 and Tables D.10–D.12 show similar results of our tests for the observed residual norms $rn^{(2)}$ 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 7.2).

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 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 6.1, 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].

9 Conclusions

It is known that a standard Gaussian random matrix (we refer to it as Gaussian for short) has full rank with probability 1 and is well-conditioned with a probability close to 1. These properties mo-
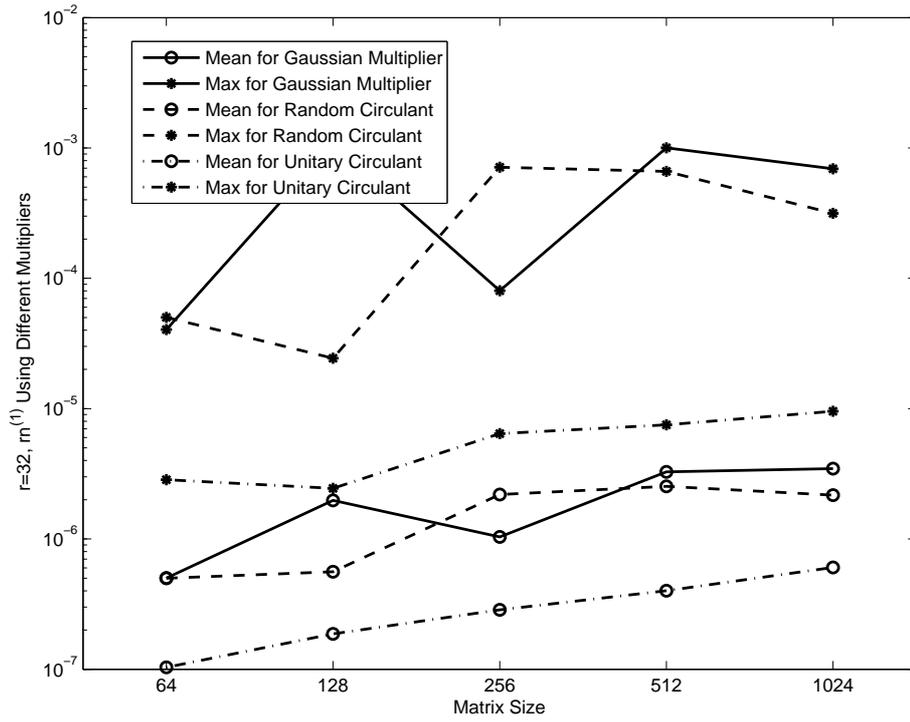
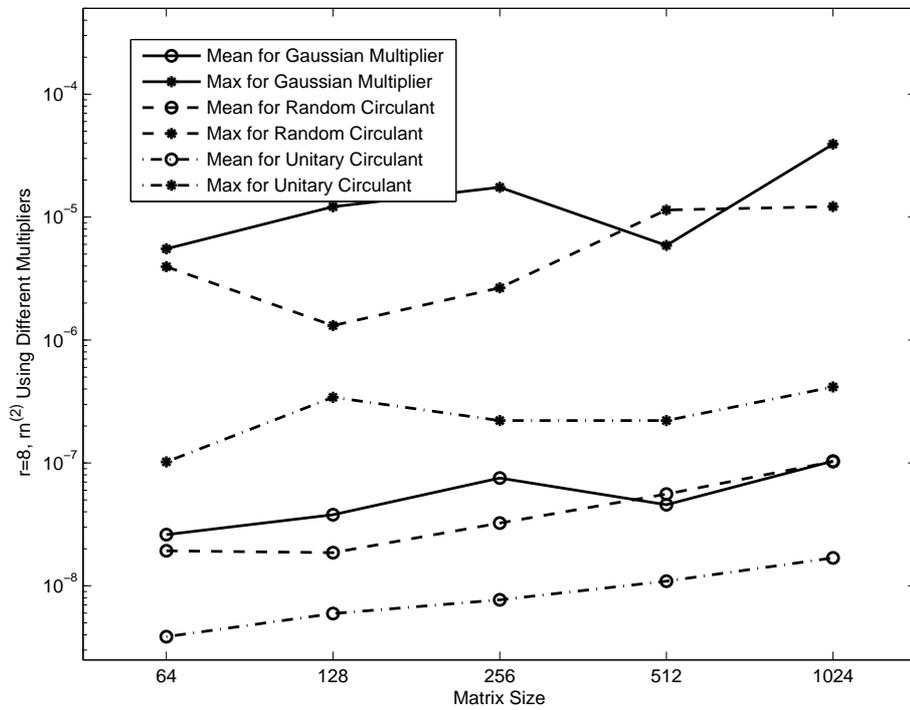Figure 5: Residual norms $rn^{(1)}$ using different random multipliers, case r=32

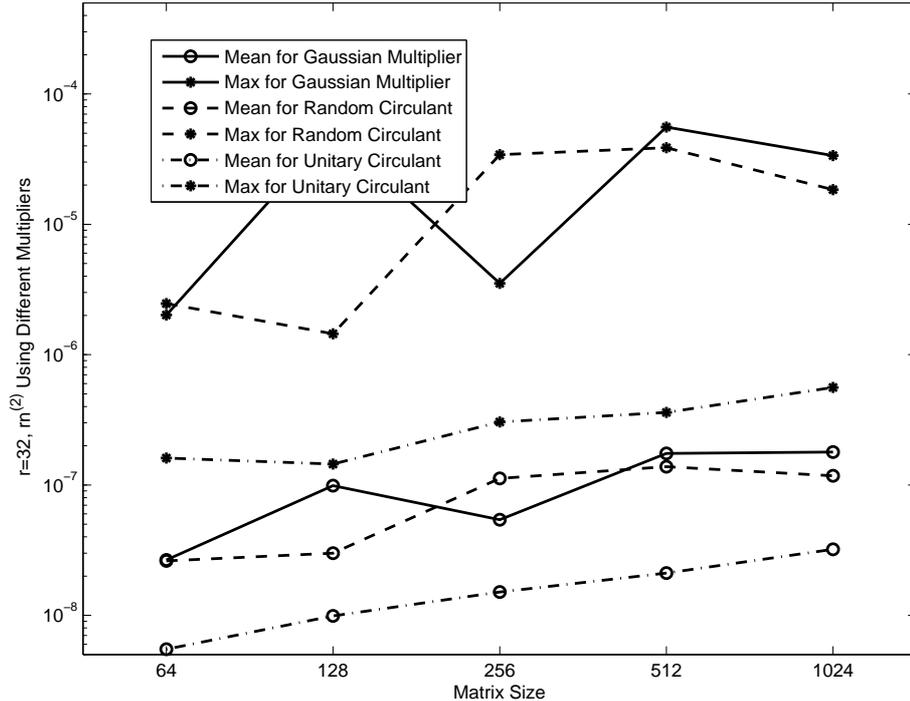Figure 6: Residual norms $rn^{(2)}$ using different random multipliers, case r=8

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

tivated our application of Gaussian multipliers to advancing matrix computations. In particular we preprocessed well-conditioned nonsingular input matrices by using Gaussian multipliers to support GENP (that is, Gaussian elimination with no pivoting) and block Gaussian elimination. These algorithms can readily fail in practical numerical computations without preprocessing, but we proved that we can avoid these problems with a probability close to 1 if we preprocess the input matrix by pre- or post-multiplying it by a Gaussian matrix.

Our tests were in good accordance with that formal result, that is, we generated matrices that were hard for GENP, but the problems were consistently avoided when we preprocessed the inputs with Gaussian multipliers. In that case a single loop of iterative refinement was always sufficient to match the output accuracy of the customary algorithm of GEPP, indicating that GENP with preprocessing followed by even a single step of iterative refinement is backward stable, similarly to the celebrated result of [S80].

In our tests we observed similar results even where we applied Gaussian circulant (rather than Gaussian) multipliers. Under this choice we generated only n random parameters for an $n \times n$ input, and the multiplication stage was accelerated by a factor of $n/\log(n)$. The acceleration factor increases to $n^2/\log(n)$ when the input matrix has the structure of Toeplitz type, but we could support numerical stabilization of GENP with Gaussian circulant multipliers only empirically. Moreover, we proved that with a high probability Gaussian circulant multipliers cannot fix numerical instability of the elimination algorithms for a specific narrow class of inputs (see Theorem 6.2 and Remark 6.1).

This should motivate the search for alternative randomized structured multipliers that would be expected to stabilize numerical performance of GENP and block Gaussian elimination for any input or, say, for any Toeplitz and Toeplitz-like input matrix. Among the candidate multipliers, one can consider the products of random circulant and skew-circulant matrices, possibly used as both pre- and post-multipliers. Suppose that indeed they are expected to stabilize block Gaussian elimination numerically. Then their support would be valuable for numerical application of the MBA celebrated algorithm, because it is superfast for Toeplitz and Toeplitz-like input matrices and hence for their products with circulant and skew-circulant matrices (cf., e.g., [B85], [P01, Chapter 5], and [PQZ11]).

We have extended our analysis to the problem of rank-r approximation of an $m \times n$ matrix A having a numerical rank r. With a probability close to 1 the column set of the matrix AH, for an $n \times (r+p)$ Gaussian matrix H and a small positive oversampling integer parameter p, approximates a 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 an approximate basis available, one can readily approximate the matrix A by a matrix of rank r.

This is an efficient, well developed algorithm (see [HMT11]), but we proved that this algorithm is expected to produce a reasonable rank-r approximation with Gaussian multipliers already for $p = 0$, that is, 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, defined as the maximal leading submatrices of random circulant matrices. This has accelerated the multiplication stage and has limited 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 we immediately extended it to the case when the products of random unitary circulant multipliers and random rectangular permutation matrices were applied instead of the SRFT matrices (see Section 6).

A natural research challenge is the combination of our randomized multiplicative preprocessing with randomized augmentation and additive preprocessing, studied in [PQ10], [PQ12], [PQZC], [PQZ13], and [PQZb].

References

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

[B85]   J. R. Bunch, Stability of Methods for Solving Toeplitz Systems of Equations, *SIAM J. Sci. Stat. Comput.*, **6(2)**, 349–364, 1985.

[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.

[BCD14]   G. Ballard, E. Carson, J. Demmel, M. Hoemmen, N. Knight, O. Schwartz, Communication Lower Bounds and Optimal Algorithms for Numerical Linear Algebra. *Acta Numerica*, **23**, 1–155, 2014.

[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.

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

[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.

[P15]   V. Y. Pan, Transformations of Matrix Structures Work Again, *Linear Algebra and Its Applications*, 465, 1–32, 2015. 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, 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.

[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*, 2012.

           Available at http://www.cs.gc.cuny.edu/tr/techreport.php?id=438 and

           http://arxiv.org/abs/1210.7476

[PQZb]   V. Y. Pan, G. Qian, L. Zhao, Randomized Augmentation and Additive Preprocessing, preprint, 2014.

[PQZC]   V. Y. Pan, G. Qian, A. Zheng, Z. Chen, Matrix Computations and Polynomial Root-finding with Preprocessing, *Linear Algebra and Its Applications*, 2014, DOI: 10.1016/j.laa.2014.06.027. Also arxiv:1311.3730[math.NA].

[PSZa]   V. Y. Pan, J. Svadlenka, L. Zhao, Estimating the Norms of Circulant and Toeplitz Random Matrices and Their Inverses, *Linear Algebra and Its Applications*, 2014 (in press), DOI: 10.1016/j.laa.2014.06.027. Also arxiv:1311.3730[math.NA]

[PZa]   V. Y. Pan, L. Zhao, A Single Random Triangular Toeplitz Multiplier Ensures Strong Nonsingularity, preprint, 2014.

[S80]   R. D. Skeel, Iterative Refinement Im[plies Numerical Stability for Gaussian Elimimation, *Math. of Computation*, **35, 151**, 817–832, 1980.

[S80a]   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, Philadelphia, 1998.

[S03]   G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, 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 EUROSAM'79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.

Appendix

A   Norm and condition of Gaussian matrices and the errors of randomized low-rank approximation

Let us reproduce some known bounds for the expected values of the norms and condition numbers of random matrices.

Theorem A.1. *(i)* $\mathbb{E}(\nu_{n,n}) \leq 2\sqrt{n}$, *(ii)* $\mathbb{E}(\log(\kappa_{m,n})) \leq \log(\frac{n}{m-n+1}) + 2.258$ *for* $m \geq n \geq 2$.

Proof. See [S91] for part (i) and [CD05, Theorem 6.1] for part (ii).   □

The bounds of part (i) of the theorem are quite tight (cf. Theorem 5.3). The bounds of part (ii) imply the following more specific estimates.

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

The paper [HMT11] proposed using Algorithm 7.1 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 $P_{AH} = QQ^T$,

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

24

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

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

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

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

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

with a probability at least $1 - 10^{-r}$. Here $\mathbf{g}_j$ is the jth column of $n \times r$ Gaussian matrix, that is, $\mathbf{g}_1, \ldots, \mathbf{g}_r$ are r independent Gaussian vectors of length n, and r is an integer parameter (see our Remark 7.1 on improving this approximation). Here is an alternative simplified expression from [HMT11, equation (1.9)],

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

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 5.2 and provides some reasonable formal support even where $p = 0$.

B   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 B.1. *[DL78], [S80a], [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 B.1. *Under the assumptions of Lemma B.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 B.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. Clearly the claims of the corollary hold for generic matrices. Now note that 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 B.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. $\qquad \square$

C Perturbation errors of matrix inversion

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

Theorem C.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,

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

Corollary C.1. *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 C.2. □

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 6.1

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 6.2

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 and iterative refinement

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 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** 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 rn$^{(1)}$ and the mean ratios of them and their upper bounds $\tilde{\delta}_+$, in the case of using Toeplitz random multipliers and Example 6.1

q	n	**mean**	**max**	**mean of ratio** 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 rn$^{(1)}$ and the mean ratios of them and their upper bounds $\tilde{\delta}_+$, in the case of using Toeplitz random multipliers and Example 6.2

q	n	**mean**	**max**	**mean of ratio** 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 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** 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 rn$^{(2)}$ and the mean ratio of them and their upper bounds, in the case of using Toeplitz random multipliers and Example 6.1

q	n	**mean**	**max**	**mean of ratio** 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 rn$^{(2)}$ and the mean ratio of them and their upper bounds, in the case of using Toeplitz random multipliers and Example 6.2

q	n	**mean**	**max**	**mean of ratio** rn$^{(2)}/\tilde{\Delta}_+$
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 6.1

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	$1.16 \times 10^{+2}$	$3.48 \times 10^{+3}$	$1.97 \times 10^{+1}$	$1.79 \times 10^{+2}$