SIAM J. MATRIX ANAL. APPL.
Vol. 35, No. 4, pp. 1490–1520

THE EFFECT OF COHERENCE ON SAMPLING FROM MATRICES WITH ORTHONORMAL COLUMNS, AND PRECONDITIONED LEAST SQUARES PROBLEMS*

ILSE C. F. IPSEN† AND THOMAS WENTWORTH†

Abstract. Motivated by the least squares solver *Blendenpik*, we investigate three strategies for uniform sampling of rows from $m \times n$ matrices Q with orthonormal columns. The goal is to determine, with high probability, how many rows are required so that the sampled matrices have full rank and are well-conditioned with respect to inversion. Extensive numerical experiments illustrate that the three sampling strategies (without replacement, with replacement, and Bernoulli sampling) behave almost identically, for small to moderate amounts of sampling. In particular, sampled matrices of full rank tend to have two-norm condition numbers of at most 10. We derive a bound on the condition number of the sampled matrices in terms of the coherence μ of Q. This bound applies to all three different sampling strategies; it implies a, not necessarily tight, lower bound of $\mathcal{O}(m\mu \ln n)$ for the number of sampled rows; and it is realistic and informative even for matrices of small dimension and the stringent requirement of a 99 percent success probability. For uniform sampling with replacement we derive a potentially tighter condition number bound in terms of the leverage scores of Q. To obtain a more easily computable version of this bound, in terms of just the largest leverage scores, we first derive a general bound on the two-norm of diagonally scaled matrices. To facilitate the numerical experiments and test the tightness of the bounds, we present algorithms to generate matrices with user-specified coherence and leverage scores. These algorithms, the three sampling strategies, and a large variety of condition number bounds are implemented in the MATLAB toolbox *kappa_SQ*.

Key words. condition number, singular values, leverage scores, sums of random matrices, majorization, preconditioning, QR factorization

AMS subject classifications. 65F08, 65F10, 65F20, 65F25, 65F35, 68W20, 15A12, 15A18, 15A42, 15B10, 15B52

DOI. 10.1137/120870748

1. Introduction.

Our paper was inspired by Avron, Maymounkov, and Toledo's *Blendenpik* algorithm and analysis [1].

Blendenpik is an iterative method for solving overdetermined least squares/regression problems $\min_x \|Ax - b\|_2$ with the Krylov space method *LSQR* [21]. In order to accelerate convergence, *Blendenpik* constructs a preconditioner R_s and solves instead the preconditioned least squares problem $\min_z \|AR_s^{-1}z - b\|_2$. The solution to the original problem is recovered by solving a linear system with coefficient matrix R_s. The innovative feature is the construction of the preconditioner R_s by a random sampling method.

1.1. Motivation.

The purpose of our paper is a thorough experimental and analytical investigation of random sampling strategies for producing efficient preconditioners. The challenge is to ensure not only that R_s is nonsingular, but also that

*Received by the editors March 21, 2012; accepted for publication (in revised form) by M. Benzi July 24, 2014; published electronically December 11, 2014. The research of the authors was supported in part by NSF grant CCF-1145383.

http://www.siam.org/journals/simax/35-4/87074.html

†Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, http://www4.ncsu.edu/∼ipsen/, thomas_wentworth@ncsu.edu). The first author acknowledges the support from the XDATA Program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323. This material was based upon work partially supported by the National Science Foundation under grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute.

AR_s^{-1} is well-conditioned with respect to inversion, which is required for fast convergence and numerical stability.

Here is a conceptual point of view of how *Blendenpik* constructs the preconditioner: First it "smoothes out" the rows of A by applying a randomized unitary transform F, and then it uniformly samples (i.e., selects) a small number of rows M_s from FA. At last it computes a QR factorization of the smaller sampled matrix, $M_s = Q_s R_s$, where the triangular factor R_s serves as the preconditioner.

The neat and crucial observation in [1] is to realize that sampling rows from FA amounts, conceptually, to sampling rows from an orthonormal basis of FA. That is, if the columns of Q represent an orthonormal basis for the column space of FA, and if S is a sampling matrix then SQ has the same two-norm condition number as AR_s^{-1}. This means, it suffices to consider sampling from matrices Q with orthonormal columns.

The analysis in [1] suggests that SQ is well-conditioned, if Q has low "coherence." Intuitively, coherence gives information about the localization or "uniformity" of the elements of Q. Mathematically, coherence is the largest (squared) norm of any row of Q. For instance, if Q consists of canonical vectors, then the nonzero elements are concentrated in only a few rows, so that Q has high coherence. However, if Q is a submatrix of a Hadamard matrix, then all elements have the same magnitude, so that Q has low coherence.

If Q has low coherence then, in the context of sampling, all rows are equally important. Hence any sampled matrix SQ with sufficiently many rows is likely to have full rank. The purpose of the randomized transform F is to produce a matrix FA whose orthonormal basis Q has low coherence.

We were intrigued by the analysis of *Blendenpik* because it appears to be the first to exploit the concept of coherence for numerical purposes. We also wanted to get a better understanding of the condition number bound for SQ in [1, Theorem 3.2], which contains an unspecified constant, and of the effect of uniform sampling strategies.

1.2. Overview and main results. We survey the contents of the paper, with a focus on the main results.

From preconditioned matrices to sampled matrices with orthonormal columns (section 2). We start with a brief sketch of the *Blendenpik* least squares solver (section 2.1), and make the important transition from preconditioned matrices AR_s^{-1} to sampled matrices SQ with orthonormal columns, made possible by the observation ([1, 23] and Lemma 2.1) that both have the same two-norm condition number,[1]

$$\kappa(AR_s^{-1}) = \kappa(SQ).$$

Then we discuss the notion of coherence and its properties (section 2.2). For an $m \times n$ matrix Q with orthonormal columns, $Q^T Q = I_n$, the *coherence*

$$\mu \equiv \max_{1 \le j \le m} \|e_j^T Q\|_2^2$$

is the largest squared row norm.[2]

[1] Here $\kappa(X) \equiv \|X\|_2 \, \|X^\dagger\|_2$ denotes the Euclidean two-norm condition number with respect to inversion of a full-rank matrix X. The matrix $X^\dagger$ is the Moore–Penrose inverse of X.

[2] The superscript T denotes transpose, and I_n is the $n \times n$ identity matrix with columns e_j.

Sampling methods (section 3). We discuss three randomized methods for producing sampling matrices S: sampling without replacement (section 3.1), sampling with replacement (section 3.2), and Bernoulli sampling (section 3.3). We show that Bernoulli sampling can be viewed as a form of sampling without replacement (section 3.4).

The sampling matrices S from all three methods are constructed so that S^TS is an unbiased estimator of the identity matrix. The action of applying S to a matrix Q with orthonormal columns, SQ, amounts to randomly sampling rows from Q.

The numerical experiments (section 3.5) illustrate two points: First, the three sampling methods behave almost identically, in terms of the percentage of sampled matrices SQ that have full rank and their condition numbers, in particular, for small to moderate sampling amounts. Second, those sampled matrices SQ that have full rank tend to be very well-conditioned, with condition numbers $\kappa(SQ) \leq 10$.

As a consequence (section 3.6), we recommend sampling with replacement for *Blendenpik*, because it is fast, and it is easy to implement.

Numerical experiments. Since random sampling methods can be expected to work well in the asymptotic regime of very large matrix dimensions, we restrict all numerical experiments to matrices of small dimension.

Furthermore, we consider only matrices that have many more rows than columns, $m \gg n$. This is the situation where random sampling methods can be most efficient. In contrast, random sampling methods are not efficient for matrices that are almost square, because the number of rows in SQ has to be at least equal to n, otherwise $\operatorname{rank}(SQ) = n$ is not possible.

Condition number bounds based on coherence (section 4). We derive a probabilistic bound, in terms of coherence, for the condition numbers of the sampled matrices (Theorem 4.1 in section 4.1). The bound applies to all three sampling methods. From this we derive the following lower bound, not necessarily tight, on the required number of sampled rows.

PREVIEW OF COROLLARY 4.2. *Given a failure probability $0 < \delta < 1$, and a tolerance $0 \leq \epsilon < 1$, to achieve the condition number bound $\kappa(SQ) \leq \sqrt{\frac{1+\epsilon}{1-\epsilon}}$, the number of rows from Q, sampled by any of three methods, should be at least*

$$c \geq 3m\mu \, \frac{\ln(2n/\delta)}{\epsilon^2}. \tag{1.1}$$

This suggests that one has to sample more rows for SQ if Q has high coherence (μ close to 1), if one wants a low condition number bound (small ϵ), or if one wants a high success probability (small δ).

Numerical experiments (section 4.2) illustrate that the bounds are informative for matrices with sufficiently low coherence μ and sufficiently high aspect ratio m/n. Our bounds have the following advantages (section 4.3):

1. They are tighter than those in [1, Theorem 3.2] because they are nonasymptotic, with all constants explicitly specified.
2. They apply to three different sampling methods.
3. They imply a lower bound, of $\Omega(m\mu \ln n)$; on the required number of sampled rows.
4. They are realistic and informative—even for matrices of small dimension and the stringent requirement of a 99 percent success probability.

Condition number bounds based on leverage scores, for uniform sampling with replacement (section 5). The goal is to tighten the coherence-based bounds from section 4 by making use of all the row norms of Q, instead of just the largest one. To this end we introduce *leverage scores* (section 5.1), which are the squared row norms of Q,

$$\ell_j = \|e_j^T Q\|_2^2, \qquad 1 \le j \le m.$$

We use them to derive a bound for uniform sampling with replacement (Theorem 5.2 in section 5.2). Then we present a more easily computable bound, in terms of just a few of the largest leverage scores (section 5.3). It implies the following lower bound, not necessarily tight, on the number of samples.

PREVIEW OF COROLLARY 5.6. *Given a failure probability* $0 < \delta < 1$*, a tolerance* $0 \le \epsilon < 1$*, and a labeling of leverage scores in nonincreasing order,*

$$\mu = \ell_{[1]} \ge \cdots \ge \ell_{[m]}.$$

To achieve the condition number bound $\kappa(SQ) \le \sqrt{\frac{1+\epsilon}{1-\epsilon}}$*, the number of rows from* Q*, sampled uniformly with replacement, should be at least*

$$c \ge \tfrac{2}{3} m \, (3\tau + \epsilon\mu) \, \frac{\ln(2n/\delta)}{\epsilon^2}, \tag{1.2}$$

where $t \equiv \lfloor 1/\mu \rfloor$ *and* $\tau \equiv \mu \, \sum_{j=1}^{t} \ell_{[j]} + (1 - t\,\mu)\, \ell_{[t+1]}$.

We show (section 5.4) that (1.2) is indeed tighter than (1.1). This is confirmed by numerical experiments (section 5.5). The difference becomes more drastic for matrices Q with widely varying nonzero leverage scores, and can be as high as ten percent. Hence (section 5.6), when it comes to lower bounds for the number of rows sampled uniformly with replacement, we recommend (1.2) over (1.1).

Algorithms for generating matrices with prescribed coherence and leverage scores (section 6). The purpose is to make it easy to investigate the efficiency of the sampling methods in section 3, and test the tightness of the bounds in sections 4 and 5.

To this end we present algorithms for generating matrices with prescribed leverage scores and coherence (section 6.1), and for generating particular leverage score distributions with prescribed coherence (section 6.2). Furthermore we present two classes of structured matrices with prescribed coherence that are easy and fast to generate (section 6.3). The basis for the algorithms is the following majorization result.

PREVIEW OF THEOREM C.2. *Given integers* $m \ge n$ *and a vector* ℓ *with* m *elements that satisfy* $0 \le \ell_j \le 1$ *and* $\sum_{j=1}^{m} \ell_j = n$*, there exists an* $m \times n$ *matrix* Q *with orthonormal columns that has leverage scores* $\|e_j^T Q\|_2^2 = \ell_j$*,* $1 \le j \le m$*, and coherence* $\mu = \max_{1 \le j \le m} \ell_j$.

Bound for two-norms of diagonally scaled matrices (section B). The bound (1.2) is based on a special case of the following general bound for the two-norm of diagonally scaled matrices.

PREVIEW OF THEOREM B.4. *Let* Z *be an* $m \times n$ *matrix with* $\operatorname{rank}(Z) = n$ *and largest squared row norm* $\mu_z \equiv \max_{1 \le j \le m} \|e_j^T Z\|_2^2$*. Let* D *be an* $m \times m$ *nonnegative diagonal matrix, and a labeling of diagonal elements in nonincreasing order,*

$$\|D\|_2 = d_{[1]} \ge \cdots \ge d_{[m]} \ge 0.$$

If $t \equiv \lfloor (\|Z^\dagger\|_2^2 \, \mu_z)^{-1} \rfloor$, *then either*

$$\|DZ\|_2^2 \leq \mu_z \sum_{j=1}^{t} d_{[j]}^2 + \left(\|Z\|_2^2 - t \, \mu_z \right) \, d_{[t+1]}^2 \qquad \text{if } \|Z\|_2^2 - t \, \mu_z \leq \mu_z$$

or

$$\|DZ\|_2^2 \leq \mu_z \sum_{j=2}^{t+1} d_{[j]}^2 + \left(\|Z\|_2^2 - t \, \mu_z \right) \, d_{[1]}^2 \qquad \text{if } \|Z\|_2^2 - t \, \mu_z > \mu_z.$$

MATLAB toolbox. In order to perform the experiments in this paper, we developed a MATLAB toolbox `kappaSQ` with a user-friendly interface [29]. The toolbox contains implementations of the three random sampling methods in section 3, the matrix generation algorithms in section 6, the bounds in sections 4 and 5, and a variety of other condition number bounds. It also allows the user to input her/his own matrices.

Proofs (sections A, B, and C). All proofs, except those for sections 2 and 3, have been relegated to these three sections, which form the appendix.

Section A contains the proofs for sections 4 and 5, which are based on two matrix concentration inequalities: A Chernoff bound (section A.1) and a Bernstein bound (section A.4).

Section B contains the proofs for the easily computable bounds in sections 5.3 and 5.4, together with the majorization results (section B.1) required for the proofs.

The majorization results in section C represent the foundation for the algorithms in section 6.

Future work (section 7). We list a few issues that suggest themselves immediately as a follow up to this paper.

1.3. Literature. Existing randomized least squares methods are based on randomized projections. This means, conceptually they multiply A by a random matrix F, and then sample a few rows from FA.

The algorithms in [4, 10, 11] solve a smaller sampled problem by a direct method. Like *Blendenpik* [1], the algorithm in [23] computes a preconditioner from the QR factorization of a sampled submatrix, but then solves the preconditioned problem by applying the conjugate gradient method to the normal equations. The parallel solver *LSRN* [19] computes a preconditioner from the SVD of a sampled submatrix, and then solves the preconditioned problem with an iterative method. This solver applies to general matrices rather than just those of full column rank.

As for randomized algorithms in general, the excellent surveys [14, 18] provide clear analyses and good intuition.

1.4. Notation. The norm $\|\cdot\|_2$ denotes the Euclidean two-norm, and the two-norm *condition number* with respect to inversion of a real $m \times n$ matrix Z with $\text{rank}(Z) = n$ is denoted by $\kappa(Z) \equiv \|Z\|_2 \|Z^\dagger\|_2$, where $Z^\dagger$ is the Moore–Penrose inverse. The $k \times k$ *identity matrix* is $I_k = \begin{pmatrix} e_1 & \dots & e_k \end{pmatrix}$, and its columns are the *canonical vectors* e_j, $1 \leq j \leq k$.

The *probability* of an event $\mathcal{X}$ is denoted by $\mathbf{Pr}[\mathcal{X}]$, and the *expected value* of a random variable X is denoted by $\mathbf{E}[X]$.

2. The Blendenpik algorithm, and coherence. We describe the *Blendenpik* algorithm for solving least squares problems (section 2.1), and present the notion of coherence (section 2.2).

2.1. Algorithm. The *Blendenpik* algorithm [1, Algorithm 1] solves full column rank least squares problems with the Krylov space method *LSQR* [21] and a randomized preconditioner. Algorithm 2.1 presents a conceptual sketch of *Blendenpik*. The subscript "s" denotes quantities associated with the sampled matrix.

ALGORITHM 2.1. SKETCH OF *Blendenpik* [1].

Input: $m \times n$ matrix A with $m \geq n$ and $\operatorname{rank}(A) = n$, $m \times 1$ vector b
$m \times m$ random unitary matrix F
$k \times n$ sampling matrix S with $k \geq n$
Output: Solution of $\min_x \|Ax - b\|_2$

$M = FA$ {Improve coherence}
$M_s = SM$ {Sample for preconditioner}
Thin QR factorization $M_s = Q_s R_s$ {Generate preconditioner}
Determine solution y to $\min_z \|AR_s^{-1}z - b\|_2$ {Solve preconditioned problem}
Solve $R_s\hat{x} = y$ {Recover solution to original problem}

The matrix F is the product of a random diagonal matrix with ± 1 entries, and a unitary transform, such as a Walsh–Hadamard transform, or a discrete Fourier, Hartley, or cosine transform [1, section 3.2]. The transformed matrix $M = FA$ is $m \times n$ with $m \geq n$ and $\operatorname{rank}(M) = n$.

The sampling matrix S selects $k \geq n$ rows from the transformed matrix M. We discuss different types of sampling matrices in section 3. The $k \times n$ sampled matrix M_s has a thin QR decomposition $M_s = Q_s R_s$, where Q_s is $k \times n$ with orthonormal columns and R_s is $n \times n$ upper triangular.

The basis for the analysis is the thin QR decomposition $M = QR$, where Q is $m \times n$ with orthonormal columns and R is $n \times n$ upper triangular. This QR decomposition is *not* computed. The next result links the condition number of the preconditioned matrix to that of the matrix SQ; see, also, [1, section 3.1] and [23, Theorem 1].

LEMMA 2.1. *With the notation in Algorithm* 2.1, *if* $\operatorname{rank}(M_s) = n$, *then*

$$\kappa(AR_s^{-1}) = \kappa(SQ).$$

Proof. From $FA = M = QR$ and the fact that the two-norm is invariant under premultiplication by matrices with orthonormal columns, it follows that

$$\begin{aligned}\kappa(AR_s^{-1}) &= \kappa(MR_s^{-1}) = \kappa(RR_s^{-1}) = \kappa(R_s R^{-1}) = \kappa(M_s R^{-1}) = \kappa(SMR^{-1}) \\ &= \kappa(SQ). \qquad \square\end{aligned}$$

In sections 4 and 5 we derive bounds for the condition number of the preconditioned matrix, $\kappa(AR_s^{-1})$. Our bounds are tighter than those in [1, Theorem 3.2], because they have all constants explicitly specified, and apply to three different sampling strategies. Since Lemma 2.1 implies $\kappa(AR_s^{-1}) = \kappa(SQ)$, we state the bounds for $\kappa(SQ)$ only. An important ingredient in these bounds is the coherence of Q.

2.2. Coherence. Coherence gives information about the localization or "uniformity" of the elements in an orthonormal basis. The more general concept of *mutual coherence* between two orthonormal bases was introduced in [8, section VII], in the context of signal processing and computational harmonic analysis, to describe a condition for the existence of sparse representations of signals. What we use here is a special case, and can be viewed as a measure for how close an orthonormal basis is to sharing a vector with a canonical basis.

DEFINITION 2.2 (Definition 3.1 in [1], Definition 1.2 in [5]). *Let Q be a real $m \times n$ matrix with orthonormal columns, $Q^T Q = I_n$, then the* coherence *of Q is*

$$\mu \equiv \max_{1 \le j \le m} \|e_j^T Q\|_2^2.$$

If the columns of Q are an orthonormal basis for the column space of a matrix M, then the coherence of M is μ.

The second part of Definition 2.2 emphasizes that coherence is really a property of the column space, hence basis independent. In other words, if $\hat{Q} = QV$, where V is a real $n \times n$ orthogonal matrix, then $\hat{Q}$ and Q have the same coherence.

The range for coherence is $\frac{n}{m} \le \mu \le 1$. If Q is an $m \times n$ submatrix of the $m \times m$ Hadamard matrix, then $\mu = n/m$. If a column of Q is a canonical vector, then $\mu = 1$. Hence an orthonormal basis has high coherence if it shares a vector with a canonical basis.

There are other definitions of coherence that differ from the above by factors depending on the matrix dimensions [22, Definition 1], [25, Definition 1]. However, the notion of *statistical coherence* in Bayesian analysis [17] appears to be unrelated.

3. Sampling methods. We present three different types of sampling methods: sampling without replacement (section 3.1), sampling with replacement (section 3.2), and Bernoulli sampling (section 3.3). We show that Bernoulli sampling can be viewed as a form of sampling without replacement (section 3.4). The numerical experiments illustrate that there is little difference among the three methods for small to moderate amounts of sampling (section 3.5). Hence we recommend sampling with replacement for Algorithm 2.1 (section 3.6).

The sampling matrices S in all three methods are scaled so that $S^T S$ is an unbiased estimator of the identity matrix.

3.1. Sampling without replacement. The obvious sampling strategy, in Algorithm 3.1, picks the requested number of rows, so that the sampling matrix S is just a scaled submatrix of a permutation matrix.

Uniform sampling without replacement can be implemented via *random permutations*.[3] A permutation $\pi_1, \ldots, \pi_m$ of the integers $1, \ldots, m$ is a *random permutation*, if it is equally likely to be one of $m!$ possible permutations [20, pages 41 and 48].

ALGORITHM 3.1. UNIFORM SAMPLING WITHOUT REPLACEMENT [12, 13].

Input: Integers $m \ge 1$ and $1 \le c \le m$
Output: $c \times m$ sampling matrix S with $\mathbf{E}[S^T S] = I_m$

Let $k_1, \ldots, k_m$ be a random permutation of $1, \ldots, m$
$S = \sqrt{\frac{m}{c}} \begin{pmatrix} e_{k_1} & \ldots & e_{k_c} \end{pmatrix}^T$

The following lemma presents the probability that sampling without replacement picks a particular row.

LEMMA 3.1. *If Algorithm* 3.1 *samples c out of m indices, then the probability that a particular index is picked equals c/m.*

Proof. The probability that some index, say r, is not sampled in the first trial is $1 - \frac{1}{m} = \frac{m-1}{m}$. Now there are only $m-1$ indices left. So the probability that index r is not sampled in the second trial is $1 - \frac{1}{m-1} = \frac{m-2}{m-1}$. Repeating this argument shows that with probability $\prod_{t=1}^{c} \frac{m-t}{m-t+1} = \frac{m-c}{m}$, index r is not sampled in c trials.

[3] We thank an anonymous reviewer for this advice.

The complementary event, the probability that index r is sampled, equals $1 - \frac{m-c}{m} = \frac{c}{m}$. ∎

3.2. Sampling with replacement. This is the sampling strategy that appears to be analyzed in [1]. It samples exactly the requested number of rows, but with replacement, which means a row may be sampled more than once. Algorithm 3.2 is the same as the *EXACTLY(c)* algorithm [11, Algorithm 3] with uniform probabilities, which is also used in the *BasicMatrixMultiplication Algorithm* [9, Fig. 2].

ALGORITHM 3.2. UNIFORM SAMPLING WITH REPLACEMENT [9, 11].

Input: Integers $m \geq 1$ and $1 \leq c \leq m$
Output: $c \times m$ sampling matrix S with $\mathbf{E}[S^T S] = I_m$

for $t = 1 : c$ **do**
  Sample k_t from $\{1, \ldots, m\}$ with probability $1/m$,
  independently and with replacement
end for
$S = \sqrt{\frac{m}{c}} \begin{pmatrix} e_{k_1} & \ldots & e_{k_c} \end{pmatrix}^T$

Sampling with replacement (Algorithm 3.2) is often easier to analyze and implement than sampling without replacement (Algorithm 3.1), and it can also be more robust to errors [20, section 1.2].

3.3. Bernoulli sampling. The sampling strategy in Algorithm 3.3 is implemented in *Blendenpik* [1, Algorithm 1]. Following [13, section A], we use the term *Bernoulli sampling*, because the strategy treats each row as an independent, identically distributed Bernoulli random variable. Each row is either sampled or not, with the same probability for each row. Algorithm 3.3 produces an $m \times m$ square matrix S—in contrast to Algorithms 3.1 and 3.2, which produce $c \times m$ matrices.

ALGORITHM 3.3. BERNOULLI SAMPLING [1, 12, 13].

Input: Integers $m \geq 1$ and $1 \leq c \leq m$
Output: $m \times m$ sampling matrix S with $\mathbf{E}[S^T S] = I_m$

$S = 0_{m \times m}$
for $t = 1 : m$ **do**
$$S_{tt} = \sqrt{\frac{m}{c}} \begin{cases} 1 & \text{with probability } \frac{c}{m} \\ 0 & \text{with probability } 1 - \frac{c}{m} \end{cases}$$
end for

The number of sampled rows, which is equal to the number of nonzero diagonal elements in S, is not known a priori, but the expected number of sampled rows is c. The lemma below shows that the actual number of rows picked by Bernoulli sampling is characterized by a binomial distribution [24, section 2.2.2].

LEMMA 3.2. *If Algorithm* 3.3 *samples from m indices with probability $\gamma \equiv c/m$, then the probability that it picks exactly k indices equals $\binom{m}{k} \gamma^k (1-\gamma)^{m-k}$.*

Proof. Determining the diagonal elements of the $m \times m$ sampling matrix S in Algorithm 3.3 can be viewed as performing m independent trials, where trial t is a success ($S_{tt} \neq 0$) with probability γ, and a failure ($S_{tt} = 0$) with probab-

ility $1-\gamma$. The probability of k successes is given by the binomial distribution $\binom{m}{k}\gamma^k\,(1-\gamma)^{m-k}$. ∎

3.4. Relating Bernoulli sampling and sampling without replacement. We show that Bernoulli sampling (Algorithm 3.3) is the same as first determining the number of samples with a binomial distribution (motivated by Lemma 3.2), and then sampling without replacement (Algorithm 3.1). This is described in Algorithm 3.4 below.

ALGORITHM 3.4. SIMULATING ALGORITHM 3.3 WITH ALGORITHM 3.1.

Input: Integers $m \geq 1$ and $1 \leq c \leq m$

Output: $\tilde{c} \times m$ sampling matrix S with $\mathbf{E}[S^T S] = I_m$ that "behaves like" a sampling matrix generated by Algorithm 3.3

$\gamma \equiv c/m$

Sample $\tilde{c}$ from $\{1,\ldots,m\}$ where $\mathbf{Pr}[\tilde{c}=k] = \binom{m}{k}\gamma^k\,(1-\gamma)^{m-k}$

Use Algorithm 3.1 to sample $\tilde{c}$ indices $k_1,\ldots,k_{\tilde{c}}$ uniformly and without replacement

$S = \sqrt{\frac{m}{\tilde{c}}}\,\begin{pmatrix} e_{k_1} & \ldots & e_{k_{\tilde{c}}}\end{pmatrix}^T$

Below we describe the sense in which Algorithm 3.4 "behaves like" Bernoulli sampling in Algorithm 3.3.

LEMMA 3.3. *The probability that Algorithm* 3.4 *picks a particular index equals* $\gamma = c/m$.

Proof. Motivated by Lemma 3.2, the actual number of samples k in Algorithm 3.4 is given by a binomial distribution. Once a specific k has emerged, one applies Lemma 3.1 to conclude that the probability that Algorithm 3.1 picks some index r is k/m.

Now the probability that Algorithm 3.4 picks some index r is obtained by conditioning [24, section 3.5] on the number of samples, k, and equals

$$\begin{aligned}
&\sum_{k=0}^{m} \mathbf{Pr}\left[k \text{ indices sampled}\right]\ \mathbf{Pr}\left[\text{index } r \text{ sampled}|\ k \text{ indices sampled}\right] \\
&\quad = \sum_{k=1}^{m} \binom{m}{k}\gamma^k\ (1-\gamma)^{m-k}\ \frac{k}{m} \\
&\quad = \gamma \sum_{k=0}^{m-1} \binom{m-1}{k}\gamma^k\,(1-\gamma)^{m-1-k} = \gamma\,(\gamma+(1-\gamma))^{m-1} = \gamma,
\end{aligned}$$

where the first equality follows from the zero summand for $k=0$. ∎

Finally, we can conclude that sampling with Algorithm 3.4 is the same as sampling with Algorithm 3.3.

THEOREM 3.4. *Both, Algorithms* 3.4 *and* 3.3 *pick a particular set of indices* $i_1,\ldots,i_c$ *with probability* $\gamma^c(1-\gamma)^{m-c}$.

Proof. The probability that Algorithm 3.3 samples indices $i_1,\ldots,i_c$ is equal to $\gamma^c(1-\gamma)^{m-c}$.

We show that the same is true for Algorithm 3.4. The choice of the sampling distribution in Algorithm 3.4 implies that it samples $\tilde{c}=c$ indices with probability $\binom{m}{c}\gamma^c(1-\gamma)^{m-c}$. Since there are $\binom{m}{c}$ ways to sample c out of m indices, the probability

that the particular index set $i_1, \ldots, i_c$ is picked, given that c indices are being sampled, is $1/\binom{m}{c}$. Thus, the probability that Algorithm 3.4 picks indices $i_1, \ldots, i_c$ equals

$$\frac{1}{\binom{m}{c}} \binom{m}{c} \gamma^c (1-\gamma)^{m-c} = \gamma^c (1-\gamma)^{m-c}. \qquad \square$$

3.5. Numerical experiments. We present two representative comparisons of the three sampling strategies, with two plots for each strategy: the condition numbers of full-rank sampled matrices SQ, and the failure percentage, that is the percentage of sampled matrices SQ that are numerically rank deficient (as determined by the MATLAB command `rank`).

The experiments are limited to very tall and skinny matrices (with many more rows than columns, $m \gg n$), because that's when the sampling strategies are most efficient. In particular, since $c \geq n$ is required for SQ to have full column rank, sampling methods are inefficient when n is not much smaller than m, in which case a deterministic algorithm would be preferable.

Experimental setup. The $m \times n$ matrices Q with orthonormal columns have $m = 10^4$ rows and $n = 5$ columns. The condition numbers and failure percentages are plotted against various sampling amounts c, with 30 runs for each c. For the failure percentages we display only those sampling amounts c that give rise to rank deficient matrices, in these particular 30 runs. For Algorithm 3.3 the horizontal axis represents the numerator c in the probability, that is, the expected number of sampled rows. All three strategies sample from the same matrix.

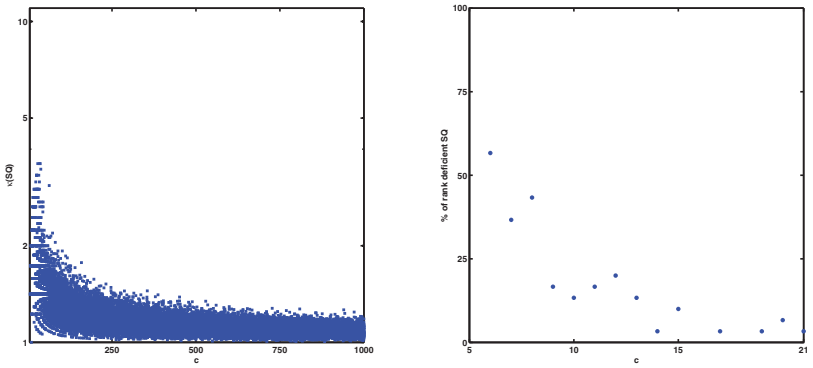
We consider two different types of matrices: matrices with low coherence $\mu = 1.5n/m$ in Figure 1, and matrices with higher coherence $\mu = 150n/m$ and many zero rows in Figure 2. Our numerical experiments indicate that these coherence values are representative, in the sense that different values of coherence would not produce any other interesting effects.

Figure 1. Shown are condition numbers and percentage of rank deficient matrices for a matrix Q with low coherence $\mu = 1.5n/m$ generated by Algorithm 6.2. At most 10 percent of the rows are sampled. The three strategies exhibit almost identical behavior: The sampled matrices SQ of full rank are very well-conditioned, with $\kappa(SQ) \leq 5$. Numerically rank deficient matrices SQ occur only for sampling amounts $c \leq 47$.

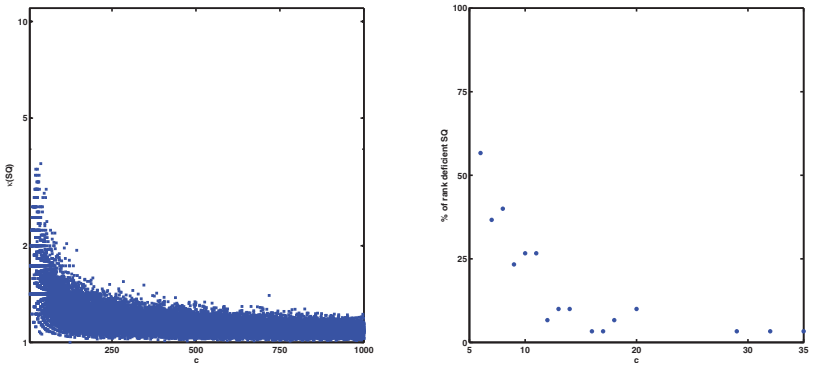
Figure 2. Shown are condition numbers and percentage of rank deficient matrices for a matrix Q, generated by Algorithm 6.3, with coherence $150n/m$ and many zero rows. The number of sampled rows ranges from $c = 4{,}000$ to m. The sampled matrices SQ of full rank are very well-conditioned, with $\kappa(SQ) \leq 10$. Even for $c = 4{,}000$, as many as 10 percent of the sampled matrices can still be rank deficient. All three algorithms have to sample more than half of the rows of Q in order to always produce matrices SQ with full column rank. Specifically in these particular runs, Algorithms 3.1 and 3.3 need to sample $c \geq 5{,}222$ and $c \geq 5{,}301$ rows, respectively, while Algorithm 3.1 needs $c \geq 7{,}732$.

Note that the condition numbers of matrices from Algorithms 3.1 and 3.3 approach 1 as more and more rows are sampled. This is because no row is sampled more than once, and for $c = m$ all rows are sampled.

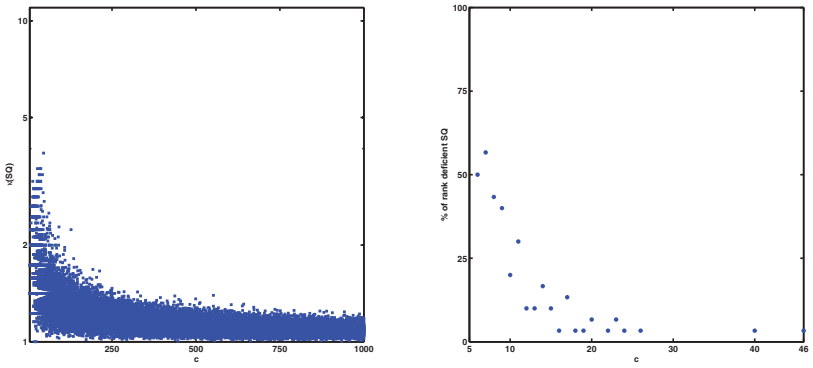
Again, the three strategies exhibit almost identical behavior: The sampled matrices SQ of full rank are very well-conditioned with $\kappa(SQ) \leq 10$. However, due to the higher coherence, numerically rank deficient matrices occur more frequently.



(a) Algorithm 3.1: Sampling without replacement



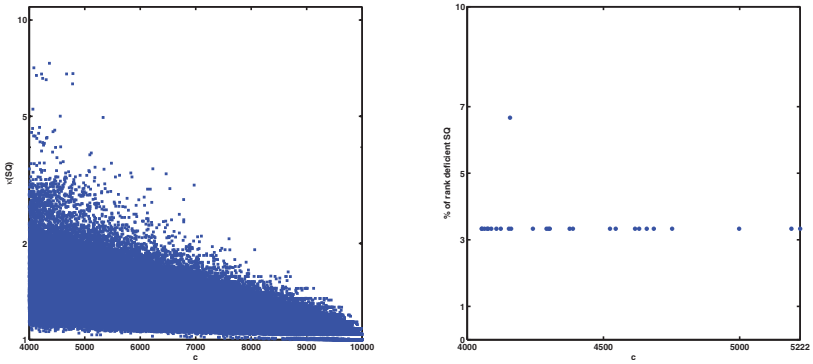
(b) Algorithm 3.2: Sampling with replacement



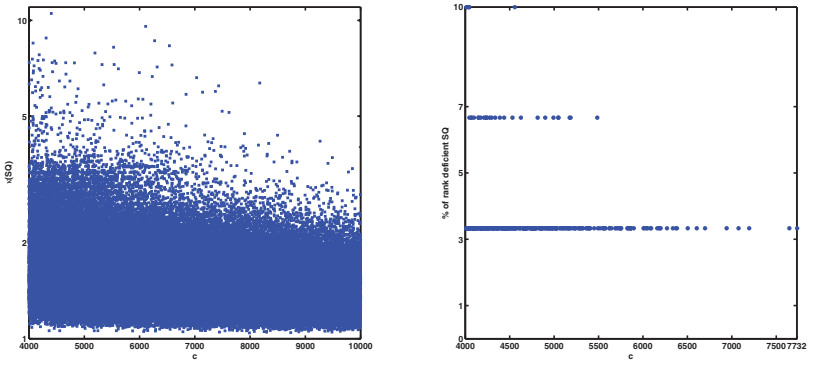
(c) Algorithm 3.3: Bernoulli sampling

FIG. 1. *Condition numbers and percentage of rank deficiency for matrices with low coherence and small amounts of sampling. Here Q is $m \times n$ with orthonormal columns, $m = 10{,}000$, $n = 5$, coherence $\mu = 1.5n/m$, and is generated with Algorithm* 6.2. *Left panels: Horizontal coordinate axes represent amounts of sampling $n \leq c \leq 1{,}000$. Vertical coordinate axes represent condition numbers $\kappa(SQ)$; the maximum is* 10. *Right panels: Horizontal coordinate axes represent amounts of sampling that give rise to numerically rank deficient matrices SQ. Vertical coordinate axes represent percentage of numerically rank deficient matrices.*

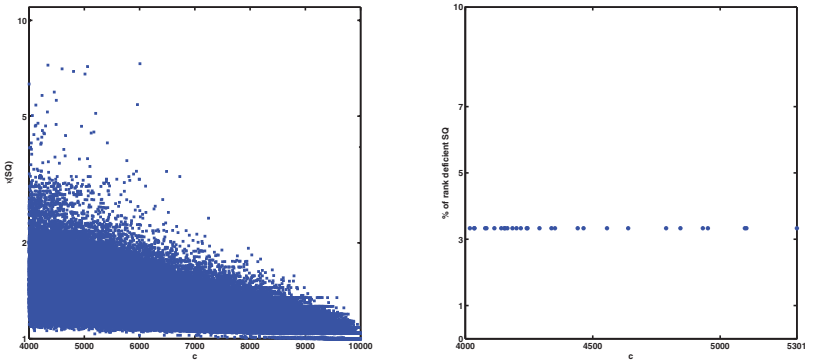
3.6. Conclusions for section 3. The numerical experiments illustrate that the three sampling strategies behave almost identically, in particular, for small to moderate sampling amounts, and that sampled matrices of full rank tend to be very



(a) Algorithm 3.1: Sampling without replacement



(b) Algorithm 3.2: Sampling with replacement



(c) Algorithm 3.3: Bernoulli sampling

FIG. 2. *Condition numbers and percentage of rank deficiency for matrices with higher coherence and large amounts of sampling. Here Q is $m \times n$ with orthonormal columns, $m = 10{,}000$, $n = 5$, coherence $\mu = 150n/m$, and is generated with Algorithm* 6.3. *Left panels: Horizontal coordinate axes represent amounts of sampling $4{,}000 \leq c \leq m$. Vertical coordinate axes represent condition numbers $\kappa(SQ)$; the maximum is* 10. *Right panels: Horizontal coordinate axes represent amounts of sampling that give rise to numerically rank deficient matrices SQ. Vertical coordinate axes represent percentage of numerically rank deficient matrices SQ; the maximum is* 10 *percent.*

well-conditioned.[4] Furthermore, section 3.4 shows that Bernoulli sampling can be viewed as a form of sampling without replacement, and the numerical experiments confirm the similarity in behavior.

[4]We have not been able to show rigorously why the condition numbers tend to be less than 10.

Among the three strategies, we recommend sampling with replacement (Algorithm 3.2) for small to moderate amounts of sampling in Algorithm 2.1. It is fast and easy to implement in both.

4. Condition number bounds based on coherence. We derive bounds for the condition numbers of matrices produced by the sampling strategies in section 3, in terms of coherence. These bounds are based on a specific concentration inequality and imply a, not necessarily tight, lower bound for the number of sampled rows (section 4.1). Numerical experiments illustrate that the bounds are informative (section 4.2). We end this section by summarizing the main features of the bounds (section 4.3).

4.1. Bounds. We show that the three sampling strategies in section 3 all have the same condition number bound, in terms of coherence.

Theorem 4.1 below is based on a matrix Chernoff concentration inequality (section A.1). We chose this particular inequality because extensive numerical experiments with our MATLAB toolbox `kappaSQ` [29] suggest that it tends to produce the tightest bound.

THEOREM 4.1. *Let Q be a real $m \times n$ matrix with $Q^TQ = I_n$ and coherence μ. Let S be a sampling matrix produced by Algorithm* 3.1, 3.2, *or* 3.3 *with $n \leq c \leq m$. For $0 < \epsilon < 1$ and $f(x) \equiv e^x(1+x)^{-(1+x)}$ define*

$$\delta \equiv n \left(f(-\epsilon)^{c/(m\mu)} + f(\epsilon)^{c/(m\mu)} \right).$$

If $\delta < 1$, then with probability at least $1 - \delta$ we have $\operatorname{rank}(SQ) = n$ *and*

$$\kappa(SQ) \leq \sqrt{\frac{1+\epsilon}{1-\epsilon}}.$$

Proof. The proof is based on results from [12, 26, 27] and is relegated to section A.2. ∎

Since $0 < f(\pm\epsilon) < 1$ for $0 < \epsilon < 1$, Theorem 4.1 implies that the sampling strategies in section 3 are more likely to produce full-rank matrices as the number c of sampled rows increases. Furthermore, for a given total number of rows m, matrices Q with fewer columns n and lower coherence μ are more likely to give rise to sampled matrices SQ that have full rank.

Theorem 4.1 implies the following lower bound on the number of samples, but we make no claims about the tightness of this bound.

COROLLARY 4.2. *Under the assumptions of Theorem* 4.1,

$$c \geq 3m\mu \, \frac{\ln(2n/\delta)}{\epsilon^2}$$

samples are sufficient to achieve $\kappa(SQ) \leq \sqrt{\frac{1+\epsilon}{1-\epsilon}}$ with probability at least $1 - \delta$.

Proof. See section A.3. ∎

Corollary 4.2 implies that the sampling strategies in section 3 should sample at least $c = \Omega(m\mu \ln n)$ rows to produce a full-rank, well-conditioned matrix. In particular, if Q has minimal coherence $\mu = n/m$, then Corollary 4.2 implies that the number of sampled rows should be at least

$$c \geq 3n \, \frac{\ln(2n/\delta)}{\epsilon^2}, \tag{4.1}$$

that is, $c = \Omega(n \ln n)$.

To achieve $\kappa(SQ) \leq 10$ with probability at least .99 requires that the number of sampled rows be at least

$$c \geq 3.2\, m\mu\, (\ln(2n) + 4.7)\,. \tag{4.2}$$

Here we chose $\epsilon_0 = 99/101$, so that the condition number bound equals $\sqrt{\frac{1+\epsilon_0}{1-\epsilon_0}} = 10$.

Remark 4.3. Theorem 4.1 is informative only for sufficiently low coherence values.

For instance, consider the higher coherence matrices from Figure 2 in section 3.5 with $m = 10{,}000$, $n = 5$, and coherence $\mu = 150n/m$. Choose $\epsilon = 99/101$ so that $\kappa(SQ) \leq 10$, and a failure probability $\delta = .01$. Then Corollary 4.2 implies the lower bound $c \geq 12{,}408$, which means that the number of sampled rows would have to be larger than the total number of rows.

4.2. Numerical experiments. We compare the bound for the condition numbers of the sampled matrices (Theorem 4.1) with the true condition numbers of matrices produced by sampling with replacement (Algorithm 3.2).

There are several reasons why it suffices to consider only a single sampling strategy: The three sampling methods all have the same bound (Theorem 4.1); Bernoulli sampling is a form of sampling without replacement (section 3.4); and all three sampling methods exhibit very similar behavior for matrices of low coherence (sections 3.5 and 3.6). Furthermore, this allows a clean comparison with the bounds in section 5 which apply only to Algorithm 3.2.

Experimental setup. The $m \times n$ matrices Q with orthonormal columns have $m = 10^4$ rows and $n = 5$ columns. The left panels in Figure 3 show the condition numbers of the full-rank sampled matrices SQ produced by Algorithm 3.2 against different sampling amounts c, with 30 runs for each c. The right panels in Figure 3 show the percentage of rank deficient matrices SQ against different sampling amounts c. We display only those sampling amounts c that give rise to rank deficient matrices in these particular 30 runs.

The left panels in Figure 3 also show the condition number bound $\kappa_\epsilon \equiv \sqrt{\frac{1+\epsilon}{1-\epsilon}}$ from Theorem 4.1. For each value of c, we obtain ϵ as the solution of the nonlinear equation $F_c(x)^2 = 0$ associated with Theorem 4.1 and defined as

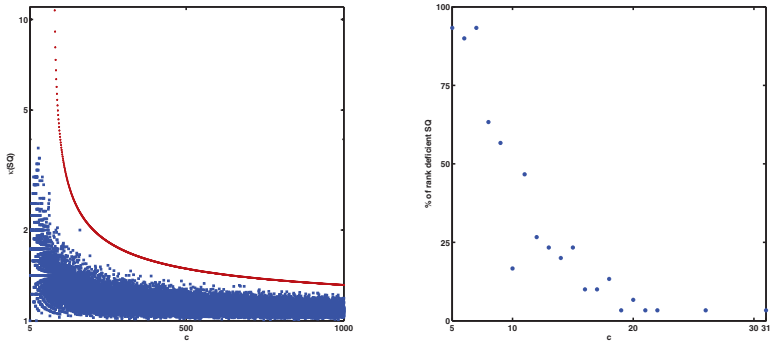
$$F_c(x) \equiv \delta - n\left(f(-x)^{c/(m\mu)} + f(x)^{c/(m\mu)}\right).$$

We impose the stringent requirement of $\delta = .01$, corresponding to a 99 percent success probability. Since an explicit expression seems out of reach, we use unconstrained nonlinear optimization (a Nelder–Mead simplex direct search) to solve $F_c(x)^2 = 0$. This is done in MATLAB with a code equivalent to

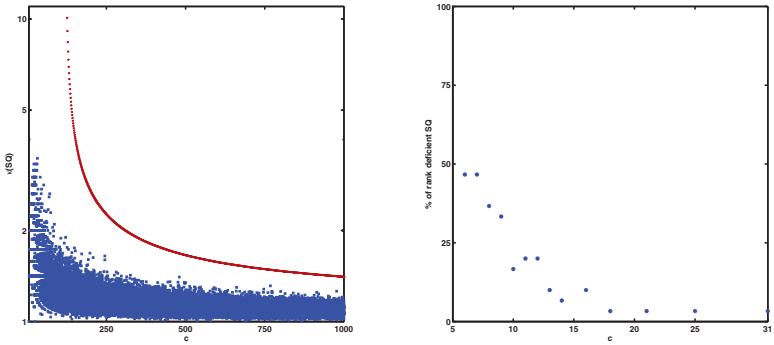
$$\epsilon = \left|\texttt{fminsearch}(F_c(x)^2, 0, 10^{-30})\right|,$$

where `fminsearch` starts at the point 0, and terminates when $|F_c(\epsilon)|^2 \leq 10^{-30}$. If $0 < \epsilon < 1$ then κ_ϵ is plotted, otherwise nothing is plotted.

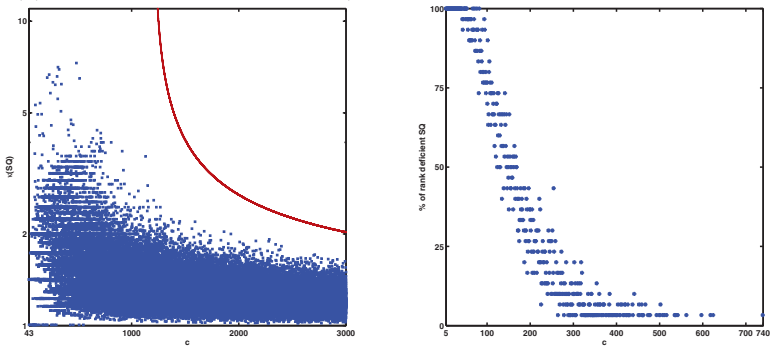
As explained in Remark 4.3, Theorem 4.1 is not informative for higher coherence values, so we consider matrices with the following properties: Minimal coherence $\mu = n/m$ in Figure 3(a); low coherence $\mu = 1.5n/m$ in Figure 3(b); slightly higher coherence $\mu = 15n/m$ with many zero rows in Figure 3(c). The matrices for Figures 3(a) and 3(b) were generated with Algorithm 6.2, while the matrix for Figure 3(c) was generated with Algorithm 6.3.



(a) Q has minimal coherence $\mu = n/m$. Sampling amounts are $n \leq c \leq 1{,}000$.



(b) Q has low coherence $\mu = 1.5n/m$. Sampling amounts are $n \leq c \leq 1{,}000$.



(c) Q has slightly higher coherence $\mu = 15n/m$ and many zero rows. Sampling amounts are $n \leq c \leq 3{,}000$.

FIG. 3. *Condition numbers and bound from Theorem* 4.1, *and percentage of rank deficiency. Here* Q *is* $m \times n$ *with orthonormal columns,* $m = 10{,}000$, *and* $n = 5$. *Left panels: The horizontal coordinate axes represent amounts of sampling* c. *The vertical coordinate axes represent condition numbers* $\kappa(SQ)$; *the maximum is* 10. *The dots at the bottom represent the condition numbers of matrices sampled with Algorithm* 3.2, *while the upper line represents the bound from Theorem* 4.1. *Right panels: The horizontal coordinate axes represent amounts of sampling that produce numerically rank deficient matrices* SQ. *The vertical coordinate axes represent the percentage of numerically rank deficient matrices* SQ.

Figure 3. The left panels illustrate that Theorem 4.1, constrained to a 99 percent success probability, correctly predicts the magnitude of the condition numbers, i.e., $\kappa(SQ) \leq 10$. Hence Theorem 4.1 provides informative qualitative bounds for matrices with very low coherence, as well as for matrices with slightly higher coherence and many zero rows.

TABLE 1
Comparison of information from Figure 3, *with Theorem* 4.1 *and Corollary* 4.2.

Figure	Coherence μ	Last rank deficiency occurs at $c =$	Theorem 4.1 starts at $c =$	(4.2)
3(a)	n/m	31	81	83
3(b)	$1.5\,n/m$	31	121	125
3(c)	$15\,n/m$	740	1,207	1,241

Table 1. This is a comparison of the numerical experiments in Figure 3 with the bounds from Theorem 4.1 and Corollary 4.2, both restricted to a 99 percent success probability.

The third column depicts the highest values of c for which a rank deficient matrix occurs, during these particular 30 runs. It should be kept in mind that these values are highly dependent on the particular sampling runs. This column is to be compared to the fourth column which contains the lowest values of c where Theorem 4.1 starts to apply. Although there is a gap between the occurrence of the last rank deficiency and the onset of Theorem 4.1, the values have qualitatively the same order of magnitude.

The rightmost column in Table 1 contains the values of the lower bound (4.2), and is to be compared to the column with the starting values for Theorem 4.1. Although (4.2) is weaker than Theorem 4.1, its values are close to the starting values of Theorem 4.1, especially for lower coherence. Hence, the lower bound (4.2) captures the correct magnitude of the sampling amounts where Theorem 4.1 starts to become informative.

Table 1 illustrates that, although Theorem 4.1 and Corollary 4.2 tend to become more pessimistic with increasing coherence, they still provide qualitative information for matrices with low coherence—even when restricted to a 99 percent success probability.

4.3. Conclusions for section 4. The bounds in Theorem 4.1 and Corollary 4.2 have the following advantages:

1. They are nonasymptotic bounds, where all constants have explicit numerical values, hence they are tighter than the bounds in [1, Theorem 3.2].
2. They apply to three different sampling methods.
3. They imply a lower bound, of $\Omega\left(m\mu \ln n\right)$, on the required number of sampled rows. Although we did not give a formal proof of tightness, numerical experiments illustrate that sampling only the required number of rows implied by the bound is realistic. Numerical experiments illustrate that the bound is realistic.
4. Even under the stringent requirement of a 99 percent success probability, they are informative for matrices of small dimension because they correctly predict the magnitude of the condition numbers for the sampled matrices.

Note that the bounds in Theorem 4.1 and Corollary 4.2 are informative only for matrices that are tall and skinny ($m \gg n$) and have low coherence. The restriction to tall and skinny matrices is not an imposition, because it is required for the effectiveness of the sampling strategies; see section 3.5.

In the next section we try to relax the restriction to low coherence matrices by more thoroughly exploiting the information available from the row norms of Q.

5. Condition number bounds based on leverage scores, for uniform sampling with replacement. The goal is to tighten Theorem 4.1 by making use

of all the row norms of Q, instead of just the largest one. To this end we introduce leverage scores (section 5.1), which are the squared row norms of Q. We use them to derive a bound for uniform sampling with replacement (section 5.2), and for more easily computable versions of the bound (section 5.3). Analytical (section 5.4) and experimental (section 5.5) comparisons demonstrate that the implied lower bound on the number of sampled rows is better than the coherence-based bounds in section 4. A review with some reflection ends this section (section 5.6).

5.1. Leverage scores. So-called *statistical leverage scores* were first introduced in 1978 by Hoaglin and Welsch [15] to detect outliers when computing regression diagnostics; see also [6, 28]. Mahoney and Drineas pioneered the use of leverage scores for importance sampling strategies in randomized matrix computations [18].

Specifically, if M is a real $m \times n$ matrix with $\text{rank}(M) = n$, then the $m \times m$ *hat matrix*

$$H \equiv M(M^T M)^{-1} M^T$$

is the orthogonal projector onto the column space of M, and its diagonal elements are called *leverage scores* [15, section 2]. Hence, leverage scores are basis independent. For our purposes, though, it suffices to define them in terms of a thin QR decomposition $M = QR$, so that the hat matrix can be expressed as $H = QQ^T$.

DEFINITION 5.1. *If Q is an $m \times n$ matrix with $Q^T Q = I_n$, then its* leverage scores *are*

$$\ell_j \equiv \|e_j^T Q\|_2^2, \qquad 1 \le j \le m.$$

The $m \times m$ diagonal matrix of leverage scores is

$$L \equiv \text{diag}(\ell_1, \ldots, \ell_m).$$

Note that the coherence is the largest leverage score,

$$\mu = \max_{1 \le j \le m} \ell_j = \|L\|_2.$$

5.2. Bounds. The bound in Theorem 5.2 below involves leverage scores and is based on a matrix Bernstein concentration inequality (section A.4), rather than on the matrix Chernoff concentration inequality (section A.1) for Theorem 4.1. Although the Bernstein inequality may not always be as tight, we did not see how to insert leverage scores into the Chernoff inequality.

THEOREM 5.2. *Let Q be an $m \times n$ real matrix with $Q^T Q = I_n$, leverage scores ℓ_j, $1 \le j \le m$, and coherence μ. Let S be a sampling matrix produced by Algorithm* 3.2 *with $n \le c \le m$. For $0 < \epsilon < 1$ set*

$$\delta \equiv 2n \exp\left(-\tfrac{3}{2} \frac{c\epsilon^2}{m\,(3\|Q^T L Q\|_2 + \epsilon\mu)} \right).$$

If $\delta < 1$, then with probability at least $1 - \delta$ we have $\text{rank}(SQ) = n$ *and*

$$\kappa(SQ) \le \sqrt{\frac{1+\epsilon}{1-\epsilon}}.$$

Proof. The proof uses results from [2, 22] and is relegated to section A.5. ∎

Like Theorem 4.1, Theorem 5.2 implies that sampling with replacement is more likely to produce full-rank matrices as the number c of sampled rows increases. Furthermore, for a given total number of rows m, matrices Q with fewer columns n and lower coherence μ are more likely to yield sampled matrices SQ that have full rank. The dependence of $\|Q^T LQ\|_2$ on μ is discussed below.

Remark 5.3. The norm $\|Q^T LQ\|_2$ has simple and tight bounds in terms of the coherence,

$$\mu^2 \leq \|Q^T LQ\|_2 \leq \mu. \tag{5.1}$$

The lower bound follows from $\|Q^T LQ\|_2 = \|L^{1/2}Q\|_2^2$ and

$$\|L^{1/2}Q\|_2 \geq \|e_j^T L^{1/2} Q\|_2 = \ell_j^{1/2}\, \|e_j^T Q\|_2 = \ell_j, \qquad 1 \leq j \leq m,$$

which implies $\|L^{1/2}Q\|_2 \geq \mu$.

The bounds (5.1) are attained for extreme values of the coherence:

- In the case of minimal coherence $\mu = \ell_j$ for all $1 \leq j \leq m$, we have $L = \mu I_m$. Thus $\|Q^T LQ\|_2 = \mu \|Q^T Q\|_2 = \mu$, and the upper bound is attained.
- In the case of maximal coherence $\mu = 1$, we have $\mu^2 = \mu$. Thus $\|Q^T LQ\|_2 = \mu^2 = \mu$, and both lower and upper bounds are attained.

5.3. Computable bounds. We present easily computable bounds for $\|Q^T LQ\|_2$, based on coherence and several of the largest leverage scores.

To this end, we use a labeling of the leverage scores in nonincreasing order,

$$\mu = \ell_{[1]} \geq \cdots \geq \ell_{[m]}.$$

COROLLARY 5.4. *Under the assumptions of Theorem* 5.2, *if* $t \equiv \lfloor 1/\mu \rfloor$, *then*

$$\|Q^T LQ\|_2 \leq \mu \sum_{j=1}^{t} \ell_{[j]} + (1 - t\,\mu)\, \ell_{[t+1]} \leq \mu.$$

If, in addition, t *is an integer, then* $\|Q^T LQ\|_2 \leq \mu \sum_{j=1}^{t} \ell_{[j]}$.

Proof. See section B.2. ∎

The number of large leverage scores appearing in Corollary 5.4 depends on the coherence: few leverage scores for high coherence, but more for low coherence. Henceforth we will use the approximation from Corollary 5.4 instead of the true value $\|Q^T LQ\|_2$, for two reasons: First, numerical experiments show that the approximation tends to be very accurate. Second, the approximation is convenient, because it requires only a leverage score distribution rather than a full-fledged matrix Q.

Remark 5.5. Corollary 5.4 is tight for the extreme cases of minimal and maximal coherence.

- In the case of minimal coherence $\mu = \ell_j$ for all $1 \leq j \leq m$, Remark 5.3 implies $\|Q^T LQ\|_2 = \mu$. The bound in Corollary 5.4 is $\|Q^T LQ\|_2 \leq \mu$, thus tight.
- In the case of maximal coherence $\mu = 1$, Remark 5.3 implies $\|Q^T LQ\|_2 = \mu^2 = \mu$. Corollary 5.4 holds with $t = 1$ and gives the bound $\|Q^T LQ\|_2 \leq \mu$, which is tight as well.

Inserting this approximation for $\|Q^T LQ\|_2$ into the expression for δ in Theorem 5.2 yields a, not necessarily tight, lower bound on the number of samples.

COROLLARY 5.6. *Under the assumptions of Theorem* 5.2,

$$c \geq \tfrac{2}{3} m\, (3\tau + \epsilon\mu)\, \frac{\ln(2n/\delta)}{\epsilon^2},$$

where $\tau \equiv \mu \sum_{j=1}^{t} \ell_{[j]} + (1 - t\,\mu)\,\ell_{[t+1]}$, *samples are sufficient to achieve* $\kappa(SQ) \leq \sqrt{\frac{1+\epsilon}{1-\epsilon}}$ *with probability at least* $1 - \delta$.

In particular, if Q has minimal coherence $\mu = n/m$, then Corollary 5.6 implies that the number of sampled rows should be at least

$$c \geq 3n\,\frac{\ln(2n/\delta)}{\epsilon^2}.$$

This is the same as the coherence-based lower bound (4.1).

To achieve $\kappa(SQ) \leq 10$ with probability at least .99 requires that the number of sampled rows be at least

$$c \geq m\,(2.1\tau + .7\mu)\;(\ln(2n) + 4.7)\,. \tag{5.2}$$

5.4. Analytical comparison of the bounds in sections 4.1 and 5.2. An analytical comparison between Theorems 4.1 and 5.2 is not obvious because they are based on different concentration inequalities. Instead we compare the implied lower bounds for the number of sampled rows, and show that the leverage-score-based bound in Corollary 5.6 is at least as tight as the coherence-based bound in Corollary 4.2.

COROLLARY 5.7. *Under the assumptions of Theorem* 5.2 *and Corollary* 5.4,

$$\tfrac{2}{3}m\,(3\tau + \epsilon\mu)\,\frac{\ln(2n/\delta)}{\epsilon^2} \leq 3m\mu\,\frac{\ln(2n/\delta)}{\epsilon^2}.$$

Hence Corollary 5.6 *is at least as tight as Corollary* 4.2.

Proof. See section B.3. ∎

5.5. Experimental comparison of the bounds in sections 4.1 and 5.2. We present numerical experiments to compare the lower bounds for the number of sampled rows in Corollaries 4.2 and 5.6 for different values of coherence. This gives quantitative insight into the comparison in Corollary 5.7, and illustrates the reduction in the number of sampled rows from Corollary 5.6, as compared to Corollary 4.2.

Experimental setup. As in previous sections, we use $m \times n$ matrices with $m = 10^4$ rows and $n = 5$ columns. The success probability is .99, and $\epsilon = 99/101$ so that the bound for $\kappa(SQ)$ is equal to 10. Hence the bounds in Corollaries 4.2 and 5.6 amount to (4.2) and (5.2), respectively.

We consider two different leverage score distributions: A distribution generated by Algorithm 6.2 with one large leverage score in Table 2, and a distribution generated by Algorithm 6.3 with as many zeros as possible in Table 3.

Table 2. This table shows the lower bounds on the number of sampled rows, for a leverage score distribution generated with Algorithm 6.2 that consists of one large leverage score, equal to the coherence, and all remaining leverage scores being nonzero and identical. The bounds, as well as the approximation τ to $\|Q^T L Q\|_2$, are displayed for eight different values of coherence, ranging from minimal coherence $\mu = n/m$ to $\mu = 100n/m$.

Table 2 illustrates that with increasing coherence, the number of sampled rows implied by Corollary 5.6 is only about 20 percent of that from Corollary 4.2. This is because τ increases much more slowly than μ. For instance, $\tau \approx \mu/10$ when $\mu = 100n/m$.

TABLE 2

Lower bounds for number of sampled rows in Corollaries 4.2 *and* 5.6*, and approximation* τ*, for different values of coherence* μ*. The first value represents minimal coherence* $\mu = n/m$*. Here* $m = 10{,}000$*,* $n = 5$*,* $\delta = .01$*,* $\epsilon = 99/101$*, with leverage scores generated by Algorithm* 6.2*.*

$\mu/(n/m)$	1	5	10	15	20	25	50	100
Cor. 4.2	108	540	1,079	1,618	2,157	2,697	5,393	10,786
Cor. 5.6	96	191	310	432	556	682	1,3343	2,777
$\tau/(n/m)$	1.00	1.01	1.04	1.10	1.19	1.30	2.22	9.95

TABLE 3

Lower bounds for number of sampled rows in Corollaries 4.2 *and* 5.6*, for different values of coherence* μ*. The first value represents minimal coherence* $\mu = n/m$*. Here* $m = 10{,}000$*,* $n = 5$*,* $\delta = .01$*,* $\epsilon = 99/101$*, with leverage scores generated by Algorithm* 6.3*.*

$\mu/(n/m)$	1	5	10	15	20	25	50	100
Cor. 4.2	108	540	1,079	1,618	2,157	2,697	5,393	10,787
Cor. 5.6	96	477	954	1,431	1,908	2,385	4,770	9,539

Table 3. This table shows the lower bounds on the number of sampled rows. The corresponding leverage score distribution is generated with Algorithm 6.3 and consists of as many zeros as possible. All nonzero leverage scores, except possibly one, are equal to the coherence μ, so that $\tau \approx \mu$. The bounds are displayed for eight different values of coherence, ranging from minimal coherence $\mu = n/m$ to $\mu = 100n/m$.

The bounds for Corollary 4.2 are the same as in Table 2, because the coherence values are the same. Since $\tau = \mu$, the difference between Corollaries 4.2 and 5.6 is not as drastic as in Table 2, yet it increases with increasing coherence. For $\mu = 100n/m$, Corollary 5.6 remains informative, while Corollary 4.2 does not.

5.6. Conclusions for section 5. The goal of this section was to derive condition number bounds that are based on leverage scores rather than just coherence, when rows are sampled uniformly with replacement (Algorithm 3.2). Corollary 5.7 and the numerical experiments illustrate that the lower bound on the number of sampled rows implied by Corollary 5.6 is smaller than that from Corollary 4.2.

Although the coherence-based bound in Theorem 4.1 is derived from a stronger concentration inequality than the one for Theorem 5.2, this difference disappears in the weakening necessary to obtain lower bounds for the amount of sampling. Even in cases when the leverage score measure τ is the same as the coherence, Corollary 5.6 still retains a small advantage, which can increase with increasing coherence. Hence Corollary 5.6 tends to remain informative for larger values of coherence, even when Corollary 4.2 fails.

The difference in implied sampling amounts becomes more drastic in the presence of widely varying nonzero leverage scores, and can be as high as ten percent. This is because the coherence-based bound in Corollary 4.2 cannot take advantage of the distribution of the leverage scores.

Hence, when it comes to lower bounds for the number of rows sampled uniformly with replacement, we recommend Corollary 5.6.

We have yet to derive leverage-score-based bounds for the other two sampling strategies, uniform sampling without replacement (Algorithm 3.1) and Bernoulli sampling (Algorithm 3.3).

6. Algorithms for generating matrices with prescribed coherence and leverage scores. In order to investigate the efficiency of the sampling methods in

section 3, and test the tightness of the bounds in sections 4 and 5, we need to generate matrices with orthonormal columns that have prescribed leverage scores and coherence. The algorithms are implemented in the MATLAB package *kappa_SQ* [29].

We present algorithms for generating matrices with prescribed leverage scores and coherence (section 6.1), and for generating particular leverage score distributions with prescribed coherence (section 6.2). Such distributions can then, in turn, serve as inputs for the algorithm in section 6.1. Furthermore we present two classes of structured matrices with prescribed coherence that are easy and fast to generate (section 6.3).

6.1. Matrices with prescribed leverage scores. We present an algorithm that generates matrices with orthonormal columns that have prescribed leverage scores. In section C we prove an existence result to show that this is always possible.

Algorithm 6.1 is a transposed version of [7, Algorithm 3]. It repeatedly applies $m \times m$ Givens rotations G_{ij} that rotate two rows i and j, and are computed from numerically stable expressions [7, section 3.1]. At most $m-1$ such rotations are necessary. Since each rotation affects only two rows, Algorithm 6.1 requires $\mathcal{O}(mn)$ arithmetic operations.

ALGORITHM 6.1. GENERATING A MATRIX WITH PRESCRIBED LEVERAGE SCORES [7].

Input: Integers m and n with $m \geq n \geq 1$
Vector ℓ with elements $0 \leq \ell_1 \leq \cdots \leq \ell_m \leq 1$ and $\sum_{j=1}^{m} \ell_j = n$

Output: $m \times n$ matrix Q with $Q^T Q = I_n$ and leverage scores $\|e_j^T Q\|_2^2 = \ell_j$, $1 \leq j \leq m$

$Q = \begin{pmatrix} I_n & 0_{n \times (m-n)} \end{pmatrix}^T$ {Initialization}
repeat
  Determine indices $i < k < j$ with
  $\|e_i^T Q\|_2^2 < \ell_i$, $\|e_k^T Q\|_2^2 = \ell_k$, $\|e_j^T Q\|_2^2 > \ell_j$
  if $\ell_i - \|e_i^T Q\|_2^2 \leq \|e_j^T Q\|_2^2 - \ell_j$ **then**
    Apply rotation G_{ij} to rows i and j so that $\|e_i^T G_{ij}\, Q\|_2^2 = \ell_i$
  else
    Apply rotation G_{ij} to rows i and j so that $\|e_j^T G_{ij}\, Q\|_2^2 = \ell_j$
  end if
  $Q = G_{ij}\, Q$ {Update}
until no more such indices exist

6.2. Leverage score distributions with prescribed coherence. We present algorithms that generate leverage score distributions for prescribed coherence. The resulting distributions then serve as inputs for Algorithm 6.1. These particular leverage score distributions help to distinguish the effect of coherence, which is the largest leverage score, from that of the remaining leverage scores.

One large leverage score. Given a prescribed coherence μ, Algorithm 6.2 generates a distribution consisting of one large leverage score equal to μ and the remaining leverage scores being identical and nonzero.

In the special case of minimal coherence $\mu = n/m$, Algorithm 6.2 generates m identical leverages equal to μ, which is the only possible leverage score distribution in this case.

ALGORITHM 6.2. GENERATING A LEVERAGE SCORE DISTRIBUTION WITH PRESCRIBED COHERENCE: ONE LARGE LEVERAGE SCORE.

Input: Integers m and n with $m \geq n \geq 1$
Real number μ with $n/m \leq \mu \leq 1$
Output: Vector ℓ with elements $\ell_1 = \mu$, $0 < \ell_j \leq 1$ and $\sum_{j=1}^{m} \ell_j = n$

$\ell_1 = \mu$
for $j = 2 : m$ **do**
  $\ell_j = \frac{n-\mu}{m-1}$
end for

Many zero leverage scores. Given a prescribed coherence, Algorithm 6.3 generates a distribution with as many zero leverage scores as possible. This serves as an "adversarial" distribution for the sampling algorithms in section 3.

Given a prescribed coherence μ, Algorithm 6.3 first determines the smallest number of rows m_s that can realize this coherence, sets $m_s - 1$ leverage scores equal to μ, assigns another leverage score to take up the possibly nonzero slack, and sets the remaining leverage scores to zero.

ALGORITHM 6.3. GENERATING A LEVERAGE SCORE DISTRIBUTION WITH PRESCRIBED COHERENCE: MANY ZERO LEVERAGE SCORES.

Input: Integers m and n with $m \geq n \geq 1$
Real number μ with $n/m \leq \mu \leq 1$
Output: Vector ℓ with elements $\ell_1 = \mu$, $0 \leq \ell_j \leq 1$ and $\sum_{j=1}^{m} \ell_j = n$

$m_s = \lceil n/\mu \rceil$ {Number of nonzero rows}
for $j = 1 : m_s - 1$ **do**
  $\ell_j = \mu$
end for
$\ell_{m_s} = n - (m_s - 1)\,\mu$
for $j = m_s + 1 : m$ **do**
  $\ell_j = 0$
end for

6.3. Structured matrices with prescribed coherence. We present two classes of structured matrices with orthonormal columns that have prescribed coherence. Although the structure puts constraints on the matrix dimensions, the generation of these matrices is faster than running Algorithm 6.1. Note that the matrices produced by Algorithm 6.1 also have structure, but it is not easily characterized.

Stacks of diagonal matrices. Given matrix dimensions m and n, where $s = m/n$ is an integer, and prescribed coherence μ. The $m \times n$ matrix Q below has orthonormal columns and coherence μ, and consists of s stacks of $n \times n$ diagonal matrices:

$$Q = \begin{pmatrix} \sqrt{\mu}\, I_n \\ \phi\, I_n \\ \vdots \\ \phi\, I_n \end{pmatrix}, \qquad \textit{where} \qquad \phi \equiv \sqrt{\frac{1-\mu}{\frac{m}{n} - 1}}.$$

Matrices with Hadamard structure. Given matrix dimensions m and n, where $m = 2^k$ and $n < m$ is also a power of two, and prescribed coherence μ. The $m \times n$

matrix

$$Q = D_k \begin{pmatrix} I_n \\ 0 \end{pmatrix}$$

has orthonormal columns and coherence μ, and is defined recursively as follows. For

$$\alpha \equiv \sqrt{\frac{\mu - \frac{n-1}{m-1}}{1 - \frac{n-1}{m-1}}}, \qquad \beta \equiv \sqrt{\frac{1-\alpha^2}{m-1}},$$

define square matrices B_j of dimension 2^j and square matrices D_j of dimension 2^{j+1} as follows:

$$B_0 = \beta, \qquad B_{j+1} = \begin{pmatrix} -B_j & B_j \\ B_j & B_j \end{pmatrix}, \qquad 0 \le j \le k-1,$$

$$D_1 = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}, \qquad D_{j+1} = \begin{pmatrix} D_j & -B_j \\ B_j & D_j \end{pmatrix}.$$

Note that only the final matrix Q has orthonormal columns and coherence μ while, in general, the intermediate matrices B_j and D_j do not. We omit the messy induction proof, because it does not provide much insight.

7. Future work. We have investigated three strategies for uniform sampling of rows from matrices with orthonormal columns: without replacement, with replacement, and Bernoulli sampling. We derived bounds on the condition numbers of the sampled matrices, in terms of coherence and leverage scores. Numerical experiments confirm that the bounds are realistic, even for high success probabilities and matrices with small dimensions.

The following work still needs to be done.

- Conversion of the *kappa_SQ* MATLAB toolbox from a research code to a robust, flexible, and user-friendly GUI that facilitates reproducible research in the randomized algorithms community.
- Tightening of Corollary 4.2 so that it retains the strength of the Chernoff concentration inequality inherent in Theorem 5.2.
- Extension of the condition number bounds in section 5 to uniform sampling without replacement (Algorithm 3.1) and Bernoulli sampling (Algorithm 3.3).
- Determination of a statistically significant number of runs for each sampling amount c, for two purposes:
  1. To assert, within a specific confidence interval, bounds on the condition numbers of the *actually sampled* matrices.
  2. To assert with a specific confidence that the probabilistic expressions in sections 4 and 5 do indeed represent bounds.

Appendix A. Proofs for sections 4 and 5.2. For the coherence-based bounds in section 4 we first present a matrix concentration inequality (section A.1), and then the proofs of Theorem 4.1 (section A.2) and Corollary 4.2 (section A.3).

For the bound based on leverage scores in section 5.2, we first present a matrix concentration inequality (section A.4), and then the proof of Theorem 5.2 (section A.5).

A.1. Matrix Chernoff concentration inequality. The matrix concentration inequality below is the basis for Theorem 4.1 and Corollary 4.2.

Denote the eigenvalues of a Hermitian matrix Z by $\lambda_j(Z)$, and the smallest and largest eigenvalues by $\lambda_{\min}(Z) \equiv \min_j \lambda_j(Z)$ and $\lambda_{\max}(Z) \equiv \max_j \lambda_j(Z)$, respectively.

THEOREM A.1 (Corollary 5.2 in [27]). *Let X_j be a finite number of independent random $n \times n$ Hermitian positive semidefinite matrices with $\max_j \|X_j\|_2 \le \tau$. Define*

$$\omega_{\min} \equiv \lambda_{\min}\left(\sum_j \mathbf{E}[X_j]\right), \qquad \omega_{\max} \equiv \lambda_{\max}\left(\sum_j \mathbf{E}[X_j]\right),$$

and $f(x) \equiv e^x(1+x)^{-(1+x)}$. Then for any $0 \le \epsilon < 1$

$$\mathbf{Pr}\left[\lambda_{\min}\left(\sum_j X_j\right) \le (1-\epsilon)\,\omega_{\min}\right] \le n\, f(-\epsilon)^{\omega_{\min}/\tau},$$

and for any $\epsilon \ge 0$

$$\mathbf{Pr}\left[\lambda_{\max}\left(\sum_j X_j\right) \ge (1+\epsilon)\,\omega_{\max}\right] \le n\, f(\epsilon)^{\omega_{\max}/\tau}.$$

A.2. Proof of Theorem 4.1. We present a separate proof for each sampling method.

Algorithm 3.1: *Sampling without replacement.* The proof follows directly from [26, Lemma 3.4].

Algorithm 3.2: *Sampling with replacement.* The proof is based on Theorem A.1, and turns out to be somewhat similar to that of [26, Lemma 3.4].

Set $X_t \equiv \frac{m}{c}\, Q^T e_{k_t} e_{k_t}^T Q$, $1 \le t \le c$. Then X_t is $n \times n$ Hermitian positive semidefinite and $\|X_t\|_2 \le \frac{m}{c}\|e_{k_t}^T Q\|_2^2 \le \frac{m\mu}{c}$. Hence we set $\tau = m\mu/c$. Furthermore,

$$\mathbf{E}[X_t] = \sum_{j=1}^{m} \tfrac{1}{m}\,\left(\tfrac{m}{c}\, Q^T e_j e_j^T Q\right) = \tfrac{1}{c}\,\sum_{j=1}^{m} Q^T e_j e_j^T Q = \tfrac{1}{c} I_n.$$

Hence the eigenvalues of the sum are $\lambda_j\left(\sum_{t=1}^{c} \mathbf{E}[X_t]\right) = \lambda_j(I_n) = 1$, $1 \le j \le n$, and we set $\omega_{\min} = \omega_{\max} = 1$. Applying Theorem A.1 to $\sum_{t=1}^{c} X_t = Q^T S^T S Q$ gives

$$\mathbf{Pr}\left[\lambda_{\min}\left(Q^T S^T S Q\right) \le 1-\epsilon\right] \le n f(-\epsilon)^{c/(m\mu)},$$
$$\mathbf{Pr}\left[\lambda_{\max}\left(Q^T S^T S Q\right) \ge 1+\epsilon\right] \le \; n f(\epsilon)^{c/(m\mu)}.$$

The result follows from Boole's inequality [24, p. 16].

Algorithm 3.3: *Bernoulli sampling.* The proof is similar to the one above, and a special case of [12, Theorem 6.1].

Set

$$X_j \equiv \tfrac{m}{c}\begin{cases} Q^T e_j e_j^T Q & \text{with probability } \frac{c}{m}, \\ 0_{n\times n} & \text{with probability } 1-\frac{c}{m}, \end{cases} \qquad 1 \le j \le m.$$

Then X_j is $n \times n$ Hermitian positive semidefinite, $\|X_j\|_2 \le \frac{m}{c}\|e_j^T Q\|_2^2 \le \frac{m\mu}{c}$. As above, we set $\tau = m\mu/c$. Furthermore,

$$\mathbf{E}[X_j] = \tfrac{c}{m} \cdot \tfrac{m}{c} Q^T e_j e_j^T Q + (1-\tfrac{c}{m}) \cdot 0_{n\times n} = Q^T e_j e_j^T Q,$$

which implies $\sum_{j=1}^{m} \mathbf{E}[X_j] = \sum_{j=1}^{m} Q^T e_j e_j^T Q = I_n$. Now proceed as in the above proof for Algorithm 3.2, and apply Theorem A.1 to $\sum_{j=1}^{m} X_j = Q^T S^T S Q$.

A.3. Proof of Corollary 4.2. First we simplify the bound in Theorem 4.1 based on the inequality $f(-x) \leq f(x)$ for $0 < x < 1$. This implies for Theorem 4.1 that

$$\delta \equiv n \left(f(-\epsilon)^{c/(m\mu)} + f(\epsilon)^{c/(m\mu)} \right) \delta \leq 2n \, f(\epsilon)^{c/(m\mu)}.$$

Solving for c gives

$$c \geq m\mu \, \frac{\ln(2n/\delta)}{-\ln f(\epsilon)}.$$

If we can show that $-\ln f(\epsilon) > \epsilon^2/3$, then the above lower bound for c definitely holds if

$$c \geq 3m\mu \, \frac{\ln(2n/\delta)}{\epsilon^2}.$$

To show $-\ln f(\epsilon) > \epsilon^2/3$ for $0 < \epsilon < 1$, apply the definition $f(x) = e^x(1+x)^{-(1+x)}$ so that $h(x) \equiv -\ln f(x) = (1+x)\ln(1+x) - x$. Expand into the power series $\ln(1+x) = \sum_{j=1}^{\infty} (-1)^{j+1} \frac{x^j}{j}$. For $0 < x < 1$ this yields $h(x) = \frac{1}{2}x^2 - \frac{1}{6}x^3 + E(x)$, where

$$E(x) \equiv \sum_{j=4}^{\infty} (-1)^j \, \frac{x^j}{(j-1)j} = \sum_{j=2}^{\infty} \left(\frac{2j+1-(2j-1)x}{(2j-1)2j(2j+1)} \right) \, x^{2j} > 0,$$

since each summand is positive for $0 < x < 1$. Thus for $0 < x < 1$ we obtain

$$h(x) > \frac{1}{2}x^2 - \frac{1}{6}x^3 = \frac{3-x}{6} \, x^2 \geq \frac{x^2}{3}.$$

A.4. Matrix Bernstein concentration inequality. The matrix concentration inequality below is the basis for Theorem 5.2. It is a version specialized to square matrices of [22, Theorem 4]. In numerical experiments we found it to be tighter than [11, Theorem 4] and the Frobenius norm bound [9, Theorem 2].

THEOREM A.2 (Theorem 4 in [22]). *Let X_j be m independent random $n \times n$ matrices with $\mathbf{E}[X_j] = 0_{n\times n}$, $1 \leq j \leq m$. Let $\rho_j \equiv \max\{\|\mathbf{E}[X_j X_j^T]\|_2, \|\mathbf{E}[X_j^T X_j]\|_2\}$ and $\max_{1\leq j\leq m} \|X_j\|_2 \leq \tau$. Then for any $\epsilon > 0$*

$$\mathbf{Pr}\left[\left\| \sum_{j=1}^{m} X_j \right\|_2 > \epsilon \right] \leq 2n \, \exp\left(-\tfrac{3}{2} \, \frac{\epsilon^2}{3\sum_{j=1}^{m} \rho_j + \tau\epsilon} \right).$$

A.5. Proof of Theorem 5.2. The proof is similar to that of [2, Lemma 3]. Represent the outcome of uniform sampling with replacement in Algorithm 3.2 by $Q^T S^T S Q = \sum_{t=1}^{c} Y_t$, where $Y_t \equiv \frac{m}{c} \, Q^T e_{k_t} e_{k_t}^T Q$ are $n \times n$ matrices, $1 \leq t \leq c$, with expected value

$$\mathbf{E}[Y_t] = \sum_{j=1}^{m} \frac{1}{m} \, \frac{m}{c} Q^T e_j e_j^T Q = \frac{1}{c} \, \sum_{j=1}^{m} Q^T e_j e_j^T Q = \frac{1}{c} I_n.$$

Thus, the zero mean versions are $X_t \equiv Y_t - \frac{1}{c} I_n$. To apply Theorem A.2 to the X_t we need to verify that they fulfill the required conditions. First, by construction, $\mathbf{E}[X_t] = 0$, $1 \leq t \leq c$. Second, since Y_t and I_n are symmetric positive semidefinite,

$$\|X_t\|_2 \leq \max\left\{ \|Y_t\|_2, \left\| \frac{1}{c} I_n \right\|_2 \right\} = \frac{1}{c} \max\{m \, \|e_{k_t}^T Q\|_2^2, 1\} \leq \frac{m\mu}{c},$$

where the last inequality follows from the definition of μ, and $\mu \geq n/m$. Hence we set $\tau = m\mu/c$. Third, since X_t is symmetric,

$$X_t^T X_t = X_t X_t^T = X_t^2 = Y_t^2 - \frac{2}{c} Y_t + \frac{1}{c^2} I_n.$$

From $\mathbf{E}[Y_t] = \frac{1}{c} I_n$ it follows

$$\mathbf{E}[X_t^2] = \mathbf{E}[Y_t^2] - \frac{2}{c}\,\mathbf{E}[Y_t] + \frac{1}{c^2} I_n = \mathbf{E}[Y_t^2] - \frac{1}{c^2} I_n. \tag{A.1}$$

Since $Y_t^2 = \frac{m^2}{c^2}\, \ell_{k_t}\, Q^T e_{k_t} e_{k_t}^T Q$, we obtain

$$\mathbf{E}[Y_t^2] = \sum_{j=1}^{m} \frac{1}{m}\, \frac{m^2}{c^2}\, \ell_j Q^T e_j e_j^T Q = \frac{m}{c^2}\, Q^T \left(\sum_{j=1}^{m} \ell_j e_j e_j^T \right) Q = \frac{m}{c^2}\, Q^T L Q.$$

Substituting this into (A.1) yields

$$\mathbf{E}[X_t^2] = \tfrac{1}{c^2} \left(m\ Q^T L Q - I_n \right).$$

Positive semidefiniteness gives

$$\|\mathbf{E}[X_t^2]\|_2 \leq \frac{1}{c^2}\, \max\{m\, \|Q^T L Q\|_2,\, 1\} = \frac{m}{c^2}\, \|Q^T L Q\|_2.$$

We set $\rho_t = \frac{m}{c^2}\, \|Q^T L Q\|_2$. Applying [22, Theorem 4] to

$$\sum_{t=1}^{c} X_t = \sum_{t=1}^{c} \left(Y_t - \tfrac{1}{c} I_n \right) = (SQ)^T (SQ) - I_n$$

shows that $\| \sum_{t=1}^{c} X_t \|_2 \leq \epsilon$ with probability at least $1 - \delta$.

Appendix B. Two-norm bound for scaled matrices, and proofs for sections 5.3 and 5.4. We derive a bound for the two-norm of diagonally scaled matrices (section B.1), which leads immediately to the proofs of Corollary 5.4 (section B.2), and Corollary 5.7 (section B.3).

B.1. Bound. We present two majorization bounds for Hadamard products of vectors (Lemmas B.3 and B.2), and use them to derive a bound for the two-norm of diagonally scaled matrices (Theorem B.4).

Definition B.1 (Definition 4.3.41 in [16]). *Let a and b be vectors with m real elements. The elements, labeled in algebraically decreasing order, are $a_{[1]} \geq \cdots \geq a_{[m]}$ and $b_{[1]} \geq \cdots \geq b_{[m]}$. The vector a* weakly majorizes *the vector b, if*

$$\sum_{j=1}^{k} a_{[j]} \geq \sum_{j=1}^{k} b_{[j]}, \qquad 1 \leq k \leq m.$$

The vector a majorizes *the vector b, if a weakly majorizes b and also $\sum_{j=1}^{m} a_{[j]} = \sum_{j=1}^{m} b_{[j]}$.*

The first lemma follows from a stronger majorization inequality for functions that are monotone and lattice superadditive.

Lemma B.2 (Theorem II.4.2 in [3]). *If b and x are vectors with m nonnegative elements, then*

$$\sum_{j=1}^{k} b_j\, x_j \leq \sum_{j=1}^{k} b_{[j]}\, x_{[j]}, \qquad 1 \leq k \leq m.$$

The second lemma is a variant of a well-known majorization result for Hadamard products of vectors [16, Lemma 4.3.51]. Since the version below is slightly different, we include a proof from first principles.

LEMMA B.3. *Let x, a, and b be vectors with m nonnegative elements. If a weakly majorizes b, then*

$$\sum_{j=1}^{k} a_{[j]}\, x_{[j]} \geq \sum_{j=1}^{k} b_{[j]}\, x_{[j]}, \qquad 1 \leq k \leq m.$$

Proof. The following arguments hold for $1 \leq k \leq m-1$. Start out with the upper bound, and separate the last summand,

$$\sum_{j=1}^{k+1} a_{[j]}\, x_{[j]} = \sum_{j=1}^{k} a_{[j]}\, x_{[j]} + a_{[k+1]}\, x_{[k+1]}. \tag{B.1}$$

Rewriting the right sum and applying $x_{[j]} \geq x_{[k+1]} \geq 0$, $1 \leq j \leq k$, gives

$$\begin{aligned}
\sum_{j=1}^{k} a_{[j]}\, x_{[j]} &= \sum_{j=1}^{k} b_{[j]}\, x_{[j]} + \sum_{j=1}^{k} (a_{[j]} - b_{[j]})\, x_{[j]} \\
&\geq \sum_{j=1}^{k} b_{[j]}\, x_{[j]} + \sum_{j=1}^{k} (a_{[j]} - b_{[j]})\, x_{[k+1]} \\
&= \sum_{j=1}^{k} b_{[j]}\, x_{[j]} + \left(\sum_{j=1}^{k} a_{[j]} - \sum_{j=1}^{k} b_{[j]} \right) x_{[k+1]}.
\end{aligned}$$

Insert this into (B.1) and gather common terms,

$$\begin{aligned}
\sum_{j=1}^{k+1} a_{[j]}\, x_{[j]} &\geq \sum_{j=1}^{k} b_{[j]}\, x_{[j]} + \left(\sum_{j=1}^{k+1} a_{[j]} - \sum_{j=1}^{k} b_{[j]} \right) x_{[k+1]} \\
&\geq \sum_{j=1}^{k} b_{[j]}\, x_{[j]} + b_{[k+1]}\, x_{[k+1]} = \sum_{j=1}^{k+1} b_{[j]}\, x_{[j]},
\end{aligned}$$

where the second inequality follows from the majorization $\sum_{j=1}^{k+1} a_{[j]} \geq \sum_{j=1}^{k+1} b_{[j]}$. ∎

Now we are ready to bound the two-norm of a row scaled matrix DZ, where Z is $m \times n$ of full column rank, and $D = \operatorname{diag}\begin{pmatrix} d_1 & \dots & d_m \end{pmatrix}$ is a nonnegative $m \times m$ diagonal matrix. The obvious bound is

$$\|DZ\|_2 \leq \|D\|_2\, \|Z\|_2 = d_{[1]}\, \|Z\|_2. \tag{B.2}$$

However, the bound in Theorem B.4 below, which incorporates the largest row norm of Z and several of the largest (in magnitude) diagonal elements of D, turns out to be tighter.

THEOREM B.4. *Let Z be a real $m \times n$ matrix with* $\operatorname{rank}(Z) = n$*, smallest singular value $\sigma_z = 1/\|Z^\dagger\|_2$, and largest squared row norm $\mu_z \equiv \max_{1 \leq j \leq m} \|e_j^T Z\|_2^2$. If $t \equiv \lfloor \sigma_z^2/\mu_z \rfloor$, then*

$$\|DZ\|_2^2 \leq \begin{cases} \mu_z \sum_{j=1}^{t} d_{[j]}^2 + \left(\|Z\|_2^2 - t\, \mu_z \right)\, d_{[t+1]}^2 & \text{if } \|Z\|_2^2 - t\, \mu_z \leq \mu_z, \\ \mu_z \sum_{j=2}^{t+1} d_{[j]}^2 + \left(\|Z\|_2^2 - t\, \mu_z \right)\, d_{[1]}^2 & \text{otherwise.} \end{cases}$$

Proof. Let z be an $n \times 1$ vector with $\|z\|_2 = 1$ and $\|DZ\|_2 = \|DZz\|_2$. Furthermore let $z_j \equiv e_j^T Z z$, $1 \le j \le m$, be the elements of Zz, so that $\|Zz\|_2^2 = \sum_{j=1}^m z_j^2$.

Apply Lemma B.2. Since $d_j^2 \ge 0$ and $z_j^2 \ge 0$, $1 \le j \le m$, we can apply Lemma B.2 with $x_j = d_j^2$ and $b_j = z_j^2$, to obtain

$$\|DZ\|_2^2 = \|DZz\|_2^2 = \sum_{j=1}^m d_j^2 \, z_j^2 = \sum_{j=1}^m b_j \, x_j \le \sum_{j=1}^m b_{[j]} \, x_{[j]}.$$

Verify assumptions of Lemma B.3. In order to apply Lemma B.3 with

$$a_j = \mu_z, \quad 1 \le j \le t, \qquad a_{t+1} = \|Zz\|_2^2 - t\,\mu_z, \qquad a_j = 0, \quad t+2 \le j \le m,$$

we need to show that the assumptions are satisfied, meaning all vector elements are nonnegative and the majorization condition holds. Clearly $a_j \ge 0$ for $1 \le j \le t$ and $t+2 \le j \le m$. This leaves a_{t+1}. From $\operatorname{rank}(Z) = n$ it follows that $\sigma_z > 0$. The definition of t implies $0 \le t \le \sigma_z^2/\mu_z$, so that

$$0 \le \sigma_z^2 - t\,\mu_z = \min_{\|y\|_2 = 1} \|Zy\|_2^2 - t\,\mu_z \le \|Zz\|_2^2 - t\,\mu_z = a_{t+1}.$$

Thus, all vector elements are nonnegative.

To show the majorization condition, start with the Cauchy–Schwarz inequality,

$$b_j = z_j^2 = (e_j^T Z\, z)^2 \le \|e_j^T Z\|_2^2 \, \|z\|_2^2 = \|e_j^T Z\|_2^2 \le \mu_z, \qquad 1 \le j \le m.$$

This yields, regardless of whether $a_{t+1} = \|Zz\|_2^2 - t\,\mu_z \le \mu_z$ or not,

$$\sum_{j=1}^k a_{[j]} \ge \sum_{j=1}^k \mu_z \ge \sum_{j=1}^k z_{[j]}^2 = \sum_{j=1}^k b_{[j]}, \qquad 1 \le k \le t.$$

Moreover, for $1 \le k \le m - t$,

$$\sum_{j=1}^{t+k} a_{[j]} = \sum_{j=1}^t \mu_z + (\|Zz\|_2^2 - t\,\mu_z) = \|Zz\|_2^2 \ge \sum_{j=1}^{t+k} z_{[j]}^2 = \sum_{j=1}^{t+k} b_{[j]}.$$

This gives the weak majorization condition $\sum_{j=1}^k a_{[j]} \ge \sum_{j=1}^k b_{[j]}$, $1 \le k \le m$.

Apply Lemma B.3. Since the assumptions of Lemma B.3 are satisfied, we can conclude that $\sum_{j=1}^m b_{[j]} \, x_{[j]} \le \sum_{j=1}^m a_{[j]} \, x_{[j]}$. At last, substitute into this majorization relation the expressions for a and b. If $\|Z\|_2^2 - t\,\mu_z \le \mu_z$, then

$$\sum_{j=1}^m a_{[j]} \, x_{[j]} = \mu_z \sum_{j=1}^t d_{[j]}^2 + (\|Zz\|_2^2 - t\,\mu_z)\, d_{[t+1]}^2 \le \mu_z \sum_{j=1}^t d_{[j]}^2 + (\|Z\|_2^2 - t\,\mu_z)\, d_{[t+1]}^2,$$

otherwise

$$\sum_{j=1}^m a_{[j]} \, x_{[j]} = (\|Zz\|_2^2 - t\,\mu_z)\, d_{[1]}^2 + \mu_z \sum_{j=2}^{t+1} d_{[j]}^2 \le (\|Z\|_2^2 - t\,\mu_z)\, d_{[1]}^2 + \mu_z \sum_{j=2}^{t+1} d_{[j]}^2. \qquad \square$$

Theorem B.4 is tighter than (B.2) because $d_{[j]}^2 \le \|D\|_2^2$ implies

$$\begin{aligned} \|DZ\|_2^2 &\le \begin{cases} \mu_z \sum_{j=1}^t d_{[j]}^2 + \left(\|Z\|_2^2 - t\,\mu_z\right)\, d_{[t+1]}^2 & \text{if } \|Z\|_2^2 - t\,\mu_z \le \mu_z, \\ \mu_z \sum_{j=2}^{t+1} d_{[j]}^2 + \left(\|Z\|_2^2 - t\,\mu_z\right)\, d_{[1]}^2 & \text{otherwise} \end{cases} \\ &\le t\mu_z \|D\|_2^2 - \left(\|Z\|_2^2 - t\,\mu_z\right)\, \|D\|_2^2 = \|D\|_2^2 \|Z\|_2^2. \end{aligned}$$

COROLLARY B.5. *Let Z be a real $m \times n$ matrix with $Z^T Z = I_n$, and coherence $\mu_z \equiv \max_{1 \le j \le m} \|e_j^T Z\|_2^2$. If $t \equiv \lfloor 1/\mu_z \rfloor$, then*

$$\|DZ\|_2^2 \le \mu_z \sum_{j=1}^{t} d_{[j]}^2 + (1 - t\,\mu_z)\; d_{[t+1]}^2.$$

Proof. Applying Theorem B.4 and assuming $1 - t\,\mu_z \le \mu_z$ gives

$$\|DZ\|_2^2 \le \mu_z \sum_{j=1}^{t} d_{[j]}^2 + (1 - t\,\mu_z)\; d_{[t+1]}^2.$$

The assumption $1 - t\,\mu_z \le \mu_z$ is justified because

$$1 - t\,\mu_z = 1 - \lfloor 1/\mu_z \rfloor\; \mu_z \le 1 - (1/\mu_z - 1)\mu_z = \mu_z. \qquad \square$$

B.2. Proof of Corollary 5.4. Apply Corollary B.5 with $D = L^{1/2}$, $Z = Q$, $\mu_z = \mu$, and $t = \lfloor 1/\mu \rfloor$ to prove the first inequality,

$$\|Q^T L Q\|_2 = \|L^{1/2} Q\|_2^2 \le \mu \sum_{j=1}^{t} \ell_{[j]} + (1 - t\,\mu)\,\ell_{[t+1]}.$$

As for the second inequality, $\ell_{[j]} \le \mu$ implies

$$\mu \sum_{j=1}^{t} \ell_{[j]} + (1 - t\,\mu)\,\ell_{[t+1]} \;\le t\mu^2 + (1 - t\mu)\mu = \mu.$$

If, in addition, t is an integer, then $t = 1/\mu$ and $1 - t\,\mu = 0$.

B.3. Proof of Corollary 5.7. Define the common term $\phi \equiv m\,\ln(2n/\delta)/\epsilon^2$ in both bounds, and write Corollary 4.2 as $c \ge 3\mu\phi$, and Corollary 5.6 as $c \ge (2\tau + \frac{2}{3}\epsilon\,\mu)\phi$. From $\epsilon < 1$ and $\tau \le \mu$ it follows that

$$2\tau + \tfrac{2}{3}\epsilon\,\mu \le 3\mu.$$

Appendix C. Existence of matrices with prescribed coherence and leverage scores. This section is the basis for Algorithm 6.1. We review a well-known majorization result (Theorem C.1). We use it to show (Theorem C.2) that, given prescribed matrix dimensions and leverage scores, there always exists a matrix Q with orthonormal columns that has the required dimensions and (squared) row norms equal to the leverage scores.

Our approach is again based on majorization, see Definition B.1, and in particular on the fact that the eigenvalues of a real symmetric matrix majorize its diagonal elements.

THEOREM C.1 (Theorem 4.3.48 in [16]). *Let a and λ be vectors with real elements a_j and λ_j, respectively, $1 \le j \le m$. If λ majorizes a, then there exists an $m \times m$ real symmetric matrix with eigenvalues λ_j and diagonal elements a_j, $1 \le j \le m$.*

With the help of Theorem C.1 we show that there exists a matrix with orthonormal columns that has prescribed leverage scores and coherence.

THEOREM C.2. *Given integers m and n with $m \ge n \ge 1$, and a vector ℓ with m elements ℓ_j that satisfy $0 \le \ell_j \le 1$ and $\sum_{j=1}^{m} \ell_j = n$. Then there exists an $m \times n$*

matrix Q with orthonormal columns that has leverage scores $\|e_j^T Q\|_2^2 = \ell_j$, $1 \le j \le m$, and coherence $\mu = \max_{1\le j\le m} \ell_j$.

Proof. Let λ be a vector with m elements that satisfy $\lambda_j = 1$ for $1 \le j \le n$, and $\lambda_j = 0$ for $n+1 \le j \le m$. We are going to construct a matrix Q by applying Theorem C.1 to λ and ℓ. To this end, we first need to show that λ majorizes ℓ.

Majorization. We distinguish the cases $1 \le k \le n$ and $n+1 \le k \le m$.

Case $1 \le k \le n$: From $\ell_j \le 1$ it follows that

$$\sum_{j=1}^{k} \lambda_j = k \ge \sum_{j=1}^{k} \ell_{[j]}.$$

Case $n+1 \le k \le m$: From $\ell_j \ge 0$ and $\sum_{j=1}^{m} \ell_j = n$ it follows that

$$\sum_{j=1}^{k} \lambda_j = n = \sum_{j=1}^{k} \ell_{[j]} + \sum_{j=k+1}^{m} \ell_{[j]} \ge \sum_{j=1}^{k} \ell_{[j]}.$$

Hence

$$\sum_{j=1}^{k} \lambda_j \ge \sum_{j=1}^{k} \ell_{[j]}, \qquad 1 \le k \le m,$$

which means that λ weakly majorizes ℓ. Since also $\sum_{j=1}^{m} \lambda_j = n = \sum_{j=1}^{m} \ell_{[j]}$, we can conclude that λ majorizes ℓ.

Construction of Q. Theorem C.1 implies that there exists a real symmetric matrix W with eigenvalues λ_j and diagonal elements $W_{jj} = \ell_j$, $1 \le j \le m$. Since W has n eigenvalues equal to one, and all other eigenvalues equal to zero, it has an eigenvalue decomposition

$$W = \hat{Q} \begin{pmatrix} I_n & 0 \\ 0 & 0 \end{pmatrix} \hat{Q}^T = QQ^T,$$

where $\hat{Q}$ is an $m \times m$ real orthogonal matrix, and $Q \equiv \hat{Q} \begin{pmatrix} I_n & 0 \end{pmatrix}^T$ has n orthonormal columns. Therefore Q has leverage scores $\|e_j^T Q\|_2^2 = e_j^T QQ^T e_j = W_{jj} = \ell_j$ and coherence $\mu = \max_{1\le j\le m} \ell_j$. ∎

Acknowledgments. We are very grateful to John Holodnak, Petros Drineas, and two anonymous reviewers for reading our paper so carefully and for providing many helpful suggestions.

REFERENCES

[1] H. Avron, P. Maymounkov, and S. Toledo, *Blendenpik: Supercharging Lapack's least-squares solver*, SIAM J. Sci. Comput., 32 (2010), pp. 1217–1236.

[2] L. Balzano, B. Recht, and R. Nowak, *High-dimensional matched subspace detection when data are missing*, Information Theory Proceedings, IEEE, 2010, pp. 1638–1642.

[3] R. Bhatia, *Matrix Analysis*, Grad. Texts in Math. 169, Springer, New York, 1997.

[4] C. Boutsidis and P. Drineas, *Random projections for the nonnegative least-squares problem*, Linear Algebra Appl., 431 (2009), pp. 760–771.

[5] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[6] S. Chatterjee and A. S. Hadi, *Influential observations, high leverage points, and outliers in linear regression*, Statist. Sci., 1 (1986), pp. 379–393.

[7] I. S. Dhillon, R. W. Heath, Jr., M. A. Sustik, and J. A. Tropp, *Generalized finite algorithms for constructing Hermitian matrices with prescribed diagonal and spectrum*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 61–71.
[8] D. L. Donoho and X. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. Inform. Theory, 47 (2001), pp. 2845–2862.
[9] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices. I: Approximating matrix multiplication*, SIAM J. Comput., 36 (2006), pp. 132–157.
[10] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Sampling algorithms for l_2 regression and applications*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, 2006, ACM, New York, pp. 1127–1136.
[11] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, *Faster least squares approximation*, Numer. Math., 117 (2011), pp. 219–249.
[12] A. Gittens and J. A. Tropp, *Tail Bounds for All Eigenvalues of a Sum of Random Matrices*, Technical report, ACM 2014-01, arXiv:1104.4513, 2014.
[13] D. Gross and V. Nesme, *Note on Sampling without Replacement from a Finite Collection of Matrices*, preprint, arXiv:1001.2738, 2010.
[14] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
[15] D. C. Hoaglin and R. E. Welsch, *The hat matrix in regression and ANOVA*, Amer. Statist., 32 (1978), pp. 17–22.
[16] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed., Cambridge University Press, Cambridge, 2013.
[17] D. V. Lindley, *The Bayesian approach*, Scand. J. Statist., 5 (1978), pp. 1–26.
[18] M. W. Mahoney, *Randomized Algorithms for Matrices and Data*, Now Publishers, Hanover, MA, 2011.
[19] X. Meng, M. A. Saunders, and M. W. Mahoney, *LSRN: A parallel iterative solver for strongly over- or under-determined systems*, SIAM J. Sci. Comput., 36 (2014), pp. C95–C118.
[20] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, New York, 2006.
[21] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[22] B. Recht, *A simpler approach to matrix completion*, J. Mach. Learn. 12 (2011), pp. 3413–3430.
[23] V. Rokhlin and M. Tygert, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 13212–13217.
[24] S. Ross, *Introduction to Probability Models*, 9th ed., Academic Press, Amsterdam, 2007.
[25] A. Talwalkar and A. Rostamizadeh, *Matrix coherence and the Nyström method*, in Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence UAI 2010, AUAI Press, Arlington, VA, pp. 572–579.
[26] J. A. Tropp, *Improved analysis of the subsampled Hadamard transform*, Adv. Adapt. Data Anal., 3 (2011), pp. 115–126.
[27] J. A. Tropp, *User-friendly tail bounds for sums of random matrices*, Found. Comput. Math. 12 (2012), pp. 389–434.
[28] P. F. Velleman and R. E. Welsch, *Efficient computing of regression diagnostics*, Amer. Statist., 35 (1981), pp. 234–242.
[29] T. Wentworth and I. Ipsen, *kappa_SQ: A Matlab Package for Randomized Sampling of Matrices with Orthonormal Columns*, arXiv:1402.0642, 2014.