2010

TR-2010012: Randomized Preconditioning of Linear Systems of Equations

Victor Y. Pan

Guoliang Qian

Ai-Long Zheng

Randomized Preconditioning of Linear Systems of Equations

Victor Y. Pan[1,2],[a], Guoliang Qian[2],[b], and Ai-Long Zheng[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] azheng-1999@yahoo.com

Abstract

Effective preconditioners are known for some important but special classes of matrices. In contrast our properly scaled randomized additive preprocessing and augmentation are likely to precondition any $n \times n$ ill conditioned matrix A that has a small positive numerical nullity $r \ll n$, that is lies near a well conditioned matrix $\tilde{A}$ of rank $n - r$ for a small positive integer r. Both our randomized additive preprocessing and augmentation are likely to accelerate by roughly the factor n/r the customary cubic time solution of a nonsingular linear system $A\mathbf{y} = \mathbf{b}$ of n equations with a coefficient matrix A from this class. We achieve a similar randomized acceleration of the customary algorithms in the case of sparse or structured (e.g., Toeplitz or multilevel Toeplitz) inputs. Furthermore in the important case where the input matrix has displacement structure of Toeplitz or Hankel type, we avoid randomization and relax the restriction on numerical nullity. Our tests show significant acceleration of the known algorithms also in terms of the CPU time involved. Our preprocessing facilitates some other fundamental matrix computations as well, and our proof of the preconditioning power of our randomized preprocessing can be of independent interest.

Key Words: Linear systems of equations, Randomized preconditioning, Toeplitz matrices, Homotopy continuation methods

1 Introduction

1.1 Some background: conditioning and preconditioning

A matrix A is ill conditioned (requiring representation with a high precision) where its condition number $\mathrm{cond}(A)$ is large (in the context). Otherwise it is well conditioned, and then one can solve a nonsingular linear system of equations $A\mathbf{y} = \mathbf{b}$ faster and more accurately.

This motivates a popular subject of preconditioning (see [A94], [B02], [G97], and the bibliography therein), where one seeks a map $A \Longrightarrow C$ such that $\mathrm{cond}(C) \ll \mathrm{cond}(A)$ and the solution $\mathbf{y} = A^{-1}\mathbf{b}$ is readily expressed via the solution of linear systems of equations with the matrix C.

Of course ill conditioned input must still be processed with a high precision, but preconditioning enables us to confine high precision computations to a small fraction of all required flops. (Here

and hereafter "flop" stands for "arithmetic operation".) The power of the known preconditioning methods, however, is limited to the important but rather special matrix classes.

This is also true for the effective preprocessing methods directed to convergence acceleration of popular iterative algorithm (such as the Conjugate Gradient and GMRES algorithms) and based on clustering the singular values of the coefficient matrix [A94], [B02], [G97]. (Hereafter we use the abbreviation "CG" for "Conjugate Gradient".)

1.2 Randomized preconditioning

In contrast to the known techniques our randomized preconditioning is expected to work wherever a nonsingular ill conditioned matrix A has a small positive numerical nullity r, that is has at most r singular values that are small relatively to the norm $||A||$. Such $n \times n$ matrices A can be also identified as small norm perturbations of nearby singular well conditioned matrices $\tilde{A}$ of rank $n - r$ (see more below and in Remark 2.1).

For a matrix A in this class we define a random scaled additive preconditioner P of rank r and prove that with a high probability the matrix $C = A + P$ is nonsingular and well conditioned (see Theorem 5.3).

This extends the *Smoothed Analysis of conditioning* in [SST06], [ST02] to *randomized preconditioning*. Our proofs can be of independent technical interest. The results have prompted applications to various fundamental matrix computations in [PQ10], [PQZa], and [PQZC].

For a better insight, recall that $\mathrm{cond}(A) = \frac{\sigma_1(A)}{\sigma_n(A)}$ where A is a nonsingular $n \times n$ matrix and $\sigma_j(A)$ denotes its jth largest singular value, $j = 1, \dots, n$. We prove that the randomized map $A \Longrightarrow C = A + P$ above is likely to produce a matrix C with $\mathrm{cond}(C)$ of the order $\frac{\sigma_1(A)}{\sigma_{n-r}(A)}$. The latter ratio is not large precisely for the class of matrices A having a numerical nullity at most r.

1.3 Accelerated randomized solution of linear systems of equations

Now suppose that a nonsingular linear system $A\mathbf{y} = \mathbf{b}$ of n equations is ill conditioned, but a matrix $C = A + P$ is nonsingular and well conditioned. Then we can recover the solution $\mathbf{y} = A^{-1}\mathbf{b}$ from the solution of $r + 1$ auxiliary linear systems with the matrix C based on the Sherman–Morrison–-Woodbury formula [GL96, page 50] (hereafter we refer to it as the *SMW formula*).

To avoid error magnification expected in the case of an ill conditioned input, we can compute highly accurate solutions of these $r + 1$ linear systems as follows.

Flowchart 1.1. Randomized Solution of a Linear System with Iterative Refinement

Assume an $n \times n$ matrix A having a small positive numerical nullity r and approximate the solution $\mathbf{y}$ of a linear system $A\mathbf{y} = \mathbf{b}$.

1. *Apply additive preprocessing $A \Longrightarrow C = A + P$ for a random scaled matrix P of the smallest rank r such that the matrix C is expected to be nonsingular and well conditioned. (The integer r is equal to the numerical nullity of the matrix A and is generally unknown, but can be computed by means of binary search. Every search step tests whether the matrix $C = A+P$ is nonsingular and well conditioned for $P = UV^T$, a pair of $n \times q$ random and properly scaled matrices U and V, and a candidate integer q. At most $2\lceil \log_2 r \rceil$ tests are expected to be sufficient.)*

2. *Apply a direct algorithm (say Gaussian elimination) in the IEEE standard single or double precision to compute an approximate inverse $X \approx C^{-1}$.*

3. *Employ this inverse as the basis for iterative refinement to compute sufficiently accurate solutions of $r + 1$ auxiliary linear systems of equations with the matrix C.*

4. *From these accurate auxiliary solutions recover the vector $\mathbf{y} = A^{-1}\mathbf{b}$ via the SMW formula.*

Instead of iterative refinement, one can apply other iterative algorithms such as CG or GMRES that involve no approximate inverse (and thus save flops for its computations) but are more sensitive to the success of preconditioning. Instead of binary search one can compute the numerical nullity r

in a single aggregation–disaggregation step (see [PGMQ, Section 5], [PQ10, Section 6.3], and [PQa, Section 3.3]).

According to our probabilistic estimates in Section 6.5, our randomized algorithms are likely to run faster by the factor $\frac{n}{r \log r}$ than the customary algorithms (such as Gaussian elimination) and to reach nearly optimal time (within polylog factors from the information lower bounds) in the case of a small positive integer r.

1.4 Specializations, variations, testing, and extensions of our algorithms

If an input matrix A (having a small positive numerical nullity) is sparse or structured, we can multiply it by a vector faster. Both customary and our algorithms would run faster by roughly the same factors, and we would yield about the same expected speedup with our algorithms, which would remain nearly optimal (within polylog factors).

In particular $n \times n$ Toeplitz, Hankel, multilevel Toeplitz, and multilevel Hankel matrices can be multiplied by a vector in $O(n \log n)$ flops, based on FFT, whereas in applications to algebraic geometric computations multilevel Toeplitz and Hankel matrices are frequently sparse and can be multiplied by vectors in $O(n)$ flops. (We refer the reader to [BGY80], [BM01], [BMP00], [EP02], [EP10], [MP00], [MPR03], [OOT06], [P98/01], [P01], [VBHK01], and the bibliography therein on computations with these matrices and on some important applications.)

Toeplitz and Hankel (although neither multilevel Toeplitz nor multilevel Hankel) matrices belong to a large and highly important class of matrices with displacement structure. Such a matrix has a small displacement rank and can be represented with displacement generators of a small length d ($d \leq 2$ for Toeplitz and Hankel matrices) [P01]. In the case of such input matrices Flowchart 1.1 can be implemented to support expected acceleration of the customary algorithms by the factor $\frac{n}{d \log r}$ (see Section 6.6); furthermore we can apply the alternative homotopy continuation techniques, which require no randomization and accelerate the customary algorithms by the factor $n/\log^2 n$ with no restriction on numerical nullity of the input.

Some variations enable us to expand and accentuate the power of our algorithms. (a) We can apply augmentation, which is closely linked to additive preprocessing and has similar preconditioning power (cf. Theorem 3.2), but can most perfectly preserve matrix structure. (b) Effective algorithms for a linear system $A\mathbf{y} = \mathbf{b}$ can rely on solving an auxiliary homogeneous linear system or computing a basis for the null space of a nearby singular matrix $\tilde{A} \approx A$ (see Section 10). (c) In the case of a nonsinglar input having a small numerical rank it can be effective to employ our dual variation (3.6) of the SMW formula. (d) In the case of structured inputs, we can combine our techniques with Newton's iteration for matrix inversion to yield some additional speedup (see Remark 8.2 and Sections 6.3 and 9). Furthermore some empirical observations promise substantial progress in applications to Toeplitz and possibly other structured matrices (see the end of Section 9).

The results of our extensive tests (the contribution of the second and third authors) are in good accordance with our theoretical estimates. Our outputs are as accurate as in the case of the customary algorithms, but we outperform these algorithms in terms of the CPU time even in the case of Toeplitz and Hankel inputs (see Table 12.6). Some results of our experiments may be of independent interest, e.g., the demonstration that random Toeplitz matrices tend to be well conditioned.

We focus on preconditioning power of our randomized preprocesing, but the same techniques are expected to regularize symbolic computations in any field, that is to avoid dealing with singular and rank deficient matrices (see Corollaries 4.1, 4.2, 4.4, and 5.2).

Our methods have been extended to a number of fundamental matrix and polynomial computations (see some pointers in Section 12).

1.5 Organization of the paper

We devote the next section to some definitions and basic facts. In Section 3 we present the SMW formula, additive preprocessing, and augmentation. In Sections 4 and 5 we prove that our randomized preprocessing is expected to be preconditioning. In Section 6 we cover extended iterative

refinement and the CG algorithm and estimate the overall randomized complexity of our solution of general, sparse and structured linear systems of equations. In Section 7 we specialize application of our randomized augmentation to a Toeplitz linear system. In Section 8 we present our deterministic continuation algorithm for structured inputs. In Section 9 we briefly recall Newton's iteration for the inversion of structured matrices and propose its heuristic acceleration. Section 10 covers some effective algorithms for linear systems of equations that reduce this task to some computations in null spaces. In Section 11 we present the results of our numerical tests. In Section 12 we give some pointers to various extensions of our methods.

In *selective reading* aimed at the algorithms rather than their formal support, the reader can skip the probability estimates and their proofs. In this case right after Section 3 one can read the short paragraph following the proof of Theorem 5.3 and then go to Sections 6–11, skipping the rest of Sections 4 and 5 and the respective parts of Section 2.

2 Some definitions and basic facts on matrix computations

2.1 Some basic definitions

We use and extend the customary definitions in [GL96], [H02], [S98] on matrix computations.

$\mathbb{C}$ (resp. $\mathbb{R}$) is the field of complex (resp. real) numbers.

A flop is an arithmetic operation with these numbers.

A^T and A^H denote the transpose and the Hermitian transpose of an $m \times n$ matrix A, respectively ($A^H = A^T$ for a real matrix A),

A matrix A is Hermitian if $A = A^H$. A matrix $A = B^H B$ is Hermitian positive definite if B is a nonsingular matrix.

$A^{(k)}$ denotes its $k \times k$ leading (that is northwestern) block submatrix.

A matrix (possibly rectangular) is strongly nonsingular if so are all its leading blocks.

$(B_1, \dots, B_k) = (B_j)_{j=1}^k$ is a $1 \times k$ block matrix with blocks $B_1, \dots, B_k$.

$\mathrm{diag}(B_1, \dots, B_k) = \mathrm{diag}(B_j)_{j=1}^k$ is a $k \times k$ block diagonal matrix with diagonal blocks $B_1, \dots, B_k$.

I_n is the $n \times n$ identity matrix $(\mathbf{e}_j)_{j=1}^n = (\mathbf{e}_1, \dots, \mathbf{e}_n)$,

$O_{k,l}$ is the $k \times l$ matrix filled with zeros. $\mathbf{0}_k$ is the vector $O_{k,1}$.

We drop the subscripts and write I, O, and $\mathbf{0}$ where the size of the matrix or vector is not important or is defined by the context.

2.2 Range, null space, rank, nullity, and nmbs

$\mathcal{R}(A)$ denotes the range of the matrix A, that is the linear space $\{\mathbf{z} : \mathbf{z} = A\mathbf{x}\}$ generated by its columns, $\mathcal{N}(A)$ its null space $\{\mathbf{v} : A\mathbf{v} = \mathbf{0}\}$, $\rho = \mathrm{rank}\, A = \dim \mathcal{R}(A)$ its rank, and $\mathrm{nul}\, A = \dim \mathcal{N}(A)$ its nullity. $\mathbf{v}$ is its null vector if $A\mathbf{v} = \mathbf{0}$.

Suppose a matrix B has full column rank and $\mathcal{R}(B) = \mathcal{N}(A)$. Then we call B a *null matrix basis* or a *nmb* for a matrix A and write $B = \mathrm{nmb}(A)$.

The left nullity, the left null space, left null vectors, and left nmbs of a matrix A are said to be the nullity, the null space, null vectors, and nmbs of the Hermitian transpose matrix A^H, respectively.

2.3 Orthogonalization, norms, and SVD

A matrix $X = A^{(I)}$ is a left (resp. right) inverse of a matrix A if $XA = I$ (resp. $AX = I$). $A^{(I)} = A^{-1}$ for a nonsingular matrix A.

A matrix U is unitary or orthonormal if $U^H U = I$. Formally this definition covers $m \times n$ rectangular matrices U for $m \le n$ but we only use it for square matrices U.

QR factorization $A = QR$ of a matrix A into the product of a unitary matrix Q and an upper triangular matrix R is unique if the R-factor R is a square matrix with positive diagonal entries [GL96, Theorem 5.2.2]. In this case we write $Q = Q(A)$ and $R = R(A)$.

$||A||_h$ is the h-norm, $h = 1, 2, \infty$, and $||A||_F = \sqrt{\sum_{i,j=1}^{m,n} |a_{i,j}|^2}$ is the Frobenius norm of a matrix $A = (a_{i,j})_{i,j=1}^{m,n}$. We write $||A||_2 = ||A||$ and recall that

$$\max_{i,j=1}^{m,n}|a_{i,j}| \le ||A|| = ||A^H|| \le \sqrt{mn}\, \max_{i,j=1}^{m,n}|a_{i,j}|,\ ||A|| \le ||A||_F \le \sqrt{n}\ ||A||. \tag{2.1}$$

$A = S_A \Sigma_A T_A^H$ is an *SVD* or *full SVD* of an $m \times n$ matrix A of a rank ρ provided $S_A S_A^H = S_A^H S_A = I_m$, $T_A T_A^H = T_A^H T_A = I_n$, $\Sigma_A = \mathrm{diag}(\widehat{\Sigma}_A, O_{m-\rho,n-\rho})$, $\widehat{\Sigma}_A = \mathrm{diag}(\sigma_j(A))_{j=1}^{\rho}$, $\sigma_j = \sigma_j(A) = \sigma_j(A^H)$ is the jth largest singular value of a matrix A. These values have the minimax characterization

$$\sigma_j = \max_{\dim(\mathbb{S})=j} \min_{\mathbf{x}\in\mathbb{S},\ ||\mathbf{x}||=1} ||A\mathbf{x}||,\ j = 1, \dots, \rho, \tag{2.2}$$

where $\mathbb{S}$ denotes linear spaces [GL96, Theorem 8.6.1] . They turn into zero, $\sigma_j = 0$, where $j > \rho$. It follows that $\sigma_j(A)$ is the distance from the matrix A to the nearest matrix of a rank $j-1$ for $j = 1, \dots, \rho+1$ and

$$\sigma_1 = \max_{||\mathbf{x}||=1} ||A\mathbf{x}|| = ||A||,\quad \sigma_n = \min_{||\mathbf{x}||=1} ||A\mathbf{x}||. \tag{2.3}$$

If $\sigma_q > \sigma_{q+1}$, so that $q \le \rho$, then the first q columns of the matrices S_A and T_A generate the leading left and right singular spaces $\mathbb{S}_A^{(q)} = \mathcal{R}(S_A(I_q, O_{q,m-q})^T)$ and $\mathbb{T}_A^{(q)} = \mathcal{R}(T_A(I_q, O_{q,n-q})^T)$, respectively, associated with the q largest singular values of the matrix A. The orthogonal complements $\mathbb{S}_{A,m-q}$ and $\mathbb{T}_{A,n-q}$ of these singular spaces are the left and right trailing singular spaces associated with the remaining (smallest) singular values of the matrix A.

$\Sigma_A^+ = \mathrm{diag}((\widehat{\Sigma}_A)^{-1}, O_{n-\rho,m-\rho})$ and $A^+ = T_A \Sigma_A^+ S_A^H$ are the Moore–Penrose generalized (or pseudo) inverses of the matrices Σ_A and A, respectively. A^+ is a left or right inverse of a matrix A of full rank. $A^+ = A^{-1}$ for a nonsingular matrix A. $||A^+|| = 1/\sigma_\rho(A)$ for a matrix A of rank ρ.

2.4 Condition number, perturbation norm bounds, and numerical nullity

Hereafter the concepts "large", "small", "near", "closely approximate", "ill conditioned" and "well conditioned" are quantified in the context.

For two positive parameters a and b we write $a \gg b$ and $b \ll a$ if the ratio a/b is large and write $a \approx b$ if the ratio is close to one or if $b = 0$ and $|a|$ is small. For two matrices A and B we write $A \approx B$ if $||A - B|| \ll ||A||$.

$\mathrm{cond}(A) = \frac{\sigma_1(A)}{\sigma_\rho(A)} = ||A||\ ||A^+||$ is the condition number of a matrix A of a rank ρ. Such a matrix is *ill conditioned* if $\sigma_1(A) \gg \sigma_\rho(A)$ and is *well conditioned* otherwise. See [D83], [GL96, Sections 2.3.2, 2.3.3, 3.5.4, 12.5] [H02, Chapter 15], and [S98, Section 5.3] on effective estimation of norms and condition numbers. $\mathrm{cond}(A) = ||A||\ ||A^{-1}||$ for a nonsingular matrix A.

The next theorem bounds the output perturbation norm of the solution of a linear system of equations via its input perturbation norms and the condition number $\mathrm{cond}(A)$. As the bound ϵ on the input perturbation norm converges to zero, the bound on the output perturbation norm has the order of $2\epsilon\,\mathrm{cond}(A)$.

Theorem 2.1. *(See [H02, Section 7.1, page 121].) Let $A\mathbf{y} = \mathbf{b}$ and $(A + \Delta(A))\tilde{\mathbf{y}} = \mathbf{b} + \Delta(\mathbf{b})$ for a pair of nonsingular matrices A and $A+\Delta(A)$ and two vectors $\mathbf{b}$ and $\Delta(\mathbf{b})$ such that $||\Delta(A)|| \le \epsilon||A||$, $||\Delta(\mathbf{b})|| \le \epsilon||\mathbf{b}||$ for $\epsilon\,\mathrm{cond}(A) < 1$. Then $\frac{||(\tilde{\mathbf{y}}-\mathbf{y})||}{||\tilde{\mathbf{y}}||} \le 2\epsilon\frac{\mathrm{cond}(A)}{1-\epsilon\,\mathrm{cond}(A)}$.*

Backward analysis of rounding errors extends this result as follows (cf., e.g., [GL96, Section 3.4.6], [H02, Theorems 19.5], [S98, Theorem 3.4.9]).

Corollary 2.1. *Gaussian elimination with pivoting uses $\frac{2}{3}n^3 + O(n^2)$ flops with rounding to a precision p to produce an approximate solution $\tilde{\mathbf{y}}$ to a nonsingular linear system $A\mathbf{y} = \mathbf{b}$ of n equations and an approximate inverse $X \approx A^{-1}$ such that $\frac{||\tilde{\mathbf{y}}-\mathbf{y}||}{||\mathbf{y}||} = O(2^{-p}n^2\,\mathrm{cond}(A, \mathbf{b}))$ and $\frac{||X-A^{-1}||}{||A^{-1}||} = O(2^{-p}n^2\,\mathrm{cond}(A))$ as $p \to \infty$.*

Now suppose an $n \times n$ matrix $\tilde{A}$ has a rank ρ and nullity $r = n - \rho$. If such a matrix is well conditioned, then all its sufficiently close neighbours A have numerical rank ρ and numerical nullity r, that is have exactly r singular values that are small relative to the norm $||\tilde{A}||$ (even though almost all of these neiboughrs have rank n and nullity zero). The minimax characterization implies that, conversely, every $n \times n$ matrix A having a positive numerical nullity r is close to a well conditioned singular matrix $\tilde{A}$ of rank $n - r$. The range $\mathcal{R}(\tilde{A})$ (resp. null space $\mathcal{N}(\tilde{A})$) of the latter matrix approximates the leading (resp. *trailing*) singular space $\mathbb{T}_A^{(n-r)}$ (resp. $\mathbb{T}_{A,r}$) of the former matrix associated with its $n - r$ largest (resp. r smallest) singular values.

r is also the left numerical nullity of a matrix A, whose left leading (resp. trailing) singular space $\mathbb{S}_A^{(n-r)}$ (resp. $\mathbb{S}_{A,r}$), associated with its $n - r$ largest (resp. r smallest) singular values, is approximated by the range (resp. null space) of the matrix $\tilde{A}^H$.

Remark 2.1. *Unlike the nullity and the rank, numerical nullity and numerical rank are not well defined for a large class of ill conditioned matrices, in particular for all matrices A having nested clusters of small singular values but also for the matrix class represented by a 1000×1000 matrix A with singular values $\sigma_j(A) = 2^{1000-j}$, $j = 1, 2, \dots, 1000$.*

3 The SMW formula, additive preprocessing, and augmentation

Theorem 3.1. *Suppose $A \in \mathbb{C}^{n\times n}$, $U, V \in \mathbb{C}^{n\times r}$, the matrix $C = A + UV^H$ is nonsingular, $G = I_r - V^H C^{-1} U$, and $0 < r < n$. (The matrix G is called Gauss transform and Schur complement [GL96].) Then we have factorizations*

$$\begin{pmatrix} C & U \\ V^H & I_r \end{pmatrix} = \begin{pmatrix} I_n & U \\ O_{r,n} & I_r \end{pmatrix} \begin{pmatrix} A & O_{n,r} \\ O_{r,n} & I_r \end{pmatrix} \begin{pmatrix} I_n & O_{n,r} \\ V^H & I_r \end{pmatrix} \tag{3.1}$$

and

$$\begin{pmatrix} C & U \\ V^H & I_r \end{pmatrix} = \begin{pmatrix} I_n & O_{n,r} \\ V^H C^{-1} & I_r \end{pmatrix} \begin{pmatrix} C & O_{n,r} \\ O_{r,n} & G \end{pmatrix} \begin{pmatrix} I_n & C^{-1}U \\ O_{r,n} & I_r \end{pmatrix}. \tag{3.2}$$

Furthermore if the matrix A is nonsingular, then so is the matrix G, and we have the SMW formula $A^{-1} = (C - UV^H)^{-1} = C^{-1} + C^{-1}UG^{-1}V^H C^{-1}$.

Proof. The claimed factorizations and nonsigularity are readily verified. The SMW formula (cf. [GL96, page 50], [S98, Corollary 4.3.2]) follows if we invert factorization (3.1). □

We have the straightforward bound

$$||G|| \le ||V||\ ||C^{-1}||\ ||U|| + 1. \tag{3.3}$$

Furthermore, by inverting factorization (3.2) we obtain that

$$||G^{-1}|| \le \max\{||A^{-1}||, 1\}(||V||\ ||C^{-1}|| + 1)(||U||\ ||C^{-1}|| + 1)(||U|| + 1)(||V|| + 1). \tag{3.4}$$

Post-multiply the SMW formula by a vector $\mathbf{b}$ and obtain $A^{-1}\mathbf{b} = C^{-1}\mathbf{b} + C^{-1}UG^{-1}V^H C^{-1}\mathbf{b}$ for $G = I_r - V^H C^{-1}U$. Substitute $W_U = C^{-1}U$ and $\mathbf{w_b} = C^{-1}\mathbf{b}$ and obtain $A^{-1}\mathbf{b} = \mathbf{w_b} + W_U G^{-1} V^H \mathbf{w_b}$ for $G = I_r - V^H W_U$. This reduces the solution of the linear system $A\mathbf{y} = \mathbf{b}$ essentially to the solution of the matrix equation $CW_U = U$ and the linear system $C\mathbf{w_b} = \mathbf{b}$, computing the above matrix G, and its inversion.

We can combine the equations $CW_U = U$ and $C\mathbf{w_b} = \mathbf{b}$ into the single matrix equation

$$CW = (U, \mathbf{b}) \text{ for } W = (W_U, W_\mathbf{b}). \tag{3.5}$$

By applying the SMW formula to the matrix A^{-1} and a pair of matrices $U_-, V_- \in \mathbb{C}^{n\times q}$ for $0 < q < n$, we obtain the *dual SMW formula*

$$A^{-1} = (C_-)^{-1} - U_- V_-^H \text{ for } C_- = A - AU_- H^{-1} V_-^H A \text{ and } H = I_q + V_-^H A U_- \tag{3.6}$$

provided that H and $(C_-)^{-1}$ are nonsingular matrices.

Both SMW and dual SMW formulae can be extended to the case of rectangular matrices A [PQ10]. In the case of square matrices A they have the following determinantal versions,

$$\det A = (\det C)\det G = (\det C_-)\det H. \tag{3.7}$$

To deduce the former equation equate the right hand sides of equations (3.1) and (3.2) and recall that the determinant of a unit triangular matrix equals one. Apply this equation in (3.6) to the expression $(C_-)^{-1} = A^{-1} - U_- V_-^H$ replacing $C = A + UV^H$ and obtain the latter equation in (3.7).

In Section 5 we prove that additive preprocessing $A \Longrightarrow C = A + UV^H$ is expected to decrease the condition number $\operatorname{cond} A = \frac{\sigma_1(A)}{\sigma_n(A)}$ to the level of $\frac{\sigma_1(A)}{\sigma_{n-r}(A)}$ (that is $\operatorname{cond} C$ is expected to be at this level) provided U and V are $n \times r$ Gaussian random matrices with the mean zero and a variance of the order $||A||^2$, so that the map $A \Longrightarrow C$ is expected to precondition an $n \times n$ matrix A if $\frac{\sigma_1(A)}{\sigma_n(A)} \gg \frac{\sigma_1(A)}{\sigma_{n-r}(A)}$.

These properties are extended to the southeastern augmentation $A \Longrightarrow K = \begin{pmatrix} A & U \\ WV^H & W \end{pmatrix}$ where $W \in \mathbb{C}^{r\times r}$, $U, V \in \mathbb{C}^{n\times r}$, $K \in \mathbb{C}^{(n+r)\times(n+r)}$, the matrix A is nonsingular, U, V, and W are random scaled matrices, and $0 < r < n$. The extension can be proved directly [PQa] or via linking augmentation to additive preprocessing in the following simple factorizations or the alternative factorizations in [PQ10], [PQa].

Theorem 3.2. *Suppose* $K = \begin{pmatrix} A & -U \\ WV^H & W \end{pmatrix} \in \mathbb{C}^{(m+r)\times(n+r)}$, $W \in \mathbb{C}^{r\times r}$ *is a nonsingular matrix,* $C = A + UV^H$. *Then* $K = \operatorname{diag}(I_m, W)\widehat{U}\operatorname{diag}(C, I_r)\widehat{V}$, *for* $\widehat{U} = \begin{pmatrix} I_m & -U \\ O_{r,m} & I_r \end{pmatrix}$, $\bar{U} = \widehat{U}^{-1} = \begin{pmatrix} I_m & U \\ O_{r,m} & I_r \end{pmatrix}$, $\widehat{V} = \begin{pmatrix} I_n & O_{n,r} \\ V^H & I_r \end{pmatrix}$, $\bar{V} = \widehat{V}^{-1} = \begin{pmatrix} I_n & O_{n,r} \\ -V^H & I_r \end{pmatrix}$. *Moreover if the matrices* C *and* K *are square and nonsingular, then we have* $K^{-1} = \bar{V}\operatorname{diag}(C^{-1}, I_r)\bar{U}\operatorname{diag}(I_m, W^{-1})$ *and consequently* $C^{-1} = (I_n, O_{n,r})K^{-1}\operatorname{diag}(I_m, W)(I_m, O_{m,r})^T$.

One can similarly employ the northwestern augmentation

$$A \Longrightarrow \begin{pmatrix} W & WV^H \\ -U & A \end{pmatrix} = \begin{pmatrix} O_{r,m} & I_r \\ I_m & O_{m,r} \end{pmatrix} K \begin{pmatrix} O_{n,r} & I_n \\ I_r & O_{r,n} \end{pmatrix} \tag{3.8}$$

as well as northeastern and southwestern augmentations.

In [PQa] the preconditioning property is proved for the more general class of augmentations $K = \begin{pmatrix} A & U \\ V^H & W \end{pmatrix}$ where $U \in \mathbb{C}^{m\times r}$, $V \in \mathbb{C}^{n\times r}$, and $W \in \mathbb{C}^{r\times r}$ are scaled random matrices. In this case if $m = n$, then we have

$$\det K = (\det A)\det(W - V^H A^{-1} U). \tag{3.9}$$

Our next theorem employs additive preprocessing to compute some matrix bases for the null space and the left null space of a singular matrix $\tilde{A}$ with a nullity r. In virtue of Theorem 3.2 we can apply augmentation instead of additive preprocessing. Theorem 3.4 extends these results to approximate the right and left trailing (resp. leading) singular spaces $\mathbb{T}_{A,r}$ and $\mathbb{S}_{A,r}$ (resp. $\mathbb{T}_A^{(q)}$ and $\mathbb{S}_A^{(q)}$) associated with the $r = n - q$ smallest (resp. q largest) singular values of an $n \times n$ nonsingular matrix $A \approx \tilde{A}$ that has a positive numerical nullity r (cf. Section 2.4).

Theorem 3.3. *Assume that* $\tilde{C} = \tilde{A} + UV^H$, $\tilde{A}, \tilde{C} \in \mathbb{C}^{n\times n}$, $n > r > 0$, $r = \operatorname{nul}\tilde{A} = n - \operatorname{rank}\tilde{A}$, $U, V \in \mathbb{C}^{n\times r}$, *and the matrix* $\tilde{C}$ *is nonsingular. Then the matrix* $\tilde{C}^{-1}U$ *is a* $\operatorname{nmb}(\tilde{A})$, *whereas the matrix* $\tilde{C}^{-H}V$ *is a left* $\operatorname{nmb}(\tilde{A})$.

Proof. See [PQ10, Theorem 3.1], where the theorem is deduced in the more general case of rectangular input matrices A. □

Theorem 3.4. *(a) Assume that an $n \times n$ nonsingular matrix A has a numerical nullity r (that is the ratio $\sigma_1(A)/\sigma_{n-r+1}(A)$ is large, but the ratio $\sigma_1(A)/\sigma_{n-r}(A)$ is not large), $n > r > 0$, U and V are $n \times r$ matrices, and the matrix $C = A + UV^H$ is nonsingular. Then there exist a scalar c independent of A, U, V, n and r and two matrices B_U and B_V such that $\mathcal{R}(B_U) = \mathbb{T}_{A,r}$, $\mathcal{R}(B_V) = \mathbb{S}_{A,r}$, $||C^{-1}U - B_U|| \le c\sigma_{n-r+1}(A)||B_U||$, and $||C^{-H}V - B_V|| \le c\sigma_{n-r+1}(A)||B_V||$.*

(b) Furthermore suppose U_- and V_- are $n \times q$ matrices for $q = n - r$ and the matrix $(C_-)^{-1} = A^{-1} + U_-V_-^H$ is nonsingular. Then there exist two matrices B_{U_-} and B_{V_-} and a scalar c_- independent of A, U_-, V_-, n and q such that $\mathcal{R}(B_{U_-}) = \mathbb{T}_A^{(q)}$, $\mathcal{R}(B_{V_-}) = \mathbb{S}_A^{(q)}$, $||C_-U_- - B_{U_-}|| \le c_-\sigma_{n-q+1}(A)||B_{U_-}||$, and $||C_-^HV_- - B_{V_-}|| \le c_-\sigma_{n-q+1}(A)||B_{V_-}||$.

Proof. See [PQ10, Section 7]. □

Part (a) (resp. (b)) of the theorem shows that $\mathcal{R}(C^{-1}U) \approx \mathbb{T}_{A,r}$ and $\mathcal{R}(C^{-H}V) \approx \mathbb{S}_{A,r}$ (resp. $\mathcal{R}(C_-U_-) \approx \mathbb{T}_A^{(q)}$ and $\mathcal{R}(C_-^HV_-) \approx \mathbb{S}_A^{(q)}$), that is, the linear spaces $\mathcal{R}(C^{-1}U)$ and $\mathcal{R}(C^{-H}V)$ (resp. $\mathcal{R}(C_-U_-)$ and $\mathcal{R}(C_-^HV_-)$) approximate the right and left trailing (resp. leading) singular spaces associated with the r smallest (resp. q largest) singular values of the matrix A.

4 Auxiliary results for proving regularization and preconditioning power of randomized preprocesing

4.1 Singular values of submatrices and matrix products

The two following theorems are used in the proofs of Theorems 4.6 and 4.7, which are the basis for proving our estimates for randomized preconditioning in Theorem 5.3, which implies regularization and preconditioning power of our randomized preprocessing, specified in Corollary 5.2 and Remark 5.1. Theorem 4.1 follows from the minimax characterization (2.2). Theorem 4.2 bounds the rank and condition number of a matrix product AB in terms of the ranks and condition numbers of the matrices A and B.

Theorem 4.1. *Fix two positive integers p and q and assume that A_0 is a $p \times q$ submatrix of a matrix A. Then $\sigma_j(A) \ge \sigma_j(A_0)$ for $j = 1, 2, \dots, \min\{p, q\}$.*

Theorem 4.2. *Let $A \in \mathbb{C}^{m\times r}$ and $B \in \mathbb{C}^{r\times n}$ and write $r_A = \operatorname{rank} A$, $r_B = \operatorname{rank} B$, $r_- = \min\{r_A, r_B\}$ and $r_+ = \max\{r_A, r_B\}$. Let $r_+ = r$. (In particular this holds if at least one of the matrices A and B is nonsingular.) Then* $\operatorname{rank}(AB) = r_-$, $\sigma_{r_-}(AB) \ge \sigma_{r_A}(A)\sigma_{r_B}(B)$ *and* $\operatorname{cond}(AB) \le (\operatorname{cond}(A))\operatorname{cond}(B)$.

Proof. Let $M = S_M\Sigma_M T_M^H$ be the full SVD where $\Sigma_M = \operatorname{diag}(\widehat{\Sigma}_M, O_{s,t})$, $\widehat{\Sigma}_M = \operatorname{diag}(\sigma_j(M))_{j=1}^{r_M}$ for $M = A$, $s = m - r_A$, $t = r - r_A$ as well as for $M = B$, $s = r - r_B$, $t = n - r_B$. Let the matrix $\widehat{A} \in \mathbb{C}^{r_A\times r}$ (resp. $\widehat{B}^H \in \mathbb{C}^{r_B\times r}$) be obtained by deleting the zero rows of the matrix $\Sigma_A T_A^H = S_A^H A$ (resp. $\Sigma_B^T S_B^H = T_B^H B^H$), so that $\widehat{A}\widehat{B} \in \mathbb{C}^{r_A\times r_B}$. This pruning keeps all singular values (and therefore the ranks) of the matrices A, B, and AB intact. Clearly $\widehat{A}$ and $\widehat{B}$ are full rank matrices. Furthermore the equation $r_+ = r$ implies that at least one of the matrices $\widehat{A}$ and $\widehat{B}$ is nonsingular and the product $\widehat{A}\widehat{B}$ has full rank $r_- = \operatorname{rank}(AB)$.

Suppose $r_A = r$. Then $m \ge r_A = r_+ = r \ge r_B$. Minimax characterization (2.2) implies that $\sigma_{r_-}(\widehat{A}\widehat{B}) = ||\widehat{A}\widehat{B}\mathbf{x}||$ for some vector $\mathbf{x}$ such that $||\mathbf{x}|| = 1$. We have $\widehat{B}\mathbf{x} \ne \mathbf{0}$ because $\widehat{B} \in \mathbb{C}^{r\times r_B}$. Write $\mathbf{y} = \widehat{B}\mathbf{x}/||\widehat{B}\mathbf{x}||$, $\sigma_A = ||\widehat{A}\mathbf{y}||$, and $\sigma_B = ||\widehat{B}\mathbf{x}||$, and obtain that $\sigma_{r_-}(\widehat{A}\widehat{B}) = \sigma_A\sigma_B$. Since $\widehat{A} \in \mathbb{C}^{r_A\times r} = \mathbb{C}^{r_A\times r_A}$, we can apply equation (2.3) for $A = \widehat{A}$ and $n = r_A$ to obtain that $\sigma_{r_A}(\widehat{A}) = \min_{||\mathbf{z}||=1} ||\widehat{A}\mathbf{z}||$. Consequently $\sigma_{r_A}(\widehat{A}) = \sigma_{r_A}(A) \le \sigma_A$. Likewise $\widehat{B} \in \mathbb{C}^{r\times r_B}$, and so we can apply equation (2.3) for $A = \widehat{B}$ and $n = r_B$ to obtain that $\sigma_{r_B}(\widehat{B}) = \min_{||\mathbf{z}||=1} ||\widehat{B}\mathbf{z}||$. Therefore $\sigma_{r_B}(\widehat{B}) = \sigma_{r_B}(B) \le \sigma_B$. It follows that $\sigma_{r_-}(AB) = \sigma_{r_-}(\widehat{A}\widehat{B}) = \sigma_A\sigma_B \ge \sigma_{r_A}(A)\sigma_{r_B}(B)$.

If $r_A < r_+ = r = r_B$, then the same argument shows that $\sigma_{r_-}(AB) = \sigma_{r_-}(B^HA^H) \ge \sigma_{r_A}(B^H)\sigma_{r_B}(A^H) = \sigma_{r_A}(A)\sigma_{r_B}(B)$.

It follows that $\operatorname{cond}(AB) \le (\operatorname{cond}(A))\operatorname{cond}(B)$ because $||AB|| \le ||A||\,||B||$. □

Remark 4.1. cond(AB) *can be arbitrarily large even for* $m \times r$ *submatrices* A^H *and* B *of* $m \times m$ *unitary matrices if* $m > r$.

4.2 Random sampling and random matrices

$|\Delta|$ is the cardinality of a set Δ in any fixed ring. *Random sampling* of elements from a set Δ is their selection from this set at random and independently of each other. A matrix is *random* if its entries are randomly sampled from a fixed set Δ. Random sampling is *uniform* if it is done under the uniform probability distribution on the set Δ.

Recall that 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 4.1. *[DL78], [S80], [Z79]. For a set* Δ *of cardinality* $|\Delta|$ *(in any fixed ring), let a polynomial in* m *variables have a total degree* d*, and let it not vanish identically on this set. Then the polynomial vanishes in at most* $d|\Delta|^{m-1}$ *points.*

Lemma 4.1 implies that a fixed nonvanishing polynomial vanishes with a probablity converging to zero if the values of its variables are sampled under any reasonable probability distribution on the set Δ whose cardinality converges to infinity. Under the uniform probability distribution the probability is estimated most readily.

Corollary 4.1. *Under the assumptions of Lemma 4.1 let the values of the variables of the polynomial be randomly and uniformly sampled from the set* Δ*. Then the polynomial vanishes with a probability at most* $\frac{d}{|\Delta|}$.

Corollary 4.2. *Let the entries of an* $m \times n$ *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* $k \le l$ *is singular with a probability at least* $1 - k/|\Delta|$ *and (b) is strongly nonsingular with a probability at least* $1 - \sum_{i=1}^{k} i/|\Delta| = 1 - (k+1)k/(2|\Delta|)$*. Furthermore (c) if the submatrix* M *is indeed nonsingular, any entry of its inverse is nonzero with a probability at least* $1 - (k-1)/|\Delta|$.

Proof. The claimed bounds hold for generic matrices. The singularity of a $k \times k$ matrix means that its determinant vanishes, but the determinant is a polynomial of the total degree k in the entries. This 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 the latter entry is (up to sign) a $k \times k$ subdeterminant of the input M, and so it is a polynomial of degree $k - 1$ in its entries. □

Definition 4.1. $F_X(y) = Probability\{X \le y\}$ *for a real random variable* X *is the* cumulative distribution function (CDF) *of* X *evaluated at* y*.* $F_A(y) = F_{\sigma_l(A)}(y)$ *for an* $m \times n$ *matrix* A *and an integer* $l = \min\{m, n\}$*. A matrix (resp. vector) is a* Gaussian random matrix *(resp. vector) with a mean* μ *and a variance* σ^2 *if it is filled with independent Gaussian random variables, all having the same mean* μ *and variance* σ^2*. If* $\mu = 0$ *and* $\sigma^2 = 1$*, this is a* standard Gaussian random matrix *(resp. vector).* $F_{\mu,\sigma}(y) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{y} \exp(-\frac{(x-\mu)^2}{2\sigma^2})dx$ *is the CDF for a Gaussian random variable with a mean* μ *and a variance* σ^2.

4.3 Conditioning of random matrices and randomized matrix products

Gaussian random matrices (cf. Definition 4.1) tend to be well conditioned [D88], [E88], and even perturbations by such a matrix A is expected to make a matrix M well conditioned if the norms $||A||$ and $||M||$ have the same order [SST06], [ST02]. Let us recall some relevant results beginning with a lower bound on the probability that $||A|| \le y$ for a scalar y and a Gaussian random matrix A. This can be viewed as a probabilistic upper bound on the norm of a Gaussian random matrix A.

Theorem 4.3. *(See [DS01, Theorem II.7].) Suppose* $A \in \mathbb{R}^{n \times n}$ *is a Gaussian random matrix with mean zero and a variance* σ^2*. Then* $F_{||A||}(y) \ge 1 - \exp(-x^2/2)$ *for* $x = y/\sigma - 2\sqrt{n} \ge 0$.

Next we recall an upper bound on the probability that the smallest singular value of a matrix $W = A + M$ is less than a scalar y (for a Gaussian random matrix A). This can be viewed as a probabilistic lower bound on the smallest singular value of the matrix W.

Theorem 4.4. *(See [SST06, Theorem 3.3].) Suppose $M \in \mathbb{R}^{m\times n}$, $\bar{U} \in \mathbb{R}^{m\times m}$, and $\bar{V} \in \mathbb{R}^{n\times n}$ are three fixed matrices, $\bar{U}$ and $\bar{V}$ are unitary matrices, $A \in \mathbb{R}^{m\times n}$ is a Gaussian random matrix independent of the matrix M and having mean zero and a variance σ^2, $W = \bar{U}(A + M)\bar{V}$, $l = \min\{m, n\}$, and $y \geq 0$. Then $F_W(y) \leq 2.35\, y\sqrt{l}/\sigma$.*

Now we recall a lower bound on the probability that the condition number $\mathrm{cond}(W)$ is at most y, which can be viewed as a probabilistic upper bound on the condition number $\mathrm{cond}(W)$.

Theorem 4.5. *(See [SST06, Theorem 3.1].) Under the assumptions of Theorem 4.4, let $||M|| \leq \sqrt{l}$. Then $F_{\mathrm{cond}(W)}(y) \geq 1 - (14.1 + 4.7\sqrt{(2\ln y)/n})n/(y\sigma)$ for all $y \geq 1$.*

On further improvement of this bound by the factor of $\sqrt{\log n}$, see [W04].

Next we extend the estimates of Theorem 4.4 to yield probabilistic lower bounds on the smallest singular values of the products of fixed and random matrices.

Theorem 4.6. *Suppose $G \in \mathbb{R}^{q\times m}$ and $H \in \mathbb{R}^{n\times r}$ are two fixed matrices, $r_G = \mathrm{rank}\, G = m$, $r_H = \mathrm{rank}\, H = n$, a random matrix $W \in \mathbb{R}^{m\times n}$ has full rank with probability one, and $y \geq 0$. Then (a) $F_{GW}(y) \leq F_W(y/\sigma_{r_G}(G))$, whereas (b) $F_{WH}(y) \leq F_W(y/\sigma_{r_G}(H))$.*

Proof. At first suppose $m \leq n$. Apply Theorem 4.2 for $A = G$, $B = W$, and consequently for $r_A = r_G = m$ and $r_B = r_W = r_+ = n$. Obtain that $\mathrm{rank}(GW) = \mathrm{rank}\, G = \mathrm{rank}\, W = m$, $\sigma_m(GW) \geq \sigma_m(G)\sigma_m(W)$, and so Probability$\{\sigma_m(GW) \leq y\} \geq$ Probability $\{\sigma_m(G)\sigma_m(W) \leq y\}$ for all positive y. Here G is a fixed matrix, W is a random matrix, and therefore Probability $\{\sigma_m(G)\sigma_m(W) \leq y\}$ =Probability $\{\sigma_m(W) \leq y/\sigma_m(G)\}$. Substitute this equation into the previous bound and deduce part (a) of the theorem.

Now apply Theorem 4.2 for $A = W$, $B = H$, and conseqently for $r_A = r_W = m$ and $r_B = r_H = r_+ = n$ and obtain that $\mathrm{rank}(HW) = \mathrm{rank}\, W = m$, $\mathrm{rank}\, H = n$, $\sigma_m(WH) \geq \sigma_m(W)\sigma_n(H)$, and so Probability$\{\sigma_m(WH) \leq y\} \geq$ Probability $\{\sigma_m(W)\sigma_m(H) \leq y\}$ for all positive y. Here H is a fixed matrix, W is a random matrix, and therefore Probability $\{\sigma_m(W)\sigma_m(H) \leq y\}$ = Probability $\{\sigma_m(W) \leq y/\sigma_m(H)\}$. Substitute this equation into the previous bound and deduce part (b) of the theorem.

To extend the theorem to the case of $m > n$, apply the above proof to the matrices G^T, H^T, and W^T replacing H, G, and W, respectively. □

In view of Remark 4.1, we cannot merely drop the above assumptions that $r_G = m$ and $r_H = n$, but the next theorem (employing Theorem 4.6) circumvents the problem.

Theorem 4.7. *Suppose $G \in \mathbb{R}^{r_G\times m}$, $H \in \mathbb{R}^{n\times r_H}$, $\mathrm{rank}\, G = r_G < m$, $\mathrm{rank}\, H = r_H < n$, $y \geq 0$, and the assumptions of Theorem 4.4 hold for the matrix $X \in \mathbb{R}^{m\times n}$ replacing W, so that in particular $l = \min\{m, n\}$. Then (a) $F_{GX}(y) \leq 2.35y\sqrt{l}/(\sigma_{r_G}(G)\sigma)$ and (b) $F_{XH}(y) \leq 2.35y\sqrt{l}/(\sigma_{r_H}(H)\sigma)$.*

Proof. Let $\bar{G}$ maximize the value $\sigma_{r_G}(\bar{G})$ among all $r_G \times r_G$ submatrices $\bar{G}$ of the matrix G. Clearly $\sigma_{r_G}(\bar{G}) > 0$, and so the matrix $\bar{G}$ is nonsingular. Write $G = \bar{G}(G_1, I_{r_G})P$, $X^T = (X_1^T, \bar{X}^T)P$, and $GX = \bar{G}(A + M)$ where P is an $m \times m$ permutation matrix, $M = G_1X_1$, and $A = \bar{X}$. The matrix $A + M = G_1X_1 + \bar{X}$ has full rank with probability one because X and consequently $\bar{X}$ are Gaussian random matrices. Now observe that the assumptions of Theorem 4.6 hold where $G = \bar{G}$, $W = A + M$, and m is replaced by r_G. Deduce that $F_{GX}(y) \leq F_{A+M}(y/\sigma_{r_G}(\bar{G}))$. Obtain part (a) of Theorem 4.7 by combining this estimate with the bounds $F_{A+M}(y) \leq 2.35y\sqrt{l}/\sigma$ (implied by Theorem 4.4) and $\sigma_{r_G}(\bar{G}) \leq \sigma_{r_G}(G)$ (implied by Theorem 4.1). Obtain part (b) of Theorem 4.7 by applying its part (a) to the matrices X^T, H^T, and $(XH)^T$ replacing the matrices X, G, and GX, respectively. □

The three following corollaries show the power of Theorems 4.4 and 4.7 but are not used in this paper.

Corollary 4.3. *Suppose k, m, and n are integers, $1 \le k \le n \le m$, G, $H^T \in \mathbb{R}^{m\times n}$, $\operatorname{rank} G = \operatorname{rank} H = n$, $X \in \mathbb{R}^{n\times n}$ is a standard Gaussian random matrix with a mean μ and a variance σ, and $y \ge 0$. Then*

(a) $F_{(GX)^{(k)}}(y) \le 2.35y\sqrt{k}/(\sigma_k((I_k, O_{k,m-k})G)\sigma) \le 2.35y\sqrt{k}/(\sigma_n(G)\sigma)$ *and*

(b) $F_{(XH)^{(k)}}(y) \le 2.35y\sqrt{k}/(\sigma_n(H\begin{pmatrix} I_k \\ O_{m-k,k}\end{pmatrix})\sigma) \le 2.35y\sqrt{k}/(\sigma_n(H)\sigma)$.

Corollary 4.4. *Under the assumptions of Corollary 4.3 let $\sigma \ne 0$. Then with probability one the matrices GX and XH are strongly nonsingular.*

Corollary 4.5. *Under the assumptions of Corollary 4.3 choose two scalars y and z such that $y > 0$ and $z \ge 2\sigma\sqrt{n}$. Then*

(a) $F_{\mathrm{cond}(GX)}(||G||yz) \ge 2 - \exp(\frac{(z-2\sigma\sqrt{n})^2}{2\sigma^2}) - 2.35y\sqrt{k}/(\sigma_n(G)\sigma)$ *for* $k = \operatorname{rank}(GX)$ *and*

(b) $F_{\mathrm{cond}(XH)}(||H||yz) \ge 2 - \exp(\frac{(z-2\sigma\sqrt{n})^2}{2\sigma^2}) - 2.35y\sqrt{k}/(\sigma_n(H)\sigma)$ *for* $k = \operatorname{rank}(XH)$.

Proof. Combine Theorem 4.3 for $y = z$ and part (a) of Corollary 4.3 to obtain $F_{||X||/\sigma_k((GX)^{(k)})}(yz) \ge 2 - \exp(\frac{(z-2\sigma\sqrt{n})^2}{2\sigma^2}) - 2.35y\sqrt{k}/(\sigma_n(G)\sigma)$. Recall that $\sigma_k((GX)^{(k)}) \le \sigma_k(GX)$ in virtue of Theorem 4.1 and that $||(GX)^{(k)}|| \le ||GX|| \le ||G||\,||X||$ and deduce part (a) of Corollary 4.5. Part (b) is proved similarly. □

5 Randomized additive preprocessing is expected to be preconditioning

Suppose $A \in \mathbb{C}^{n\times n}$, $U, V \in \mathbb{C}^{n\times r}$, the matrices A and $C = A + UV^H$ are nonsingular, and $0 < r < n$. (We can assume that $\det A \ne 0$ and the matrices U and V have entries randomly and uniformly sampled from a set of a large cardinality in any fixed ring. Then Lemma 4.1 implies that $\det C \ne 0$ with a high probability.)

Our goal is to employ the results of the previous section to prove that additive preprocessing $A \Longrightarrow C = A + UV^H$ is expected to improve conditioning for quite a general class of ill conditioned matrices A assuming Gaussian random auxiliary matrices U and V with a variance of the order $||A||^2$. We first reduce the study of ill conditioned (that is nearly singular) input matrix A to the case of a nearby singular matrix $\tilde{A}$ and then use its SVD and some factorizations of the auxiliary matrices.

Proceeding orderly, let $\tilde{A} = A + E$ be the matrix of a rank $\rho = n - r$ obtained by zeroing the singular values $\sigma_j(A)$ for $j > n - r$ in the SVD $A = S\Sigma T^H$, so that $||E|| = \sigma_{n-r+1}(A)$. Write $C = A + UV^H$ and $\tilde{C} = \tilde{A} + UV^H = C + E$, assume that the matrices C and $\tilde{C}$ are nonsingular and recall that $\mathrm{cond}(\tilde{C}) \le \frac{1+\delta}{1-\delta\,\mathrm{cond}(C)}\mathrm{cond}(C)$ where $\delta = \frac{||E||}{||C||}$ and $\delta\,\mathrm{cond}(C) < 1$ [GL96, Section 5.5.5]. Next assume that the value δ is small, write $\tilde{C} = \tilde{A} + UV^H$, and in the rest of this section estimate the ratio $\frac{\mathrm{cond}(\tilde{C})}{\mathrm{cond}(\tilde{A})} = \frac{\sigma_{n-r}(\tilde{A})}{\sigma_1(\tilde{A})}\mathrm{cond}(\tilde{C})$, which closely approximates the ratio $\frac{\sigma_{n-r}(A)}{\sigma_1(A)}\mathrm{cond}(C)$. Furthermore, to simplify the notation we drop the character "tilde" and write A and C instead of $\tilde{A}$ and $\tilde{C}$ assuming that $\operatorname{rank} A = n - r$ and $C = A + UV^H$.

The following results are readily verified.

Theorem 5.1. *Let $A = S\Sigma T^H$ be full SVD of an $n \times n$ matrix A of a rank ρ where $\rho < n$, S and T are unitary matrices, S, $T \in \mathbb{C}^{n\times n}$, $\Sigma = \mathrm{diag}(\Sigma_A, O_{r,r})$ is an $n \times n$ diagonal matrix, $r = n - \rho$, and $\Sigma_A = \mathrm{diag}(\sigma_j)_{j=1}^{\rho}$ is the $\rho \times \rho$ diagonal matrix of the positive singular values of the matrix A. Suppose $U \in \mathbb{C}^{n\times r}$, $V \in \mathbb{C}^{n\times r}$, and let the $n \times n$ matrix $C = A + UV^H$ be nonsingular. Write*

$$S^H U = \begin{pmatrix} U_\rho \\ U_r \end{pmatrix},\ T^H V = \begin{pmatrix} V_\rho \\ V_r \end{pmatrix},\ R_U = \begin{pmatrix} I_\rho & U_\rho \\ O & U_r \end{pmatrix},\ R_V = \begin{pmatrix} I_\rho & V_\rho \\ O & V_r \end{pmatrix}$$

where U_r and V_r are nonsingular $r \times r$ matrices. Then $R_U \Sigma R_V^H = \Sigma$, $R_U\,\mathrm{diag}(O_{\rho,\rho}, I_r)R_V^H = S^H U V^H T$, so that

$$C = SR_U\,\mathrm{diag}(\Sigma_A, I_r)R_V^H T^H. \tag{5.1}$$

Corollary 5.1. *Write* $\gamma = \frac{||UV^H||}{||A||}$, $q = ||R_U||\ ||R_V||$ *and* $p = ||R_U^{-1}||\ ||R_V^{-1}||$. *Suppose* $\sigma_{n-r}(A) \le 1 \le \sigma_1(A)$. *Then under the assumptions of Theorem 5.1 we have* $\frac{1}{q}\max\{|1-\gamma|,\ \frac{1}{p}\} \le \frac{\mathrm{cond}(C)}{\mathrm{cond}(A)} \le p\ \min\{1+\gamma, q\}$.

Theorem 5.2. *Under the assumptions of Corollary 5.1, we have*

$$\max\{1, ||U||, ||V||, ||U||\ ||V||\} \le q \le \sqrt{(1+||U||^2)(1+||V||^2)},$$
$$1 \le p^2 \le (1+(1+||U||^2)||U_r^{-1}||^2)(1+(1+||V||^2)||V_r^{-1}||^2).$$

Proof. Combine the equations $R_U^{-1} = \begin{pmatrix} I_\rho & -U_\rho U_r^{-1} \\ O & U_r^{-1} \end{pmatrix}$ and $R_V^{-1} = \begin{pmatrix} I_\rho & -V_\rho V_r^{-1} \\ O & V_r^{-1} \end{pmatrix}$ with the bounds $\max\{||X||, ||Y||\} \le ||(X,Y)|| \le \sqrt{||X||^2 + ||Y||^2}$, which hold for all matrices (X, Y). □

We can readily bound the parameters γ and q from above and below by properly scaling the matrices A, U and V, e.g., $\frac{||C||}{||A||} \le 2$ for $||A|| = ||UV^H||$. It remains to supply probabilistic upper bounds on the norms $||U_r^{-1}|| = \frac{1}{\sigma_r(U_r)}$ and $||V_r^{-1}|| = \frac{1}{\sigma_r(V_r)}$ and consequently on the product $p = ||R_U^{-1}||\ ||R_V^{-1}|| \ge \frac{||C^{-1}||}{||A^+||} = \frac{\sigma_{n-r}(A)}{\sigma_n(C)}$ for a pair of random matrices U and V. To deduce the latter inequality, invert matrix equation (5.1) and obtain that $C^{-1} = TR_V^{-H}\,\mathrm{diag}(\Sigma_A^{-1}, I_r)R_U^{-1}S^H$, and so $||C^{-1}|| \le ||T||\ ||R_V^{-H}||\ ||\,\mathrm{diag}(\Sigma_A^{-1}, I_r)||\ ||R_U^{-1}||\ ||S^H||$. Substitute $||T|| = ||S^H|| = 1$ and $||\,\mathrm{diag}(\Sigma_A^{-1}, I_r)|| = 1/\sigma_{n-r}(A) = ||A^+||$ (recall that $\sigma_{n-r}(A) \le 1$ by assumption), and obtain the claimed upper bound on p.

Theorem 5.3. *Let* U, V, U_r, *and* V_r *denote the four matrices in Theorem 5.1, suppose* $m = n$ *and Theorem 4.7 holds (a) for* $r_G = r$, $G = (O, I_r)S^H$, $X = U$ *(in this case* $GX = U_r$*), and* $\tilde{U} = O$ *as well as (b) for* $r_G = r$, $m = n$, $G = (O, I_r)T^H$, $X = V$ *(in this case* $GX = V_r$*), and* $\tilde{V} = O$. *Then (a)* $F_{U_r}(y) \le 2.35\ y\sqrt{r}/\sigma$ *and (b)* $F_{V_r}(y) \le 2.35\ y\sqrt{r}/\sigma$, *respectively, for* $F_A(y)$ *in Definition 4.1.*

Proof. Apply part (a) of Theorem 4.7 for $r_G = r$, $G = (O, I_r)S^H$, $X = U$, and $\tilde{U} = O$ to obtain that $F_{U_r}(y) \le 2.35y\sqrt{r}/(\sigma_r((O, I_r)S^H)\sigma)$. Then apply part (a) of Theorem 4.7 for $r_G = r$, $m = n$, $G = (O, I_r)T^H$, $X = V$, and $\tilde{V} = O$ to obtain that $F_{V_r}(y) \le 2.35y\sqrt{r}/(\sigma_r((O, I_r)T^H)\sigma)$. Observe that $\sigma_r((O, I_r)S^H) = \sigma_r((O, I_r)T^H) = 1$ because S^H and T^H are unitary matrices. Substitute these equations into the above bounds on $F_{U_r}(y)$ and $F_{V_r}(y)$, and obtain both parts (a) and (b) of Theorem 5.3. □

Corollary 5.2. *Under the assumptions of Theorem 5.3 the matrix* C *is singular with the probability zero.*

Proof. Theorem 5.3 implies that the matrices U_r and V_r are singular with probability zero. Therefore the corollary follows from equation (5.1). □

One can readily deduce from Corollary 4.2 that the matrix C is likely to be nonsingular where the entries of the matrices U and V have been sampled randomly and uniformly from a set of a large cardinality in any ring.

Theorems 5.2 and 5.3 and Corollaries 5.1 and 5.2 together imply that under the assumed randomization and scaling, the matrix C is expected to be nonsingular and to have the condition number of the order $\sigma_1(A)/\sigma_{n-r}(A)$, versus $\mathrm{cond}(A) = \sigma_1(A)/\sigma_n(A)$.

Remark 5.1. *Assume standard Gaussian random matrices* U *and* V *and a matrix* A *scaled so that its norm is neither large nor small. Then also the* $k \times k$ *leading principal blocks* $C^{(k)}$ *of the matrix* C *for all* k, $k = 1, \dots, n$, *are expected to have condition numbers of the order at most* $\sigma_1(A)/\sigma_{n-r}(A)$. *To deduce this property (having algorithmic applications in [PQZa]) first write* $C^{(k)} = (I_k, O)C(I_k, O)^T = (I_k, O)A(I_k, O)^T + U_k V_k^H = A^{(k)} + U_k V_k^H$ *where* $U_k = (I_k, O)U$ *and* $V_k = (I_k, O)V$. *Then write* $\rho_k = \mathrm{rank}\, A^{(k)}$, $k' = k - \rho_k$, $U_{k'} = U_k(I_{k'}, O)^T$, $V_{k'} = V_k(I_{k'}, O)^T$, $U_k = U_{k'} + W_k$, $V_k = V_{k'} + Z_k$, *and* $C'_k = A^{(k)} + U_{k'}V_{k'}^T$. *Next extend factorization (5.1) and other results of this section from the matrix pair* $\{A, C\}$ *to the matrix pair* $\{A^{(k)}, C'_k\}$. *Finally deduce that the transition* $C'_k \longrightarrow C$, *that is addition of the random scaled matrix* $W_k Z_k^H$ *to the matrix* C'_k, *has only limited impact on the condition number* $\mathrm{cond}\, C'_k$ *(cf. [W07, Section 4, case 3]).*

Remark 5.2. *We can preserve the regularization and preconditioning power of the map $A \Longrightarrow C$ even where we choose $U = aV$ for a nonzero scalar a and thus generate fewer random parameters. The choice of $a = 1$ should be avoided where the matrix K is Hermitian positive definite because augmentation $A \Longrightarrow K = \begin{pmatrix} A & U \\ U^H & W \end{pmatrix}$ has no preconditioning power. Indeed in this case the augmentation cannot decrease the condition number due to the Interlacing Property of the eigenvalues of Hermitian matrices [GL96, Theorem 8.1.7] (cf. Remark 8.4).*

Remark 5.3. *Our map $A \Longrightarrow C$ is expected to produce a well conditioned matrix C provided that the ratio $\frac{\sigma_1(A)}{\sigma_{n-r}(A)}$ is not large, but such a ratio is large only for matrices A lying near an algebraic variety of matrices M of dimension r such that $\sigma_{n-r}(M) = 0$. Such a variety includes no matrices where $r = n - 1$ and only a narrow class of matrices A even where $r = 1$.*

Finally apply our analysis in this section to the dual additive preprocessing $A^{-1} \Longrightarrow (C_-)^{-1} = A^{-1} + U_- V_-^H$, where $C_- = A - AU_- H^{-1} V_-^H A$, $U_-, V_- \in \mathbb{C}^{n\times q}$, and $H = I_q + V_-^H A U_-$ (see (3.6)). Deduce that the condition number $\operatorname{cond} C_-$ is expected to have the order $\sigma_1(A)/\sigma_{n-q}(A)$ provided U_- and V_- are standard Gaussian random matrices and the matrix A has been scaled so that the norm $||A^{-1}||$ is neither large nor small. To define such scaling we need a crude estimate for the norm $||A^{-1}||$; we can obtain it at a low cost, e.g., by applying the randomized algorithm in [D83].

6 Iterative refinement, Conjugate Gradient (CG) algorithm, and complexity estimates

The SMW formula reduces the solution of the original nonsingular linear system $A\mathbf{y} = \mathbf{b}$ to $r + 1$ auxiliary linear systems of the form $C\mathbf{w}_i = \mathbf{u}_i$, $i = 0, 1, \ldots, r$, or equivalently to the matrix equation $CW = U_0$ for $U, V \in \mathbb{C}^{n\times r}$, $C = A + UV^H$, and $U_0 = (U, \mathbf{b})$ (cf. (3.5)). For a nonsingular $n \times n$ matrix A having a small positive numerical nullity r and for two scaled $n \times r$ Gaussian random matrices U and V we can expect that the matrix C is nonsingular and well conditioned, in virtue of Theorem 5.3. For such a matrix C, we can solve the above equations efficiently by applying some iterative algorithms. In this section we cover two such algorithms and compare their estimated convergence rate and computational cost.

6.1 Extended iterative refinement and its linear convergence

Given a well conditioned matrix C, we can readily compute its approximate inverse $X \approx C^{-1}$ and an approximate solution $\tilde{\mathbf{w}} = X\mathbf{u}$ to a linear system $C\mathbf{w} = \mathbf{u}$ (see Corollary 2.1). We can also readily compute the residual vector $\mathbf{u} - \tilde{\mathbf{u}} = \mathbf{u} - C\tilde{\mathbf{w}}$ and thus reduce our task to approximating the error vector $\mathbf{w} - \tilde{\mathbf{w}}$ that satisfies the linear system $C(\mathbf{w} - \tilde{\mathbf{w}}) = \mathbf{u} - \tilde{\mathbf{u}}$. Recursively we arrive at the classical iterative refinement. By performing sufficiently many its loops, we can refine the solution as much as we wish.

Next we extend this algorithm to yield an accurate solution of a matrix equation $CW = U_0$.

Algorithm 6.1. Extended iterative refinement. *Assume two integers n and r, such that $0 < r < n$, and three input matrices $C \in \mathbb{C}^{n\times n}$, $U_0 \in \mathbb{C}^{n\times(r+1)}$, and $X \in \mathbb{C}^{n\times n}$ such that $||I - CX|| < 1$, which implies that the matrices C and X are nonsingular ([GL96, Lemma 2.3.3], [S98, Theorem 1.4.18]). Fix an integer $k > 0$ and recursively, with no errors, compute the matrices*

$$\begin{aligned} W_i &\Longleftarrow XU_i = C^{-1}U_i - E_i \text{ and} \\ U_{i+1} &\Longleftarrow U_i - CW_i \end{aligned}$$

for $i = 0, 1, \ldots, k-1$. Output the matrix U_k. Compute and output the matrix

$$\bar{W}_k = W_0 + \cdots + W_k.$$

Theorem 6.1. $C\bar{W}_k = U_0 - CE_k$.

Proof. Recall that $CW_i = U_i - U_{i+1}$, $i = 0, 1, \ldots, k-1$. Sum in i and obtain that $C(W_0 + \cdots + W_{k-1}) = U_0 - U_k$. Substitute $U_k = CW_k + CE_k$ and $\bar{W}_k = W_0 + \cdots + W_k$. □

Theorem 6.2. *$U_{i+1} = (I - CX)U_i = (I - CX)^{i+1}U_0$ for $i = 0, 1, \ldots$.*

Proof. Substitute $W_i = XU_i$ into the matrix equation $U_{i+1} = U_i - CW_i$ and obtain that $U_{i+1} = (I - CX)U_i$. Recursively obtain that $U_{i+1} = (I - CX)^{i+1}U_0$. □

Theorem 6.3. *$U_{i+1} = CE_i$ for all i.*

Proof. Pre-multiply the matrix equation $C^{-1}U_i - W_i = E_i$ by C and add the resulting equation to the matrix equation $U_{i+1} - U_i + CW_i = 0$. □

Corollary 6.1. *$U_0 - C\bar{W}_k = U_{k+1}$, and so $C^{-1}U_0 - \bar{W}_k = C^{-1}U_{k+1} = E_k$.*

Corollary 6.2. *We have*

$$||U_0 - C\bar{W}_k|| = ||U_{k+1}|| \le ||I - CX||^{k+1}||U_0||$$

and

$$||C^{-1}U_0 - \bar{W}_k|| = ||C^{-1}U_{k+1}|| \le ||I - CX||^{k+1}||C^{-1}||\ ||U_0||.$$

Corollary 6.1 shows that the matrix U_{k+1} is the residual of the approximation of the solution $W = C^{-1}U_0$ to the matrix equation $CW = U_0$ by the matrix $\bar{W}_k$, whereas $C^{-1}U_{k+1}$ is the error matrix of this approximation. Corollary 6.2 bounds the respective residual and error norms.

It follows that the approximations $\bar{W}_k$ converge to the solution as $k \to \infty$ with linear rate provided $||I - CX|| = \theta < 1$ and the computations are performed with no errors (e.g., by using the fast advanced algorithms for accurate computation of sums and products in [DH03], [LDB02], [PMQR09], and the references therein).

In the next subsection we estimate that these error-free computations only need to use a reasonably bounded constant precision p, except for the stage of computing the sum $W_0 + \cdots + W_k$, which occupies precision of the order kp. We can, however, represent the sum implicitly, by the summands $W_i = XU_i$, $i = 0, 1, \ldots, k$, each computed with rounding to a fixed precision in $O(p)$.

Furthermore we can relax the requirement of performing multiplications XU_i error-free as long as the computed products equal $\tilde{X}_iU_i$ where $||I - C\tilde{X}_i|| \le \bar{\theta} < 1$ for a fixed constant $\bar{\theta}$ and for all i.

Finally let us comment on the computation of the approximate inverse X and on some alternatives to this operation.

Fact 6.1. *Assume a nonsingular matrix $C \in \mathbb{C}^{n\times n}$ and a positive constant $c < 1$ and let Gaussian elimination with pivoting and with rounding to a precision p be applied to compute a matrix X that approximates the inverse C^{-1} of the matrix C. Then one can ensure the bound $||I - CX|| < 2^{-cp}$ by choosing the precision p of the order of $\log(n\,\mathrm{cond}(C))$.*

Proof. In virtue of Corollary 2.1 (applied for A replaced by C), we can yield the bound $||I - CX|| = O(2^{-p}(n\,\mathrm{cond}(C))^2)$, and Fact 6.1 follows. □

Remark 6.1. *An approximation X to the inverse C^{-1} such that $||I - CX|| = \theta < 1$ for a constant θ close to one can be refined by means of Newton's iteration (9.1) in Section 9 for $X = X_0$.*

Instead of an approximate inverse X one can employ a factorization of the matrix C (e.g., its LU factorization). Having it available one can readily approximate the matrices $C^{-1}U_i$ for $i = 0, 1, \ldots$ by using $O(n^2)$ flops in every loop of Algorithm 6.1.

6.2 Precision bounds in the error-free computation of the residual matrices U_i

Let us prove a uniform upper bound on the precision of computing the residuals U_i for all i. For a binary number $b = \sigma \sum_{k=t}^{s} b_k 2^k$, where $\sigma = 1$ or $\sigma = -1$ and each b_k is zero or one, we write $t(b) = t$, $s(b) = s = \lfloor \log_2 |b| \rfloor$, and $p(b) = s - t + 1$, so that $p(b)$ is the precision in the binary representation of b. For an $n \times n$ matrix $M = (m_{i,j})_{i,j=1}^{n}$ we write $s(M) = \max_{i,j} s(m_{i,j})$, $t(M) = \min_{i,j} t(m_{i,j})$, $p(M) = s(M) - t(M) + 1$. Then

$$\log_2 \frac{||M||}{n} \le s(M) \le \lfloor \log_2 ||M|| \rfloor, \tag{6.1}$$

and the absolute value of each entry of the matrix M is the sum of some powers 2^k for integers k in the range $[t(M), s(M)]$. Equations $U_{i+1} = U_i - CW_i$ imply the following lemma.

Lemma 6.1. *We have $t(U_{i+1}) \ge \min\{t(U_i), t(CW_i)\}$ for all i. Moreover $t(CW_i) \ge t(W_i)$ if the (scaled) matrix C is filled with integers.*

Lemma 6.2. *We have $s(U_{i+1}) \le s(U_i) + \log_2(\theta n)$ for all i.*

Proof. The lemma follows from the bounds (6.1) and Theorem 6.2. □

Write $u_i = ||U_i||$ and $f = \frac{\theta n}{|1-\theta|}$ for $\theta = ||I - CX||$. Deduce from Theorem 6.2 that $u_{i+1} \le \theta u_i$ for $i = 0, 1, \ldots$

Lemma 6.3. *We have $s(U_{i+1}) \le s(CW_i) + \log_2 f$ and $s(U_{i+1}) \le s(W_i) + \log_2(f||C||)$ for $\theta < 1$ and all i.*

Proof. First recall that $u_{i+1} \le \theta u_i$, so that $|u_i - u_{i+1}| \ge |\frac{1}{\theta} - 1| u_{i+1}$. The equation $U_i - U_{i+1} = CW_i$ implies that $||CW_i|| = ||U_i - U_{i+1}|| \ge |u_i - u_{i+1}| \ge |\frac{1}{\theta} - 1| u_{i+1}$. Therefore $u_{i+1} \le \frac{f}{n} ||CW_i|| \le \frac{f||C||}{n} ||W_i||$. Combine these inequalities with bounds (6.1) for $M = U_{i+1}$, $M = CW_i$ and $M = W_i$. □

Corollary 6.3. *a) If $t(U_{i+1}) \ge t(U_i)$, then $p(U_{i+1}) \le p(U_i) + \log_2(\theta n)$.*
b) If $t(U_{i+1}) \ge t(CW_i)$, then $p(U_{i+1}) \le p(CW_i) + \log_2 f$.
c) If $t(U_{i+1}) \ge t(W_i)$, then $p(U_{i+1}) \le p(W_i) + \log_2(f||C||)$.

In virtue of Lemma 6.1, at least one of assumptions a) and b) is always satisfied, and if the matrix C is filled with integers, then so is one of assumptions a) and c) as well.

Corollary 6.4. *Assume that $p(W_i) \le \widehat{p}$ and $p(CW_i) \le \tilde{p}$ for two integers $\widehat{p}$ and $\tilde{p}$ and for all i. Furthermore let $\theta \le \frac{1}{n}$. Then we have the uniform upper bound $\max\{p(U_0), \tilde{p} + \log_2 \frac{n}{n-1}\}$ on the precision $p(U_{i+1})$ of the representation of all matrices U_{i+1} for all i. If the matrix C is filled with integers, then we also have the upper bound $\max\{p(U_0), \widehat{p} + \log_2(\frac{n}{n-1}||C||)\}$. The logarithmic terms disappear from both bounds if $\theta \le \frac{1}{n+1}$ for all i.*

Remark 6.2. *Iterative refinement converges linearly right from the start if $\theta \le \frac{1}{n}$. If, say, $\theta \le 1/2$, then $\lceil \log_2 \log_2 k \rceil$ Newton's loops (9.1) in Section 9 would ensure the bound $\theta \le 1/k$ for any $k \ge 2$.*

The corollary implies that iterative refinement succeeds for a fixed bounded precision p in the case of well conditioned matrices C, which is in good accordance with the vast empirical data accumulated by the users of iterative refinement [GL96, Section 3.5.3].

6.3 The Congujate Gradient (CG) algorithm

Many iterative algorithms for linear systems of equations involve no approximate inverse and thus save flops for its computation. The Congujate Gradient (CG) algorithm below (cf. [G97, Algorithm 2], [GL96, Section 10.2], [TB97, Algorithm 38.1]) is among the most effective ones in this class. It begins with a very crude approximate solution $\mathbf{x}_0 = \mathbf{0}$. Then it recursively selects a direction for improvement and makes a step of the optimal length into this direction.

Algorithm 6.2. The Congujate Gradient (CG) algorithm.

INPUT: *a Hermitian positive definite matrix* $C \in \mathbb{C}^{n\times n}$, *a vector* $\mathbf{b} \in \mathbb{C}^n$, *and a positive scalar* τ.

OUTPUT: *an approximate solution vector* $\mathbf{x}$ *such that* $||C\mathbf{x}-\mathbf{b}|| \le \tau||\mathbf{b}||$.

INITIALIZATION: *Set* $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{b}$, $k = 1$. *Compute the value* $\rho_0 = \mathbf{b}^H\mathbf{b}$.

COMPUTATIONS:

1. *Compute the scalar* $\alpha_k = \frac{\mathbf{r}_{k-1}^H\mathbf{r}_{k-1}}{\mathbf{p}_{k-1}^H C\mathbf{p}_{k-1}}$ *(defining the length of the step from* $\mathbf{x}_{k-1}$ *into the fixed direction* $\mathbf{p}_{k-1}$*).*
2. *Compute the approximate solution vector* $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{p}_{k-1}$.
3. *Compute the residual vector* $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k C\mathbf{p}_{k-1}$ *and the scalar* $\rho_k = \mathbf{r}_k^H\mathbf{r}_k$. *If* $\rho_k \le \tau^2$, *then output the vector* $\mathbf{x} = \mathbf{x}_k$ *and stop.*
4. *Otherwise compute the scalar* $\beta_k = \rho_k/\rho_{k-1}$, *which measures the improvement at the* k*-th CG loop.*
5. *Compute the vector* $\mathbf{p}_k = \mathbf{r}_k + \beta_k\mathbf{p}_{k-1}$ *of the new search direction, set* $k \leftarrow k+1$, *and go to Stage 1.*

Let $\mathcal{R}(\mathbf{v}_1,\dots,\mathbf{v}_h)$ denote the linear space spanned by the linear combinations of the vectors $\mathbf{v}_1,\dots,\mathbf{v}_h$ (so that $\mathcal{R}(\mathbf{v}_1,\dots,\mathbf{v}_h) = \mathcal{R}(V)$ for $V = (\mathbf{v}_1,\dots,\mathbf{v}_h)$) and let $\mathbb{K}_k(C,\mathbf{b})$ denote the Krylov space $\mathcal{R}(\mathbf{b}, C\mathbf{b},\dots,C^{k-1}\mathbf{b})$. The CG algorithm has the remarkable properties that $\mathbf{r}_j^H\mathbf{r}_k = \mathbf{p}_j^H C\mathbf{p}_k = 0$ for $j < k$ and that for all k the vector $\mathbf{y} = \mathbf{x}_k$ minimizes the C-norm of the error function $e_C(\mathbf{y}) = (\mathbf{y}-\mathbf{x})^H C(\mathbf{y}-\mathbf{x}) = \mathbf{y}^H C\mathbf{y} - 2\mathbf{y}^H\mathbf{b} + \mathbf{x}^H\mathbf{b}$ over all vectors $\mathbf{y} \in \mathbb{K}_k(C,\mathbf{b})$; moreover for all k we have $\mathbb{K}_k(C,\mathbf{b}) = \mathcal{R}(\mathbf{x}_1,\dots,\mathbf{x}_k) = \mathcal{R}(\mathbf{p}_0,\dots,\mathbf{p}_{k-1}) = \mathcal{R}(\mathbf{r}_0,\dots,\mathbf{r}_{k-1})$ [G97], [GL96], [TB97].

Correctness of the CG algorithm follows from the equations $\mathbf{r}_k = \mathbf{b} - C\mathbf{x}_k$ that hold for all k and from the fact that $\mathbf{r}_k = \mathbf{0}$ for $k \ge n$ [G97], [GL96], [TB97]. The latter equations mean convergence in at most n CG loops. This property is proved assuming error-free computations in the exact arithmetic. Rounding errors generally invalidate this upper bound n on the number of the CG loops required for convergence, but one can employ the following bound instead,

$$e_C(\mathbf{x}_k) \le 2\frac{\sqrt{\mathrm{cond}(C)}-1}{\sqrt{\mathrm{cond}(C)}+1}e_C(\mathbf{x}_{k-1}) \le 2\big(\frac{\sqrt{\mathrm{cond}(C)}-1}{\sqrt{\mathrm{cond}(C)}+1}\big)^{k-1}e_C(\mathbf{x}_0) \text{ for all } k. \tag{6.2}$$

Generally this upper bound is sharp, but for some classes of inputs the CG algorithm converges faster. The following more fundamental estimate implies bounds (6.2),

$$e_C(\mathbf{x}_k) \le \min_{p_{k-1}(x)\in\mathbf{P}_{k-1}} \max_{j=1,\dots,n}(1+\sigma_j(C)|p_{k-1}(\sigma_j(C))|)e_C(\mathbf{x}_{k-1}). \tag{6.3}$$

Here $\mathbf{P}_{k-1}$ is the class of polynomials $p_{k-1}(x)$ of degree at most $k-1$.

One can deduce from (6.3) that $e_C(\mathbf{x}_k) \le 2(\frac{\sqrt{\sigma_h(C)/\sigma_n(C)}-1}{\sqrt{\sigma_h(C)/\sigma_n(C)}+1})^{k-h}e_C(\mathbf{x}_0)$ for $h = 1,2,\dots,k$ provided $\sigma_1(C) \ge \sigma_2(C) \ge \cdots \ge \sigma_n(C)$ (cf. [G97]).

The CG algorithm is applied to Hermitian positive definite matrices. One can extend it to general nonsingular matrices M based on the symmetrizations $M \Longrightarrow C = M^H M$ or $M \Longrightarrow C = MM^H$, which can be performed implicitly. In both cases $\mathrm{cond}(C) = (\mathrm{cond}(M))^2$, and bounds (6.2) can be rewritten to express the convergence rate in terms of $\mathrm{cond}(M)$ as follows,

$$e_M(\mathbf{x}_k) \le 2\frac{\mathrm{cond}(M)-1}{\mathrm{cond}(M)+1}e_M(\mathbf{x}_{k-1}) \le 2\big(\frac{\mathrm{cond}(M)-1}{\mathrm{cond}(M)+1}\big)^{k-1}e_M\big(\mathbf{x}_0\big) \text{ for all } k. \tag{6.4}$$

In these implementations the CG loop for general linear system $M\mathbf{y} = \mathbf{b}$ essentially amounts to multiplication of the two matrices M and M^H by two vectors.

Equation (6.4) implies that

$$e_M(\mathbf{x}_k) \le 2e^s\ e_M(\mathbf{x}_0) \text{ for } k = \lceil 0.5(1+\mathrm{cond}(M))h \rceil,\ e = 2.7182818\ldots,\ \text{and } s = 1, 2, \ldots \quad (6.5)$$

We also recall that $||\mathbf{y}|| \le ||M^{-1}||\ ||\mathbf{b}||$ and deduce that

$$e_M(\mathbf{x}_0) = \mathbf{y}^H M \mathbf{y} \le ||M||\ ||M^{-1}||^2 \mathbf{b}^H \mathbf{b}. \quad (6.6)$$

Consequently $e_M(\mathbf{x}_0) \le (\mathrm{cond}(M))^2 \mathbf{b}^H \mathbf{b}$ if the matrix M is normalized so that $||M|| = 1$.

The more general bound (6.3) can be extended to reveal faster convergence of the CG algorithm for some rather special classes of matrices M, in particular for matrices whose all singular values lie in a small number of clusters. Moreover, empirically some other iterative methods such as GMRES exhibit faster convergence for general nonsingular linear systems. In the next sections, however, we apply just the simple bounds in (6.4)–(6.6).

6.4 Comparison of iterative refinement and the CG algorithm; preconditioning with an approximate inverse

Let us assume rounding to a fixed reasonably large precision p (cf. Section 6.2) and compare performances of extended iterative refinement and the CG algorithm applied to general linear system $C\mathbf{y} = \mathbf{b}$.

In both algorithms every loop essentially amounts to multiplication of two matrices by two vectors and takes $O(n^2)$ flops. Unlike iterative refinement, the CG algorithm computes no approximate inverse, thus saving order of n^3 flops (performed in a precision p of the order $\log(n\,\mathrm{cond}(C))$). Both agorithms converge linearly, with the convergence coefficient $2^{cp}/\,\mathrm{cond}(C)$ for a positive constant c in the case of iterative refinement using a precision p of the order $\log_2(n\,\mathrm{cond}\,C)$ (see Fact 6.1), versus $1 - \frac{2}{1+\mathrm{cond}(C)}$ in the case of the CG algorithm. Typically the latter coefficient is by far smaller and convergence of the CG algorithm can be more easily destroyed by rounding errors unless the matrix C is sufficiently well conditioned. Thus preconditioning improves performance of both algorithms but is more critical in the CG case.

If, however, a close approximate inverse $X \approx C^{-1}$ is available, then one can reduce the original linear system $C\mathbf{y} = \mathbf{b}$ to the equivalent linear systems $XC\mathbf{y} = X\mathbf{b}$ or $CX\mathbf{z} = \mathbf{b}$ such that $\mathbf{y} = X\mathbf{z}$ with coefficient matrices XC and CX, respectively, both lying very close to the identity matrix I. In this case both iterative refinement, CG algorithms, GMRES, an even Jacobi iteration very rapidly converge to the solution (cf. [GL96]).

6.5 Computational complexity estimates: the case of general matrices

Recall that $\mathrm{cond}(A) \approx \frac{||\text{INPUT ERROR}||}{||\text{OUTPUT ERROR}||}$ in computing the inverse A^{-1} and the solution vector $\mathbf{y} = A^{-1}\mathbf{b}$ of a linear system $A\mathbf{y} = \mathbf{b}$ (cf. Theorem 2.1). Then in virtue of Corollary 2.1 one needs an input precision of at least

$$p_s \ge p_t + \log(n\,\mathrm{cond}(A, \mathbf{b})) + O(1) \quad (6.7)$$

bits to support the output precision of p_t bits. Therefore at least $0.5mp_s$ bit-operations (e.g., at least $0.5mp_s/p$ flops in a precision p) are required to process the m input values where $m = n^2 + n$ for a nonsingular linear system of n equations with general coefficient matrix.

For comparison, Gaussian elimination solves general nonsingular linear system $A\mathbf{y} = \mathbf{b}$ of n equations by using

$$f_{\mathrm{ge}} = \frac{2}{3}n^3 + O(n^2) \quad (6.8)$$

flops in the precision p_s to produce the output with the precision p_t. The upper bound exceeds the lower bound by the factor n.

Let us compare these bounds with the estimates supported by randomized additive preprocessing $A \Longrightarrow C = A + UV^H$ that yields a well conditioned matrix C (cf. Section 5).

Theorem 6.4. *Assume a nonsingular ill conditioned linear system of n equations $A\mathbf{y} = \mathbf{b}$ where the matrix A has a positive numerical nullity r. Allow preprocessing $A \Longrightarrow C = A + UV^H$ and furthermore assume that*

1. *U and V are $n \times r$ matrices filled with numbers in a low precision p_-,*
2. *C is a nonsingular well conditioned matrix,*
3. *p_t is the output precision,*
4. *p_-, p_s, p_t, and p are four integers such that $\log_2 \operatorname{cond} C < p \le p_s$, $3p_- < p$,*
5. *the precision p supports Fact 6.1 for a fixed positive c (say for $c = 0.2$), and*
6. *p_s is defined by equation (6.7).*

Then one can compute the solution vector $\mathbf{y}$ with the output precision p_t by applying $O(n^3 + rn^2 p_s/p)$ flops in precision p.

Proof. (Outline.) Preprocessing takes $2n^2 r$ flops with low precision numbers (see assumption 1 of the theorem). Perform these flops error-free by using precision p.

Compute the solution $\mathbf{y}$ by applying the SMW formula and iterative refinement.

The computation of an approximate inverse X by a direct algorithm takes $f_{\text{inv}} = O(n^3)$ flops. We can perform them in precision p of the order $\log(n \operatorname{cond}(C))$, ensuring the bound $||I - CX|| \le 2^{-cp}$ (see assumption 5 of the theorem).

Every loop of iterative refinement in Algorithm 6.1 for the $r + 1$ linear systems involves $O(rn^2)$ flops and decreases all the $r+1$ residual norms by at least the factor 2^{-cp}. One can support an output precision $\bar{p}_s$ by using the order of $\bar{p}_s/p$ loops of the algorithm, thus involving $f_{\text{ref}} = O(rn^2 \bar{p}_s/p)$ flops.

Choose a precision $\bar{p}_s$ that ensures approximation of the solution $\mathbf{y} = A^{-1}\mathbf{b}$ within the relative error norm 2^{-p_t}. In the SMW formula we can use precision $\bar{p}_s \approx p_t + \log_2 \operatorname{cond}(A) + 2 \log_2 \operatorname{cond}(C)$ (cf. (3.3) and (3.4)), and then $\bar{p}_s \approx p_s$ because the matrix A is ill conditioned, but the matrix C is well conditioned, so that $\operatorname{cond}(A) \gg \operatorname{cond}(C)$. Consequently $f_{\text{ref}} = O(rn^2 p_s/p)$. □

We compute an approximate inverse X by involving $\frac{2}{3}n^3 + O(n^2)$ flops (cf. (6.8)); overall we use $O(n^3 \log r + rn^2 p_s/p)$ flops in precision p where we are given the numerical nullity r. Taking into account $O(\log r)$ steps of the binary search for it, we increase this estimate to

$$f_{\text{ir}} = O(n^3 \log r + rn^2 p_s/p). \tag{6.9}$$

With the CG algorithms we need

$$f_{\text{CG}} = O((r \log r) n^2 p_s / \log(1 - \frac{2}{1 + \operatorname{cond}(C)})) \tag{6.10}$$

flops in precision p.

The above estimates show that iterative refinement performed with a precision p is superior to the CG algorithm where $p - \lceil \log_2 \operatorname{cond}(C) \rceil \gg \log_2(1 - \frac{2}{1+\operatorname{cond}(C)})$, but the CG algorithm becomes more competitive where the matrix C is well conditioned, e.g., in our algorithms in Section 8 (cf. Remarks 8.1 and 8.2).

Remark 6.3. *Recall that $\mu(q) = O((q \log q) \log\log q)$ bit operations are sufficient for a flop with a precision q [AHU74], [F07]. Then bounds (6.9) and (6.10) extend to $f_{\text{ir}}\mu(p) = O((n^3 \log r + rn^2 p_s/p)\mu(p))$ and $f_{\text{cg}}\mu(p) = O(\mu(p)(r \log r) n^2 p_s / \log(1 - \frac{2}{1+\operatorname{cond}(C)}))$ bit-operations, respectively, versus the order of $n^3 \mu(p_s)$ in the solution by means of Gaussian elimination and the information lower bound $0.5n^2 p_s$.*

6.6 Computational complexity estimates: the case of sparse and special matrices

Our algorithms are intensive in matrix-by-vector mltiplications and therefore perform faster in the case of sparse or structured inputs, for which this operation is fast. To specify the acceleration, let $I(M)$ and $v(M)$ denote the number of flops required for the inversion of a matrix M and its mulltiplication by a vector, respectively. Then we can extend our estimates (6.9) and (6.10) as follows,

$$f_{\rm ir} = O(I(C)\log r + rv(C)p_s/p) \text{ for } p > \log_2 \text{cond}(C), \tag{6.11}$$

$$f_{\rm CG} = O((r\log r)v(C)p_s/\log(1-\frac{2}{1+\text{cond}(C)})). \tag{6.12}$$

Now recall that $v(UV^H) = 4nr-n-r$ for $n\times r$ matrices U and V, $v(A) \le 2f-n$ for an $n\times n$ matrix having f nonzero entries, and $v(A) = O(n\log n)$ for an $n \times n$ multilevel Toeplitz or Hankel matrix. Furthermore the latter bound decreases to $O(n)$ in typical applications to algebraic and geometric computations, due to sparseness of the input [MP00], [EP02]. Consequently our algorithms based on the the CG iterations are dramatically accelerated in the case of such matrices provided they have a small positive numerical nullity r. Generally the inversion of these matrices takes the order of n^3 flops, however, and we cannot yield similar speedup based on iterative refinement.

We have no this restriction for the large and highly important class of matrices having displacement structure. This class includes Toeplitz matrices $T = (t_{i-j})_{i,j=1}^n$, Hankel matrices $H = (h_{i+j})_{i,j=1}^n$, Vandermonde matrices $V(\mathbf{t}) = (t_i^j)_{i=1,j=1}^n$, Cauchy matrices $C(\mathbf{s},\mathbf{t}) = (\frac{1}{s_i-t_j})_{i,j=1}^n$, and matrices with Toeplitz-like, Hankel-like, Vandermonde-like and Cauchy-like structures of similar types. Each of the four classes is associated with displacement operators $M \Longrightarrow L(M)$, $L(M) = M - AMB$ or $L(M) = AM - MB$ for a pair of operator matrices A and B of shift or scaling such that the displacement $L(M)$ has a small rank d, called the displacement rank of M. ($d \le 2$ for Toeplitz and Hankel matrices, $d = 1$ for Vandermonde and Cauchy matrices.)

The $n\times n$ displacement $L(M)$ of a rank d and the matrix M itself can be readily expressed via $2dn$ parameters rather than n^2 entries. Furthermore $v(M) = O(dn\log^h n)$ where $h = 1$ for the structures of Toeplitz and Hankel types, $h \le 2$ for the two other structures (see, e.g., [P01]). Substitute this bound into equations (6.11) and (6.12) and obtain that iterative refinement involves $f_{\rm ref} = O((rdn\log^h n)p_s/p)$ flops in a low precision p (not counting $O(I(C)\log r)$ flops for computing an approximate inverse) and the CG based solution requires $O((r\log r)(dn\log^h n)p_s/\log(1-\frac{2}{1+\text{cond}(C)}))$ flops. In the case of small positive integers d and r this is close to the information lower bound of $0.5np_s/p$ flops in a precision p on processing n input entries represented with a precision $p_s \ge p$.

Inversion of an $n \times n$ matrix M with displacement rank d can be reduced to solving $2d$ linear systems with matrices M and M^T, and so for $r > d$ we can decrease the above bounds by the factor r/d provided the input structure is preserved in the transition $A \Longrightarrow C = A + UV^H$. This requirement, however, restricts the number of random parameters involved to the order $O(dr)$, and the question arises whether our randomized *structured* preprocessing still has regularization and preconditioning power under this restriction. One can prove that the regularization power is extended [PW08], but proving the extension of preconditioning power is an open problem. This is because we cannot extend the Smoothed Analysis in [SST06], [ST02] to proving that random structured matrices tend to be well conditioned. Empirically we observe this property for random Toeplitz and Hankel matrices (see Tables 12.1–12.4).

In Section 8 we avoid these problems by using an alternative deterministic approach.

Finally, low precision computation of the inverse does not dominate the overall arithmetic cost because of the following estimates, $I(M) \le cdn^2$, $I(M) \le c_{\rm large}d^2n\log^2 n$ for a moderate constant c and a considerable constant $c_{\rm large}$ (see [B85], [GKO95], [P01], [P10], [R06], and [VBHK01, Introduction], and the references therein).

7 A randomized Toeplitz solver

Let us specify our approach for the inversion of Toeplitz matrices by using some explicit formulae for the inverse.

Let $J = J_k = (\mathbf{e}_k, \dots, \mathbf{e}_1)$ denote the $k \times k$ reflection matrix, $J^2 = I$, JT and TJ are Hankel (resp. Hankel-like) matrices where T is a Toeplitz (resp. Toeplitz-like) matrix, whereas JH and HJ are Toeplitz (resp. Toeplitz-like) matrices where T is a Hankel (resp. Hankel-like) matrix. Therefore matrix algorithms with Toeplitz (resp. Toeplitz-like) inputs for matrix inversion or solving linear systems of equations can be immediately extended to Hankel (resp. Hankel-like) inputs and vice versa.

An $n \times n$ lower triangular Toeplitz matrix $Z(\mathbf{v})$ is completely defined by its first column $\mathbf{v} = Z(\mathbf{v})\mathbf{e}_1$.

The Gohberg–Semencul formula expresses the inverse T^{-1} of a nonsingular $n \times n$ Toeplitz matrix T via its two columns $T^{-1}\mathbf{e}_1$ and $T^{-1}\mathbf{e}_n$ under the mild restriction that $\mathbf{e}_1^T T \mathbf{e}_1 \neq 0$ (see [GS72], [T90]). This restriction was avoided in the Heinig's modification in [H79] and [HR84].

Here are two alternative formulae from [GS72] (cf. [BGY80, Theorem 7]) and [GK72], which express the inverse T^{-1} via two columns $K^{-1}\mathbf{e}_1$ and $K^{-1}\mathbf{e}_{n+1}$ of the inverse K^{-1} of an $(n+1)\times(n+1)$ Toeplitz matrix K that has T as its block submatrix.

Theorem 7.1. *Suppose $K = (t_{i,j})_{i,j=0}^{n}$ is a nonsingular $(n+1) \times (n+1)$ Toeplitz matrix, write $T = (t_{i,j})_{i,j=0}^{n-1}$, $\widehat{\mathbf{v}} = (v_i)_{i=0}^{n} = K^{-1}\mathbf{e}_1$, $\mathbf{v} = (v_i)_{i=0}^{n-1}$, $\mathbf{v}' = (v_i)_{i=1}^{n}$, $\widehat{\mathbf{w}} = (w_i)_{i=0}^{n} = K^{-1}\mathbf{e}_{n+1}$, $\mathbf{w} = (w_i)_{i=0}^{n-1}$, and $\mathbf{w}' = (w_i)_{i=1}^{n}$. (a) If $v_0 \neq 0$, then the matrix $T = (t_{i,j})_{i,j=0}^{n-1}$ is nonsingular and $v_0 T^{-1} = Z(\mathbf{v})Z^T(J\mathbf{w}') - Z(\mathbf{w})Z^T(J\mathbf{v}')$. (b) If $v_n \neq 0$, then the matrix $T_{10} = (t_{i,j})_{i=1,j=0}^{n,n-1}$ is nonsingular and $v_n T^{-1} = Z(\mathbf{w})Z^T(J\mathbf{v}') - Z(\mathbf{v})Z^T(J\mathbf{w}')$.*

Proof. Part (a) was proved in [GS72], part (b) in [GK72]. □

The first and the last columns of the matrix K^{-1} turn into one another up to reflection, that is $K^{-1}\mathbf{e}_1 = J_{n+1}K^{-1}\mathbf{e}_{n+1}$, where the matrix K is real symmetric because in this case the inverse K^{-1} is both symmetric and persymmetric. Then part (a) of Theorem 7.1 expresses the matrix T^{-1} via the first column of the matrix K^{-1} alone.

Remark 7.1. *For any fixed positive integer q we can embed a nonsingular $n \times n$ Toeplitz matrix T into a nonsingular $(n+q) \times (n+q)$ Toeplitz matrix T_{n+q} that has the $n \times n$ leading principal block T. Then we can recursively apply part (a) of Theorem 7.1 to express the inverse T^{-1} via the two column vectors $T_{n+q}^{-1}\mathbf{e}_1$ and $T_{n+q}^{-1}\mathbf{e}_{n+q}$. We can similarly employ part (b) of the theorem.*

Let us apply Theorem 7.1 to support our randomized augmentation techniques for solving a nonsingular Toeplitz linear system $T\mathbf{y} = \mathbf{b}$ of n equations in the case where the matrix T has numerical nullity one.

To compute the solution vector $\mathbf{y} = T^{-1}\mathbf{b}$, we first embed the matrix T into an $(n+1)\times(n+1)$ Toeplitz matrix $K = \begin{pmatrix} w & \mathbf{v}^T \\ \mathbf{f} & T \end{pmatrix}$. In virtue of the Toeplitz structure of the matrix K we have $w = \mathbf{e}_1^T T \mathbf{e}_1$ and the vectors $\mathbf{f} = (f_i)_{i=1}^{n}$ and $\mathbf{v} = (v_i)_{i=1}^{n}$ are filled with the respective entries of the matrix T, except for the two coordinates f_n and v_n, which we choose at random and then scale to have the ratio $\frac{||K||}{||T||}$ neither large nor small (cf. [GS72]).

In virtue of Corollary 4.2 this policy is likely to produce a nonsingular matrix K whose inverse is likely to have a nonzero entry $\mathbf{e}_1^T K^{-1}\mathbf{e}_1$. These two implications of Corollary 4.2 were in good accordance with our test results, in which the matrix K was also consistently well conditioned (even though we used only two random parameters).

Part (a) of Theorem 7.1 expresses the inverse T^{-1} via the first column $\mathbf{v} = K^{-1}\mathbf{e}_1$ and the last column $\mathbf{w} = K^{-1}\mathbf{e}_{n+1}$ of the inverse matrix K^{-1}.

Summarizing we reduce the solution of the original ill conditioned Toeplitz linear system $T\mathbf{y} = \mathbf{b}$ to computing highly accurate solutions of two linear systems $K\mathbf{x} = \mathbf{e}_1$ and $K\mathbf{z} = \mathbf{e}_{n+1}$, both expected to be well conditioned. (High accuracy is needed to counter magnification of the input and rounding errors, expected in the case of ill conditioned input.)

To solve the two latter systems, we first employ the effective algorithms in [KV99], [V99], [VBHK01], and [VK98] and then apply iterative refinement with double precision. We refer to the resulting algorithm as **Algorithm 7.1**.

For any positive integer $q < n$ we can follow Remark 7.1 to reduce the solution of a nonsingular Toeplitz linear system $T\mathbf{y} = \mathbf{b}$ of n equations to the computation of two columns of the inverse of the associated $(n+q) \times (n+q)$ Toeplitz matrix; we expect that it has condition number of the order $\sigma_1(T)/\sigma_{n-q}(T)$.

One can readily extend this approach to the case of Toeplitz-like, Hankel and Hankel-like inputs.

In the important special case where the Toeplitz matrix T is real symmetric, we can choose real scalar w and a single real vector $\mathbf{f} = \mathbf{v}$ to yield a real symmetric matrix $K = \begin{pmatrix} w & \mathbf{v}^T \\ \mathbf{v} & T \end{pmatrix}$. Then Algorithm 7.1 is simplified because $\mathbf{w} = K^{-1}\mathbf{e}_{n+1} = J_{n+1}\mathbf{v} = J_{n+1}K^{-1}\mathbf{e}_1$, and we only need to solve a single linear system with the matrix K. In Section 11.3 we test the resulting algorithm for solving an ill conditioned real symmetric Toeplitz linear system.

8 Continuation methods for structured matrix inversion

In this section we present our determinstic continuation algorithms for structured matrix inversion. Their convergence and complexity depend on the condition number of the input matrix, but not on its numerical nullity r. In the case of a small positive nullity r they run as fast as our randomized algorithms up to factor $\log n$, but do not slow down when the numerical nullity increases or becomes undefined (see Remark 2.1) unless the condition number of the input matrix increases.

In Sections 8.1–8.3 we assume Hermitian positive definite input matrix M with a structure of Toeplitz type. In Section 8.4 we extend this study to Hermitian indefinite and non-Hermitian Toeplitz-like inputs C and to the inputs with the structures of Hankel, Vandermonde and Cauchy types.

8.1 A continuation algorithm

We begin with fixing a positive scalar t_0 and a readily invertible matrix $M_0 = M + t_0 I$ where the ratio $\frac{||M||}{t_0}$ is noticeably less than one. Then we write $M_{-1} = t_0 I$, choose parameters $t_0, t_1, \cdots, t_q$ such that $t_0 > t_1 > \cdots > t_q > t_{q+1} = 0$, and recursively invert the matrices $M_k = M + t_k I$, for $k = 0, 1, \ldots, q+1$, by using the initial approximations $\tilde{M}_{k-1}^{-1} \approx M_{k-1}^{-1}$ and applying iterative refinement or any alternative method (see Remark 8.2 and Section 9). By choosing proper scalars $t_0, t_1, \cdots, t_q$ we yield sufficiently small initial residual norms $||I - \tilde{M}_{k-1}^{-1} M_k||$ and therefore ensure fast convergence to the inverse M_k^{-1} at the kth continuation step for $k = 1, \ldots, q$. This leads us to our next algorithm. We present it for Hermitian positive definite Toeplitz-like input matrices M, for which it is most effective.

Algorithm 8.1. Inversion based on continuation.

INPUT: *two small positive tolerance values δ and τ, an $n \times n$ Hermitian positive definite Toeplitz-like matrix M (in the class $\mathcal{T}_d$ for a small integer d), and a Subroutine INVERT that computes refinements $X_1, X_2, \ldots$ of an approximation X_0 to the inverse of a nonsingular matrix W, stops at the m-th step as soon as $||I - WX_m|| \le \tau$, and then outputs the matrix $\tilde{W}^{-1} = X_m$. (Such a subroutine can rely on iterative refinement or Newton's iteration in Section 9.)*

OUTPUT: *an approximate inverse X such that $||I - MX|| < \delta$.*

INITIALIZATION: *Choose two positive scalars t_0 and u such that $\tau < u < 1$ and $t_0 u \ge ||M||$, compute the matrix $M_0 = M + t_0 I$, and write $k = 0$ and $\tilde{M}_{-1}^{-1} = \frac{1}{t_0} I$.*

COMPUTATIONS:

1. Apply the Subroutine INVERT to the matrix $W = M_k$ by using the initial approximate inverse $X_0 = \tilde{M}_{k-1}^{-1}$. Let $\tilde{M}_k^{-1}$ denote the output matrix. Compute the scalar $t_k - t_{k+1} = \min\{0, \frac{u-\tau}{||\tilde{M}_k^{-1}||}\}$.

2. If $t_k - t_{k+1} > 0$, compute the matrix $M_{k+1} = M_k + (t_{k+1} - t_k)I$, increment k by one, that is set $k \leftarrow k+1$, and reapply Stage 1.

3. If $t_{k+1} - t_k = 0$, apply the Subroutine INVERT to the matrix $W = M$ by using the initial approximate inverse $X_0 = \tilde{M}_{k+1}^{-1}$ and stop where $||I - MX_m|| \le \delta$, that is change the tolerance from τ to δ in the stopping criterion of the subroutine. Output the computed approximation $X_m = \tilde{M}^{-1}$ and stop.

8.2 Analysis of the algorithm and a modification

The two following theorems together imply correctness of the algorithm.

Theorem 8.1. *At the k-th application of the Subroutine INVERT in Algorithm 8.1 the initial residual norm $||I - M_k\tilde{M}_{k-1}^{-1}||$ does not exceed u for all k.*

Proof. We have $\tilde{M}_{-1}^{-1} = \frac{1}{t_0}I$, and the definition of the parameters u and t_0 implies that $||I - M_0\tilde{M}_{-1}^{-1}|| = \frac{1}{t_0}||M|| \le u$. This proves the theorem for $k = 0$.

For a positive k we have $M_k = M_{k-1} + (t_k - t_{k-1})I$, and so $I - M_k\tilde{M}_{k-1}^{-1} = I - M_{k-1}\tilde{M}_{k-1}^{-1} + \tilde{M}_{k-1}^{-1}(t_{k-1} - t_k)$, $||I - M_k\tilde{M}_{k-1}^{-1}|| \le ||I - M_{k-1}\tilde{M}_{k-1}^{-1}|| + ||\tilde{M}_{k-1}^{-1}(t_{k-1} - t_k)|| \le \tau + ||\tilde{M}_{k-1}^{-1}||\,(t_{k-1} - t_k)$.

By the definition of the value $t_{k-1} - t_k$ at Stage 1 of Algorithm 8.1 we have $t_{k-1} - t_k \le \frac{u-\tau}{||\tilde{M}_{k-1}^{-1}||}$. Substitute this inequality into the above bound on the norm $||I - M_k\tilde{M}_{k-1}^{-1}||$ and obtain the theorem for a positive k. □

Theorem 8.2. *For a sufficiently small scalar τ, Algorithm 8.1 arrives at equation $t_{q+1} = 0$ for some integer q such that*

$$q \log \frac{1}{1-u} \le \log \frac{\mathrm{cond}(M)}{u^2}. \tag{8.1}$$

The proof of the theorem employs some auxilairy matrices $P_k = M_k^{-1}M_{k+1}$ and $V_k = M_k^{-1}M$ and uses the step sizes $t_0 - t_1, t_1 - t_2, \ldots$ defined by policy (8.2) below. These steps are independent of the matrices $M_1, M_2, \ldots$ and are not larger than in Algorithm 8.1, but still support bound (8.1).

Theorem 8.3. *For a matrix $M \in \mathbb{C}^{n\times n}$ define positive scalars t_0, u_k such that $u_k < 1$, and*

$$t_{k+1} = t_k(1 - u_k) = t_0 \prod_{j=0}^{k}(1 - u_j) \tag{8.2}$$

and the matrices $M_k = M + t_kI$, $P_k = I - t_ku_kM_k^{-1}$, for $k = 0, 1, \ldots$. Suppose the matrices M_k are nonsingular for all k. Then for $k = 0, 1, \ldots$ we have

(a) $M_{k+1} = M_kP_k = P_kM_k = P_kP_{k-1}\cdots P_0M_0 = M_0P_0P_1\cdots P_k$,

(b) $M = P_{k-1}\cdots P_0M_0V_k$ where $V_k = M_k^{-1}M$, and

(c) $I - V_k = t_kM_k^{-1}$.

Note that $M_k^{-1}P_k^{-1} = P_k^{-1}M_k^{-1}$ because $P_k = I - t_ku_kM_k^{-1}$. Furthermore we have $V_q = I - t_qM_q^{-1}$ and $M^{-1} = V_q^{-1}M_0^{-1}P_0^{-1}\cdots P_q^{-1}$.

Theorem 8.4. *Under the assumptions of Theorem 8.3, let a matrix M have the spectrum $\Lambda(M) = \{\lambda_1, \ldots, \lambda_n\}$. Then we have*

$$\begin{aligned} \Lambda(\tfrac{1}{t_0}M_0 - I) &= \{\tfrac{\lambda_i}{t_0}\}_{i=1}^n, \\ \Lambda(I - P_k) &= \{\tfrac{t_ku_k}{t_k+\lambda_i}\}_{i=1}^n, \quad k = 0, 1, \ldots, q-1, \\ \Lambda(I - V_q) &= \{\tfrac{t_q}{\lambda_i+t_q}\}_{i=1}^n. \end{aligned}$$

Corollary 8.1. *Under the assumptions of Theorem 8.3, suppose M is a Hermitian and positive definite matrix, $t_0 > 0$, and*

$$\lambda^+ \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq \lambda_- > 0.$$

Then

(a) the matrices M_0, $\frac{1}{t_0}M_0 - I$, P_k, $I - P_k$ (for $k = 0, 1, \ldots, q-1$), V_q and $I - V_q$ are Hermitian and positive definite,

(b) $||I - \frac{1}{t_0}M_0|| \leq \frac{\lambda_1}{t_0} \leq \frac{\lambda^+}{t_0}$, $||I - P_k|| = ||I - M_k^{-1}M_{k+1}|| < u_k$, $k = 0, 1, \ldots, q-1$,

$||I - V_q|| \leq \frac{1}{1+\lambda_n/t_q} \leq \frac{1}{1+\lambda_-/t_q}$, $t_q = t_0 \prod_{i=0}^{q-1}(1 - u_i)$,

(c) $\mathrm{cond}(M_0) = 1 + \frac{\lambda_1 - \lambda_n}{t_0 + \lambda_n} < 1 + \frac{\lambda_1}{t_0} \leq 1 + \frac{\lambda^+}{t_0}$,

$\mathrm{cond}(P_k) = \frac{1-(t_k u_k)/(t_k+\lambda_1)}{1-(t_k u_k)/(t_k+\lambda_n)} < \frac{1}{1-u_k}$, $k = 0, \ldots, q-1$,

$\mathrm{cond}(V_q) = \frac{1-t_q/(\lambda_1+t_q)}{1-t_q/(\lambda_n+t_q)} < 1 + \frac{t_q}{\lambda_n} \leq 1 + \frac{t_q}{\lambda_-}$.

Corollary 8.2. *Under the assumptions of Theorem 8.3, write $\kappa^+ = \frac{\lambda^+}{\lambda_-}$, $t_0 = \lambda^+/u$, and $u_k = u$, for a scalar u such that $0 < u < 1$ and for $k = 0, 1, \ldots, q-1$. Then $||I - \frac{1}{t_0}M_0|| \leq u$, $||I - P_k|| = ||I - M_k^{-1}M_{k+1}|| < u$, $k = 0, 1, \ldots, q-1$, $||I - V_q|| = ||I - M_q^{-1}M|| \leq \kappa^+(\frac{1}{u} - 1)^q$, $\mathrm{cond}(M_0) \leq 1 + u$, $\mathrm{cond}(P_k) < \frac{1}{1-u}$ for all k, $\mathrm{cond}(V_q) < 1 + \kappa^+(\frac{1}{u} - 1)^q$. If $q \log \frac{1}{1-u} \geq \log(\kappa^+/u^2)$, then $||I - V_q|| = ||I - M_q^{-1}M|| \leq u$ and $\mathrm{cond}(V_q) < 1 + u$, so that in Corollary 8.1 none of the residual norms exceeds u and none of the condition numbers exceeds $\frac{1}{1-u}$.*

Remark 8.1. *Instead of applying Algorithm 8.1 we can successively compute the matrices $t_0\tilde{M}_0^{-1}$, $P_k = \tilde{M}_k^{-1}M_{k+1}$, $\tilde{P}_k^{-1} \approx P_k^{-1}$, $\tilde{M}_{k+1}^{-1} = \tilde{P}_k^{-1}\tilde{M}_k^{-1}$ for $k = 0, 1, \ldots, q$, $V_q = \tilde{M}_q^{-1}M$, $\tilde{V}_q^{-1} \approx V_q^{-1}$, and $\tilde{M}^{-1} = \tilde{M}_q^{-1}\tilde{V}_q^{-1}$. Hereafter we refer to the resulting variation of Algorithm 8.1 as* **Algorithm 8.2**.

Remark 8.2. *Corollary 8.1 implies that the matrices $\frac{1}{t_0}M_0$ (for $t_0 > \lambda_1$), P_k and V_k for all k are diagonally dominant under the 2-norm distance $||\cdot||$ and moreover share the approximate inverse I with the residual norm of at most $u = \max_k u_k$. Consequently they can be inverted by means of a number of effective algorithms, including iterative refinement, Jacobi's, Gauss–Seidel's, CG and GMRES iterations, and for all of them global linear convergence (right from the start) is ensured.*

Corollary 8.2 includes the bounds $||I - M_k^{-1}M_{k+1}|| < u$, $k = 0, 1, \ldots, q$, which show that the step sizes $t_{k-1} - t_k$ defined by equation (8.2) for $u_k = u$ and $k = 0, 1, \ldots$ are not larger than in Algorithm 8.1 (for a sufficiently small tolerance τ) and that $t_{q+1} = 0$ for q satisfying bound (8.1). This implies Theorem 8.2 because we have $\kappa^+ = \mathrm{cond}(M)$ if we set $\lambda^+ = \lambda_1$ and $\lambda_- = \lambda_n$.

Remark 8.3. *One can slightly modify Algorithm 8.1 by defining step sizes $t_{k-1} - t_k$ by equation (8.2). This policy could a little increase the overall number of continuation steps but would still keep it within bound (8.1).*

Example 8.1. *Approximate the norm M by applying the known effective bounds or algorithms (see, e.g., [D83], [GL96, Section 2.2.2, 2.2.3], [S98, Section 5.3.3]) and choose $t_0 \geq 2||M||$ and $u_k = u \leq 1/2$ (resp. $t_0 \geq 4||M||$ and $u_k = u \leq 1/4$) for all k. Then all the residual norms in Corollary 8.2 are at most $1/2$ (resp. $1/4$), and under the choice of the initial approximate inverse $X_0 = I$ iterative refinement linearly converges to the matrices $t_0M_0^{-1}$, $M_1^{-1}, \ldots, M_q^{-1}$, and M^{-1} provided*

$$q \log \frac{1}{1-u} \geq \log \frac{\kappa^+}{u^2}. \tag{8.3}$$

Furthermore Corollary 8.2 implies that $\mathrm{cond}(M_0) < 1.5$, $\mathrm{cond}(P_k) < 2$ for all k, and $\mathrm{cond}(V_q) < 1.5$ (resp. $\mathrm{cond}(M_0) < 1.25$, $\mathrm{cond}(P_k) < \frac{4}{3}$ for all k, and $\mathrm{cond}(V_q) < 1.25$).

To calculate a desired number of steps q satisfying bound (8.3) we need an upper bound κ^+ on the condition number $\operatorname{cond} M$. We can apply the known condition estimators, but we can adopt a less costly alternative policy of generating and inverting the matrices M_k in Algorithm 8.1 or the matrices M_k and P_k in Algorithm 8.2 for $k = 0, 1, \dots$ until we yield $t_{k+1} = t_k$ in Algorithm 8.1 or until we observe that the norm $||I - \tilde{M}_k^{-1} M||$ is small enough in Algorithm 8.2. In the latter case we would set $q = k$ and compute the inverse M^{-1} by using the initial approximation $\tilde{M}_q^{-1}$.

Remark 8.4. *Suppose we need to invert a nonsingular ill conditioned Hermitian positive definite Toeplitz matrix M. Then our continuation process can employ the Gohberg–Semencul's formulae, thus fully exploiting this structure, whereas additive preprocessing would generally require to invert a non-Toeplitz matrix with displacement rank four, and the augmentation $M \Longrightarrow K = \begin{pmatrix} M & U^H \\ U & W \end{pmatrix}$ would have no preconditioning power if the augmented matrix K is Hermitian and positive definite (see Remark 5.2).*

Remark 8.5. *Our policy in Example 8.1 is actually quite effective. Indeed observe the following lower bound on the number l of continuation steps, $q+1 > \log_\kappa \operatorname{cond}(M)$ for every scalar κ exceeding the condition numbers of the matrices M_0, P_0, …, P_{q-1}, and V_q. This bound is implied by the inequality $\operatorname{cond}(M) \le \operatorname{cond}(M_0) \operatorname{cond}(V_q) \prod_{k=0}^{q-1} \operatorname{cond}(P_k)$.*

8.3 Complexity estimates

Choose the step sizes as in Example 8.1, so that $q = O(\log \operatorname{cond} M)$ continuation steps are sufficient. Choose $\tau = 0.5u$ in the Subroutine INVERT. Then the estimates for the residual norm $||I - M_k \tilde{M}_k^{-1}||$ in the proof of Theorem 8.1 imply that at the kth continuation step we just need to decrease the initial value of this norm by twice, and for this task a fixed constant number of the iterative refinement (or CG) loops would suffice. In fact we need $O(\log(dn))$ loops because the transition from an $n \times n$ Toeplitz-like matrix to the displacement of its inverse can increase the residual norm by the factor dn [P92], [P93], [P93a], [P01, Sections 6.4–6.6]. In Section 9 we propose a heuristic approach to avoiding this transition in the case of a Toeplitz input matrix. If succeeds, this approach would imply acceleration by the factor $\log(dn)$).

At the final continuation step we set $\tau = t = 2^{-p_t}$ where p_t is the required output precision.

Summarizing we need $O((\log n) \log \operatorname{cond} M + p_t)$ loops of iterative refinement overall, that is $O(((\log(dn)) \log \operatorname{cond} M) + p_t) dn \log n)$ flops in the case of Toeplitz-like or Hankel-like input matrices having a displacement rank d. This bound is deterministic, involves no assumption about numerical nullity of the input matrix, and is still very close to the information lower bound in Section 6.6.

8.4 Extensions to Hermitian indefinite and non-Hermitian matrices and to matrices with the structures of Hankel, Vandermonde and Cauchy types

By choosing a random trajectory $\{t_0, t_1, \dots, t_q, t_{q+1} = 0\}$ on the complex plain we can ensure with a high probability that all auxiliary matrices that we invert have condition numbers not much larger than $\operatorname{cond} M$ even where we relax the assumption that the input matrix is Hermitian and positive definite. Then our analysis and cost estimates can be extended (under a probabilistic model).

To extend our study to the cases of Hermitian indefinite matrices M by using no randomization, it is sufficient to modify the matrices M_k and P_k in Algorithms 8.1 and 8.2 by replacing $t_k \leftarrow t_k \sqrt{-1}$ for all k. To extend our study to non-Hermitian matrices C we can apply the standard symmetrizations $N = \begin{pmatrix} O & C^H \\ C & O \end{pmatrix}$, $M = C^H C$ or $M = C C^H$ based on the matrix equations $N^{-1} = \begin{pmatrix} O & C^{-1} \\ C^{-H} & O \end{pmatrix}$, $C^{-1} = (C^H C)^{-1} C^H = C^H (C C^H)^{-1}$. We refer the reader to [PKRK06, Section 7] on some other extensions of the continuation techniques to the case of non-Hermitian input matrices.

If the input matrix M has structure of Toeplitz/Hankel type or the rank (semiseparable, quasiseparable) structure [EG99], [VVM07], [VVM08], then so do the matrices M_k, P_k, and V_k for all k as well, and we can accelerate the computations respectively at every continuation step. If the

matrix M has a non-Toeplitz displacement structure, we can extend it to the matrices M_k for all k (and consequently also to the matrices P_k and V_k for all k) simply by redefining the matrices: $M_k \leftarrow M + t_k N$ where the matrix N shares the structure with the matrix M. E.g., for a Hankel-like matrix M, we can choose N equal to the matrix $\gamma||M||J$ for a fixed $\gamma > 1$ (e.g., for $\gamma \approx 2$) and the reflection matrix J, defined in Section 7. For matrices M having the structure of Vandermonde or Cauchy type, we can choose N being a fixed scaled Vandermonde or Cauchy matrix, respectively, associated with the same displacement operator as the input matrix M.

There is also an alternative. Given a nonsingular Hankel-like matrix M, we can recall that MJ and JM are Toeplitz-like matrices, invert one of them, and then obtain the inverse $M^{-1} = J(MJ)^{-1} = (JM)^{-1}J$. Likewise, to invert a nonsingular matrix M having the structures of Vandermonde or Cauchy types, we can first compute the matrix $N = VMW$ where each of V and W can be either an appropriate Vandermonde matrix, the inverse or transpose of such a matrix, or the identity matrix. Then the matrix equation $M^{-1} = W^{-1}N^{-1}V^{-1}$ would reduce the original inversion problem to the case of a Toeplitz-like matrix N. To this matrix we would apply our continuation process based on Theorem 8.3. Such a technique of *displacement transformation* is due to [P90] (cf. [P01, Sections 1.7, 4.7–4.9], was extensively studied by G. Heinig since 1995, and is most widely known because of its effective application to practical solution of Toeplitz and Toeplitz-like linear systems of equations in [GKO95] and the subsequent papers [CGLX], [CGSXZ], [G98], [P10], [R06].

9 Newton's structured iteration and preconditioning

Recall Newton's iteration for matrix inversion

$$X_{i+1} = X_i(2I - CX_i),\ i = 0, 1, \dots. \tag{9.1}$$

Its i-th loop squares the residual $I - CX_i$, that is, we have

$$I - CX_{i+1} = (I - CX_i)^2 = (I - CX_0)^{2^{i+1}}. \tag{9.2}$$

Therefore

$$||I - CX_{i+1}|| \le ||I - CX_i||^2 = ||I - CX_0||^{2^{i+1}},\ i = 0, 1, \dots, \tag{9.3}$$

so that the approximations X_i quadratically converge to the inverse C^{-1} right from the start provided that $||I - CX_0|| < 1$.

We can ensure that $||I - CX_0|| \le 1 - \frac{2n}{(\mathrm{cond}(C))^2(1+n)}$ by choosing $X_0 = \frac{2nC^H}{(1+n)||C||_1||C||_\infty}$ [PS91].

Such a map $C \Longrightarrow X_0$ preserves the matrix structure of Toeplitz or Hankel type, but is the structure maintained throughout the iteration? Not automatically. In fact a Newton's loop can triple the displacement rank of the matrix X_k. The structure can be maintained, however, via recursive compression of the displacement (also called recompression), in which case we arrive at *Newton's structured iteration*. In particular we can periodically set to zero the smallest singular values of the displacements of the matrices X_i wherever the length of the displacements exceeds the tolerance t equal to or a little exceeding the displacement rank of the input matrix C.

We refer the reader to [P01, Chapter 6] on the history, variations, and analysis of this approach, first proposed and analyzed in [P92], [P93], and [P93a]. In particular according to the estimates in [P01], the structured iteration converges quadratically right from the start provided $||I - CX_0|| < \frac{1}{(1+||Z_e||+||Z_f||)\,\mathrm{cond}(C)}||L^{-1}||$, $||L^{-1}|| \le c_{e,f}n$, L denotes the operator $\nabla_{Z_e,Z_f}(C)$ for $e \ne f$ or $\Delta_{Z_e,Z_f^T}(C)$ for $ef \ne 1$, and $c_{e,f}$ is a constant defined by e and f.

It can be beneficial to combine Newton's iteration with our preprocessing. We pointed out one direction of linking the two techniques in Remark 6.1. One can also try to apply preconditioning to avoid or to accelerate the stage of slow start of Newton's iteration, observed where the initial residual norm $||I - CX_0||$ is close to one. In this case preconditioning is a natural way to decreasing this norm, that is, with our preconditioning we can ensure that $||I - CX_0|| \le u$ for a constant $u < 1$, then apply $O(\log n)$ loops of iterative refinement to satisfy the above initialization bound, and finally shift to Newton's structured iteration. In this case we must perform extra refinement steps (to yield

the output with high accuracy), but these steps are noncostly in the case of structured Newton's iteration.

At the initial application of iterative refinement (the stage of slow start), we put up with linear convergence, but the experiments reported in [P01, Table 6.21] suggest that we can avoid this stage at least in the case of Toeplitz matrices C. Namely these experiments show global convergence of Newton's structured iteration with compression (right from the start) in about 25% of tests, including the cases where the initial residual norm $||I - CX_0||$ was very close to one.

This motivates concurrent applications of a number of variations of Newton's structured iteration (including variations of its compression policy (cf. [PS91], [P01, Chapter 6], and [PVW04])) to a number of scaled randomized small rank modifications and small size augmentations of the input matrix. As soon as one of these applications produces the inverse, we can recover the inverse of the original matrix via the SMW formula (also see Theorem 3.2 for augmetations). Likewise we can apply the iteration to $(n-r) \times (n-r)$ block submatrices of the matrix C for small integers r and to the matrices M_k in Algorithm 8.1 under various policies of choosing the step sizes t_k.

Of course it is interersting whether this approach can also work for other classes of structured matrices.

Remark 9.1. *Fast convergence of Newton's iteration implies rapid increase of the number of correct bits computed per an output value. If at some stage this number exceeds the selected precision p, one should either extend this precision or compute the output values as the sums represented implicitly by p-precision summands.*

10 Solution of a linear system of equations via nmb computation

Theorem 3.3 reduces the computation of null vectors and nmbs to the solution of some nonsingular linear systems of equations. Let us point out three converse reductions.

1. The solution of a nonsingular linear system of n equations $A\mathbf{y} = \mathbf{b}$ can be expressed via the null vector $\begin{pmatrix} \mathbf{y} \\ -1/\beta \end{pmatrix}$ of the matrix $K = (A, \beta\mathbf{b})$ for a nonzero scalar β. If the matrix A has numerical nullity one and if the ratio $||A||/||\beta\mathbf{b}||$ is neither large nor small, then on the average vector $\mathbf{b}$ the map $A \Longrightarrow K$ serves as preconditioning [PQa].

2. The solution of a linear system $A\mathbf{y} = \mathbf{b}$ can be reduced to the nmb computation based on part (a) of Theorem 3.4.

Assume that the $n \times n$ nonsingular input matrix A has a small positive numerical nullity r (that is the ratio $\frac{\sigma_1(A)}{\sigma_{n-r}(A)}$ is not large, whereas $\sigma_1(A) \gg \sigma_{n-r+1}(A)$) and devise the following randomized algorithm for its preconditioning.

Algorithm 10.1. Preconditioning by using nmbs.

INPUT: *Two integers n and r, $0 < r < n$, a nonsingular matrix $A \in \mathbb{C}^{n\times n}$ having numerical rank $n-r$ and scaled so that $||A|| = 1$, and a Subroutine* LIN·SOLVE *that solves a linear system of equations if it is nonsingular and well condtioned or outputs FAILURE otherwise.*

OUTPUT: *FAILURE or four matrices K_0, $L_0 \in \mathbb{C}^{n\times(n-r)}$ and K_1, $L_1 \in \mathbb{C}^{n\times r}$ such that $W = (K_0, K_1)^H A(L_0, L_1) = \begin{pmatrix} W_{00} & W_{01} \\ W_{10} & W_{11} \end{pmatrix}$ and with a probability near one the block submatrix $W_{00} = K_0^H A L_0$ is nonsingular, well conditioned, and strongly dominant, that is $||W_{00}|| \gg \max\{||W_{01}||, ||W_{10}||, ||W_{11}||\}$.*

INITIALIZATION: *Generate four standard Gaussian random matrices S, $T \in \mathbb{C}^{n\times(n-r)}$, U, $V \in \mathbb{C}^{n\times r}$.*

COMPUTATIONS:

1. *Compute the matrix* $C = A + UV^H$ *(expected to be nonsingular and well condtioned according to the study in Section 5).*
2. *Apply the Subroutine* LIN·SOLVE *to compute and to output the matrices* $K_1 = C^{-H}V$ *and* $L_1 = C^{-1}U$. *Stop and output FAILURE if so does the subroutine.*
3. *Compute and output the matrices* $K_0 = A^H S$ *and* $L_0 = AT$ *and stop.*

Correctness of the algorithm follows because the value $\sigma_{n-r}(W_{00})$ is likely to have the order of $\sigma_{n-r}(A)$ in virtue of Theorem 4.7, whereas the matrices W_{01}, W_{10}, and W_{11} have the norms of at most the order of $\sigma_{n-r+1}(A)$ because in virtue of part (a) of Theorem 3.4 the matrices K_1 and L_1 are approximate nmbs of the matrices A^H and A, respectively, and because the matrix A has numerical nullity r (which implies that $\sigma_{n-r}(A) \gg \sigma_{n-r+1}(A)$, whereas the ratio $\sigma_1(A)/\sigma_{n-r}(A)$ is not large).

Having the dominant, nonsingular and well conditioned block W_{00} in the 2×2 block matrix W, we can apply block Gaussian elimination and readily factorize the matrix W as follows,

$$W = \begin{pmatrix} I & O \\ W_{10}W_{00}^{-1} & I \end{pmatrix} \begin{pmatrix} W_{00} & W_{01} \\ O & G \end{pmatrix}$$

where $G = W_{11} - W_{10}W_{00}^{-1}W_{01}$ is called the Gauss tranform and Schur complement [GL96].

Based on this factorization, one can immediately reduce the inversion of the matrices W and A and the solution of a linear system $A\mathbf{y} = \mathbf{0}$ to the similar operations with the matrices W_{00} and G of smaller sizes, expected to be nonsingular and better conditioned. This dramatically improves the quality of the solution (see Tables 12.7 and 12.8).

Remark 10.1. *The* $O(n^2 r)$ *flops involved in the computation of the* $(2n - r)r$ *entries of the blocks* W_{01}, W_{10}, *and* W_{11} *(which are the* r/n *fraction of all flops used) must be performed in extended precision to counter the expected cancellation of the leading digits of the input values.*

Remark 10.2. *To work with fewer random parameters one can generate a single* $n \times q$ *standard Gaussian random matrix for* $q \geq \max\{r, n - r\}$ *and then reuse its columns while defining the four auxiliary random matrices* U, V, K_0, *and* L_0.

3. The solution of a linear system $A\mathbf{y} = \mathbf{b}$ can be reduced to the nmb computation based on part (b) of Theorem 3.4, which employs dual additive preprocessing and leads to a dual version of Algorithm 10.1. In this version a crude approximation to the norm A^{-1} of an ill conditioned input matrix A is required (see [D83] on fast randomized computation of such an approximation), but no matrix inversion is involved, except for the inversion of an auxiliary $q \times q$ matrix H, expected to be close to the identity matrix I_q.

Algorithm 10.2. Dual preconditioning by using nmbs.

INPUT: *Two integers* n *and* q, $0 < q < n$, *a nonsingular matrix* $A \in \mathbb{C}^{n \times n}$ *having numerical rank* q *and scaled so that* $||A^{-1}|| = 1$ *(and consequently the norm* $||A||$ *is small since the matrix* A *is ill conditioned under the above assumption), and a Subroutine* LIN·SOLVE *that solves a linear system of equations if it is nonsingular and well condtioned or outputs FAILURE otherwise.*

OUTPUT: *FAILURE or four matrices* K_0, $L_0 \in \mathbb{C}^{n \times q}$ *and* K_1, $L_1 \in \mathbb{C}^{n \times (n-q)}$ *such that* $W = (K_0, K_1)^H A (L_0, L_1) = \begin{pmatrix} W_{00} & W_{01} \\ W_{10} & W_{11} \end{pmatrix}$ *and with a high probability the block submatrix* $W_{00} = K_0^H A L_0$ *is nonsingular, well conditioned, and strongly dominant, that is*

$$||W_{00}|| \gg \max\{||W_{01}||, \ ||W_{10}||, \ ||W_{11}||\}.$$

INITIALIZATION: *Generate four standard Gaussian random matrices* $S,\ T \in \mathbb{C}^{n\times(n-q)}$, $U_-,\ V_- \in \mathbb{C}^{n\times q}$.

COMPUTATIONS:

1. *Compute the matrix* $H = I_q + V_- A U_-^H$ *(expected to be close to the identity since the norm* $||A||$ *is small).*
2. *Apply the Subroutine* LIN·SOLVE *to compute the matrix* H^{-1}. *Stop and output FAILURE if so does the subroutine.*
3. *Compute the matrix* $C_- = A - AU_- H^{-1} V_-^H A$.
4. *Compute and output the matrices* $K_0 = C_-^H V_-$ *and* $L_0 = C_- U_-$.
5. *Compute and output the matrices* $K_1 = A^H S$ *and* $L_1 = AT$ *and stop.*

Both Algorithms 10.1 and 10.2 can be extended to the block factorization of rectangular inputs A.

Remarks 10.1 and 10.2 can be readily extended. We only specify an extension of Remark 10.1.

Remark 10.3. *The* $O((n-q)n^2)$ *flops involved in the computation of the* $(2n-q)q$ *entries of the blocks* W_{01}, W_{10}, *and* W_{11} *(which are the* $(n-q)/n$ *fraction of all flops used) must be performed in extended precision to counter the expected cancellation of the leading digits of the input values.*

11 Numerical Experiments

Our numerical experiments with random general, Hankel, Toeplitz and circulant matrices have been performed in the Graduate Center of the City University of New York on a Dell server with a dual core 1.86 GHz Xeon processor and 2G memory running Windows Server 2003 R2. The test Fortran code was compiled with the GNU gfortran compiler within the Cygwin environment. Random numbers were generated with the random_number intrinsic Fortran function, assuming the uniform probability distribution over the range $\{x : -1 \le x < 1\}$.

11.1 Conditioning tests

Table 12.1 displays the average computed values of the 1- and 2-norms of the matrices A and their inverses as well as the ratios of the 1-norm and the 2-norm. We performed our tests with $n \times n$ general, Toeplitz, and circulant matrices A for $n = 32, 64, \dots, 1024$. We sampled the matrix entries at random in the range of $-1 \le x < 1$ and performed $m = 100$ conditioning tests for each matrix class and each size.

Besides we computed the condition numbers of $n \times n$ random general matrices for $n = 2^k$, $k = 5, 6, \dots$, with the entries sampled in the range $[-1, 1)$ as well as complex general, Toeplitz, and circulant matrices whose entries had real and imaginary parts sampled at random in the same range $[-1, 1)$. We performed $m = 100$ tests for each dimension n and represented the test results in Tables 12.1–12.4. The last four columns of each table display the average (mean), minimum, maximum, and standard deviation of the computed condition numbers of the input matrices, respectively. Specifically we computed the values $\mathrm{cond}(A) = ||A||\ ||A^{-1}||$ for general and circulant matrices A and the values $\mathrm{cond}_1(A) = ||A||_1\ ||A^{-1}||_1$ for Toeplitz matrices A.

11.2 Preconditioning tests

Table 12.5 reproduces some results of testing preconditioning power of additive preprocessing in [PIMR10]. The tests covered the input matrices constructed as follows.

1n. *Nonsymmetric matrices of type I with numerical nullity* ν. $A = S\Sigma_\nu T^H$ are $n \times n$ matrices where G and H are $n \times n$ random orthogonal matrices, that is, the Q-factors in the QR factorizations of random real matrices; $\Sigma_\nu = \mathrm{diag}(\sigma_j)_{j=1}^n$ is the diagonal matrix such that $\sigma_{j+1} \le \sigma_j$ for $j = 1, \dots, n-1$, $\sigma_1 = 1$, the values $\sigma_2, \dots, \sigma_{n-\nu-1}$ are randomly sampled in the semi-open interval

$[0.1, 1)$, $\sigma_{n-\nu} = 0.1$, $\sigma_j = 10^{-16}$ for $j = n-\nu+1, \dots, n$, and therefore $\operatorname{cond} A = 10^{16}$ (cf. [H02, Section 28.3]).

1s. *Symmetric matrices of type I with numerical nullity ν.* The same as in part 1n, but for $G = H$.

The matrices of six other classes were constructed in the form of $\frac{A}{||A||} + \beta I$ where the recipes for defining the matrices A and scalars β are specified below.

2n. *Nonsymmetric matrices of type II with numerical nullity ν.* $A = (W, WZ)$ where W and Z are random orthogonal matrices of sizes $n \times (n-\nu)$ and $(n-\nu) \times \nu$, respectively.

2s. *Symmetric matrices of type II with numerical nullity ν.* $A = WW^H$ where W are random orthogonal matrices of size $n \times (n-\nu)$.

3n. *Nonsymmetric Toeplitz-like matrices with numerical nullity ν.* $A = c(T, TS)$ for random Toeplitz matrices T of size $n \times (n-\nu)$ and S of size $(n-\nu) \times \nu$ and for a positive scalar c such that $||A|| \approx 1$.

3s. *Symmetric Toeplitz-like matrices with numerical nullity ν.* $A = cTT^H$ for random Toeplitz matrices T of size $n \times (n-\nu)$ and a positive scalar c such that $||A|| \approx 1$.

4n. *Nonsymmetric Toeplitz matrices with numerical nullity one.* $A = (a_{i,j})_{i,j=1}^{n}$ is an $n \times n$ Toeplitz matrix. Its entries $a_{i,j} = a_{i-j}$ are random for $i - j < n - 1$. The entry $a_{n,1}$ is selected to ensure that the last row is linearly expressed through the other rows.

4s. *Symmetric Toeplitz matrices with numerical nullity one.* $A = (a_{i,j})_{i,j=1}^{n}$ is an $n \times n$ Toeplitz matrix. Its entries $a_{i,j} = a_{i-j}$ are random for $|i - j| < n - 1$, whereas the entry $a_{1,n} = a_{n,1}$ is a root of the quadratic equation $\det A = 0$. We have repeatedly generated the matrices A until we arrived at the quadratic equation having real roots.

The scalar β was set equal to 10^{-16} for the symmetric matrices A, in the classes 2s, 3n, and 4s, so that $\operatorname{cond}(A) = 10^{16}+1$ in these cases. For the nonsymmetric matrices A the scalar β was defined by an iterative process such that $||A|| \approx 1$ and $10^{-18}||A|| \le \operatorname{cond}(A) \le 10^{-16}||A||$ (cf. [PIMR10, Section 8.2]).

The table displays the average values of the condition numbers $\operatorname{cond}(C)$ for the matrices $C = A + UU^T$ over 100,000 tests for the inputs in the above classes, $\nu = r$ in the range $\{1, 2, 4, 8\}$ and $n = 100$. The additive preprocessor UU^T was defined by a normalized $n \times r$ matrix $U = U/||U||$ where $U^T = (\pm I, O_{r,r}, \pm I, O_{r,r}, \dots, O_{r,r}, \pm I, O_{r,s})$, the integer s was chosen to obtain $n \times r$ matrices U, and the signs for the matrices $\pm I$ were chosen at random.

In our further tests the condition numbers of the matrices $C = A + 10^p UV^T$ for $p = -10, -5, 5, 10$ were steadily growing within the factor $10^{|p|}$ as the value $|p|$ was growing. This showed the importance of proper scaling of the additive preprocessor UV^T.

11.3 Solution of a real symmetric Toeplitz linear system of equations with randomized augmentation

We solved 100 real symmetric linear systems of equations $T\mathbf{y} = \mathbf{b}$ for each input class where we used vectors $\mathbf{b}$ with random coordinates from the range $[-1, 1)$ and Toeplitz matrices $T = S + 10^{-9} I_n$ for an $n \times n$ singular symmetric Toeplitz matrices S with nullity one, generated according to the recipe in [PQ10, Section 10.1b].

Table 12.6 shows the average CPU time of the solution by our Algorithm 7.1 and (for comparison) based on the QR factorization and SVD, which we computed by applying the LAPACK procedures DGEQRF and DGESVD, respectively.

The abbreviations "Alg. 7.1", "QR", and "SVD" point out to the respective algorithms. The last two columns of the table display the ratios of these data on the CPU time.

We measured the CPU time with the mclock function by counting cycles. One can convert them into seconds by dividing their number by a constant CLOCKS_PER_SEC, which is 1000 on our platform. The table entries are marked by a "-" where the tests required too long runtime and were not completed.

We obtained the solutions $\mathbf{y}$ with the relative residual norms of about 10^{-15} in all three algorithms, which showed that Algorithm 7.1 (employing iterative refinement) was as reliable as the QR and SVD based solutions (but ran much faster).

11.4 Solution of general linear systems of equations

We chose $n = 32, 64$ and $r = 1, 2, 4$ and for every pair (n, r) generated $m = 100$ instances of vectors $\mathbf{b}$ and matrices A, U, and V as follows.

We generated (a) random vectors $\mathbf{b}$ of dimension n, (b) the matrices A as the error-free products $S\Sigma T^H$ where S and T were $n \times n$ random real orthonormal matrices (generated with double precision), $\Sigma = \text{diag}(\sigma_j)_{j=1}^n$, $\sigma_{n-j} = 10^{j-17}$ for $j = 1, \dots, r$, and $\sigma_{n-j} = 1/(n-j)$ for $j = r+1, \dots, n-1$ (cf. [H02, Section 28.3]), and (c) $n \times r$ random matrices U and V such that $||U| = ||A||$ and $||V|| = 1$.

For every choice of these matrices we solved the linear systems $A\mathbf{y} = \mathbf{b}$ based on Algorithm 10.1. We first generated $n \times (n-r)$ random matrices K_0 and L_0 and then computed the matrices $C = A + UV^T$ (which always was nonsingular and well conditioned in our tests), $K_1 = C^{-T}V$, $L_1 = C^{-1}U$, and $W = (K_0, K_1)^T A(L_0, L_1) = \begin{pmatrix} W_{00} & W_{01} \\ W_{10} & W_{11} \end{pmatrix}$. In all our tests the $(n-r) \times (n-r)$ leading principal $(n-r) \times (n-r)$ block $W_{00} = K_0^T A L_0$ was strongly well conditioned and strongly dominated the three other blocks W_{01}, W_{10}, and W_{11} in the 2×2 block matrix W, as we expected based on our analysis in Section 10. (We computed the dominated blocks W_{01}, W_{10}, and W_{11} with extended precision.) Then we solved the linear system $W\mathbf{x} = (K_0, K_1)^T \mathbf{b}$. We first applied Gaussian elimination with no pivoting to eliminate the subdiagonal block. Then we readily computed the solution of the resulting block triangular linear system, whose both diagonal blocks were expected (and indeed turned out) to be much better conditioned than the original matrix A. Finally we computed and output the vector $\mathbf{y} = (L_0, L_1)\mathbf{x}$.

Table 12.7 shows the average (mean), minimum and maximum values of the relative residual norms $||A\mathbf{y} - \mathbf{b}||/||\mathbf{b}||$ of the output vectors $\mathbf{y}$ as well as the standard deviations observed in these tests.

For comparison we solved the same linear systems by applying the Subroutine MLDIVIDE(A,B) for Gaussian elimination from MATLAB. Table 12.8 shows the respective data on the relative residual norms in these computations.

12 Discussion

Our algorithms for linear system solving and matrix inversion can be immediately extended to numerical computation of determinants based on equations in (3.7) and (3.9) (see [BEPP], [PY99/01], [EP03/05], and the bibliography therein on important applications to geometric and algebraic computations). On various other applications of our randomized preprocessing to fundamental matrix and polynomial computations see [PGMQ], [PQ10], [PQa], [PQZa], [PQZC].

References

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

[AHU74] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Algorithms.* Addison-Wesley, Reading, MA, 1974.

[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.

[BEPP] H. Brönnimann, I. Z. Emiris, V. Y. Pan, S. Pion, Sign Determination in Residue Number Systems, *Theoretical Computer Science*, **210**, **1**, 173–197, 1999. Proceedings version in *Proceedings of 13th Annual ACM Symposium on Computational Geometry*, 174–182, ACM Press, New York, 1997.

Table 12.1: Norms of random general, Toeplitz and circulant matrices and of their inverses

matrix A	n	$\|\|A\|\|_1$	$\|\|A\|\|_2$	$\frac{\|\|A\|\|_1}{\|\|A\|\|_2}$	$\|\|A^{-1}\|\|_1$	$\|\|A^{-1}\|\|_2$	$\frac{\|\|A^{-1}\|\|_1}{\|\|A^{-1}\|\|_2}$
General	32	1.9×10^1	1.8×10^1	1.0×10^0	4.0×10^2	2.1×10^2	1.9×10^0
General	64	3.7×10^1	3.7×10^1	1.0×10^0	1.2×10^2	6.2×10^1	2.0×10^0
General	128	7.2×10^1	7.4×10^1	9.8×10^{-1}	3.7×10^2	1.8×10^2	2.1×10^0
General	256	1.4×10^2	1.5×10^2	9.5×10^{-1}	5.4×10^2	2.5×10^2	2.2×10^0
General	512	2.8×10^2	3.0×10^2	9.3×10^{-1}	1.0×10^3	4.1×10^2	2.5×10^0
General	1024	5.4×10^2	5.9×10^2	9.2×10^{-1}	1.1×10^3	4.0×10^2	2.7×10^0
Toeplitz	32	1.8×10^1	1.9×10^1	9.5×10^{-1}	2.2×10^1	1.3×10^1	1.7×10^0
Toeplitz	64	3.4×10^1	3.7×10^1	9.3×10^{-1}	4.6×10^1	2.4×10^1	2.0×10^0
Toeplitz	128	6.8×10^1	7.4×10^1	9.1×10^{-1}	1.0×10^2	4.6×10^1	2.2×10^0
Toeplitz	256	1.3×10^2	1.5×10^2	9.0×10^{-1}	5.7×10^2	2.5×10^2	2.3×10^0
Toeplitz	512	2.6×10^2	3.0×10^2	8.9×10^{-1}	6.9×10^2	2.6×10^2	2.6×10^0
Toeplitz	1024	5.2×10^2	5.9×10^2	8.8×10^{-1}	3.4×10^2	1.4×10^2	2.4×10^0
Circulant	32	1.6×10^1	1.8×10^1	8.7×10^{-1}	9.3×10^0	1.0×10^1	9.2×10^{-1}
Circulant	64	3.2×10^1	3.7×10^1	8.7×10^{-1}	5.8×10^0	6.8×10^0	8.6×10^{-1}
Circulant	128	6.4×10^1	7.4×10^1	8.6×10^{-1}	4.9×10^0	5.7×10^0	8.5×10^{-1}
Circulant	256	1.3×10^2	1.5×10^2	8.7×10^{-1}	4.7×10^0	5.6×10^0	8.4×10^{-1}
Circulant	512	2.6×10^2	3.0×10^2	8.7×10^{-1}	4.5×10^0	5.4×10^0	8.3×10^{-1}
Circulant	1024	5.1×10^2	5.9×10^2	8.7×10^{-1}	5.5×10^0	6.6×10^0	8.3×10^{-1}

[BGY80] R. P. Brent, F. G. Gustavson, D. Y. Y. Yun, Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximations, *J. Algorithms*, **1**, 259–295, 1980.

[BM01] D. A. Bini, B. Meini, Approximate Displacement Rank and Applications, in *AMS Conference "Structured Matrices in Operator Theory, Control, Signal and Image Processing"*, Boulder, 1999 (edited by V. Olshevsky), *American Math. Society*, 215–232, Providence, RI, 2001.

[BMP00] D. Bondyfalat, B. Mourrain, V. Y. Pan, Computation of a Specified Root of a Polynomial System of Equations Using Eigenvectors, *Linear Algebra and Its Applications*, **319**, 193–209 (2000). (Proceedings version in *ACM-SIGSAM ISSAC'98*.)

[BP94] D. Bini, V. Y. Pan, *Polynomial and Matrix Computations,* volume 1: Fundamental Algorithms, Birkhäuser, Boston, 1994.

[CGLX] S. Chandrasekaran, M. Gu, X. S. Li, J. Xia, Superfast multifrontal method for large structured linear systems of equations, *SIAM. J. on Matrix Analysis and Applications*, **31**, 1382–1411, 2009.

[CGSXZ] 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.

[CW90] Coppersmith, S. Winograd, Matrix Multiplicaton via Arithmetic Progressions. *J. Symbolic Comput.*, **9(3)**, 251–280, 1990.

[D83] J. D. Dixon, Estimating Extremal Eigenvalues and Condition Numbers of Matrices, *SIAM J. on Numerical Analysis*, **20, 4**, 812–814, 1983.

[D88] J. Demmel, The Probability That a Numerical Analysis Problem Is Difficult, *Math. of Computation*, **50**, 449–480, 1988.

Table 12.2: condition numbers cond(A) of random matrices A

n	**input**	**min**	**max**	**mean**	**std**
32	real	2.4×10^1	1.8×10^3	2.4×10^2	3.3×10^2
32	complex	2.7×10^1	8.7×10^2	1.1×10^2	1.1×10^2
64	real	4.6×10^1	1.1×10^4	5.0×10^2	1.1×10^3
64	complex	5.2×10^1	4.2×10^3	2.7×10^2	4.6×10^2
128	real	1.0×10^2	2.7×10^4	1.1×10^3	3.0×10^3
128	complex	1.3×10^2	2.5×10^3	3.9×10^2	3.3×10^2
256	real	2.4×10^2	8.4×10^4	3.7×10^3	9.7×10^3
256	complex	2.5×10^2	1.4×10^4	1.0×10^3	1.5×10^3
512	real	3.9×10^2	7.4×10^5	1.8×10^4	8.5×10^4
512	complex	5.7×10^2	3.2×10^4	2.3×10^3	3.5×10^3
1024	real	8.8×10^2	2.3×10^5	8.8×10^3	2.4×10^4
1024	complex	7.2×10^2	1.3×10^5	5.4×10^3	1.4×10^4
2048	real	2.1×10^3	2.0×10^5	1.8×10^4	3.2×10^4
2048	complex	2.3×10^3	5.7×10^4	6.7×10^3	7.2×10^3

Table 12.3: condition numbers $\text{cond}_1(A)$ of random Toeplitz matrices A

n	**min**	**mean**	**max**	**std**
256	9.1×10^2	9.2×10^3	1.3×10^5	1.8×10^4
512	2.3×10^3	3.0×10^4	2.4×10^5	4.9×10^4
1024	5.6×10^3	7.0×10^4	1.8×10^6	2.0×10^5
2048	1.7×10^4	1.8×10^5	4.2×10^6	5.4×10^5
4096	4.3×10^4	2.7×10^5	1.9×10^6	3.4×10^5
8192	8.8×10^4	1.2×10^6	1.3×10^7	2.2×10^6

[DH03] J. Demmel, Y. Hida, Accurate and Efficient Floating Point Summation, *SIAM J. on Scientific Computing*, **25**, 1214–1248, 2003.

[DIS] C.-E. Drevet, M. N. Islam, E. Schost, Optimization Techniques for Small Matrix Multiplication, preprint, 2010.

[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.

[EG99] Y. Eidelman, I. Gohberg, On a New Class of Structured Matrices, *Integral Equations and Operator Theory*, **34**, 293–324, Birkhäuser, Basel, 1999.

[EP02] I. Z. Emiris, V. Y. Pan, Symbolic and Numerical Methods for Exploiting Structure in Constructing Resultant Matrices, *J. of Symbolic Computation*, **33**, 393-413, 2002. Proc. Version in *ISSAC 97*.

[EP03/05] I. Z. Emiris, V. Y. Pan, Improved Algorithms for Computing Determinants and Resultants, *J. of Complexity*, **21**, **1**, 43–71, 2005. Proceedings version in *Proceedings*

Table 12.4: condition numbers cond(A) of random circulant matrices A

n	**min**	**mean**	**max**	**std**
256	9.6×10^0	1.1×10^2	3.5×10^3	4.0×10^2
512	1.4×10^1	8.5×10^1	1.1×10^3	1.3×10^2
1024	1.9×10^1	1.0×10^2	5.9×10^2	8.6×10^1
2048	4.2×10^1	1.4×10^2	5.7×10^2	1.0×10^2
4096	6.0×10^1	2.6×10^2	3.5×10^3	4.2×10^2
8192	9.5×10^1	3.0×10^2	1.5×10^3	2.5×10^2
16384	1.2×10^2	4.2×10^2	3.6×10^3	4.5×10^2
32768	2.3×10^2	7.5×10^2	5.6×10^3	7.1×10^2
65536	2.4×10^2	1.0×10^3	1.2×10^4	1.3×10^3
131072	3.9×10^2	1.4×10^3	5.5×10^3	9.0×10^2
262144	6.3×10^2	3.7×10^3	1.1×10^5	1.1×10^4
524288	8.0×10^2	3.2×10^3	3.1×10^4	3.7×10^3
1048576	1.2×10^3	4.8×10^3	3.1×10^4	5.1×10^3

of 6th International Workshop on Computer Algebra in Scientific Computing (CASC '03), E. W. Mayr, V. G. Ganzha, E. V. Vorozhtzov (editors), 81–94, Technische Univ. München, Germany, 2003.

[EP10] I. Z. Emiris, V.Y. Pan, Fast Fourier Transform and Its Applications, in *Algorithms and Theory of Computation Handbook*, (Second Edition), Volume 1: General Concepts and Techniques, 1016 pp., pages 1–31 in Chapter 18, (Mikhail J. Atallah and Marina Blanton, editors), CRC Press Inc., Boca Raton, Florida, 2010.

[F07] M. Fürer, Faster Integer Multiplication, *Proceedings of 39th Annual Symposium on Theory of Computing (STOC 2007)*, 57–66, ACM Press, New York, 2007.

[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.

[GK72] I. Gohberg, N. Y. Krupnick, A Formla for the Inversion of Finite Toeplitz Matrices, *Matematicheskiie Issledovaniia* (in Russian), **7**, **2**, 272–283, 1972.

[GKO95] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gaussian Elimination with Partial Pivoting for Matrices with Displacement Structure, *Math. of Comp.,* **64(212)**, 1557–1576, 1995.

[GL96] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1996 (third addition).

[GO94] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra and Its Applications*, **202**, 163–192, 1994.

[GS72] I. Gohberg, A. Semencul, On the Inversion of Finite Toeplitz Matrices and Their Continuous Analogs, *Matematicheskiie Issledovaniia* (in Russian), **7**, **2**, 187–224, 1972.

[H79] G. Heinig, Beitrage zur spektraltheorie von Operatorbuschen und zur algebraischen Theorie von Toeplitzmatrizen, Dissertation **B**, *TH Karl-Marx-Stadt*, 1979.

[H02] N. J. Higham, *Accuracy and Stability in Numerical Analysis*, SIAM, Philadelphia, 2002 (second edition).

Table 12.5: Preconditioning tests

Type	$\nu = r$	Cond (C)
1n	1	3.21E+2
1n	2	4.52E+3
1n	4	2.09E+5
1n	8	6.40E+2
1s	1	5.86E+2
1s	2	1.06E+4
1s	4	1.72E+3
1s	8	5.60E+3
2n	1	8.05E+1
2n	2	6.82E+3
2n	4	2.78E+4
2n	8	3.59E+3
2s	1	1.19E+3
2s	2	1.96E+3
2s	4	1.09E+4
2s	8	9.71E+3
3n	1	2.02E+4
3n	2	1.53E+3
3n	4	6.06E+2
3n	8	5.67E+2
3s	1	2.39E+4
3s	2	2.38E+3
3s	4	1.69E+3
3s	8	6.74E+3
4n	1	4.93E+2
4n	2	4.48E+2
4n	4	2.65E+2
4n	8	1.64E+2
4s	1	1.45E+3
4s	2	5.11E+2
4s	4	7.21E+2
4s	8	2.99E+2

[HR84] G. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators, Operator Theory*, **13**, Birkhäuser, 1984.

[K04] I. Kaporin, The Aggregation and Cancellation Techniques As a Practical Tool for Faster Matrix Multiplication, *Theoretical Computer Science*, **315**, **2–3**, 469–510, 2004.

[KKM79] T. Kailath, S. Y. Kung, M. Morf, Displacement Ranks of Matrices and Linear Equations, *Journal Math. Analysis and Appls*, **68(2)**, 395–407, 1979.

[KV99] P. Kravanja, M. Van Barel, Algorithms for Solving Rational Interpolation Problems Related to Fast and Superfast Solvers for Toeplitz Systems, *SPIE*, 359–370, 1999.

[LDB02] X. Li, J. Demmel, D. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, S. Kang, A. Kapur, M. Martin, B. Thompson, T. Tung, D. Yoo, Design, Implementation and Testing of Extended and Mixed Precision BLAS, *ACM Transactions on Mathematical Software*, **28**, 152–205, 2002. http //crd.lbl.gov/x̃iaoye/XBLAS/.

Table 12.6: CPU time for solving an ill conditioned Toeplitz linear system (in cycles)

n	**Alg. 7.1**	**QR**	**SVD**	**QR/Alg. 7.1**	**SVD/Alg. 7.1**
512	56.3	148.4	4134.8	2.6	73.5
1024	120.6	1533.5	70293.1	12.7	582.7
2048	265.0	11728.1	–	44.3	–
4096	589.4	–	–	–	–
8192	1304.8	–	–	–	–

Table 12.7: Relative residual norms for a linear system of equations via nmb computation

n	**r**	**min**	**max**	**mean**	**std**
32	1	1.49×10^{-13}	1.36×10^{-9}	4.25×10^{-11}	1.56×10^{-10}
32	2	3.70×10^{-13}	2.13×10^{-8}	3.83×10^{-10}	2.35×10^{-9}
32	4	9.33×10^{-13}	1.08×10^{-8}	3.37×10^{-10}	1.26×10^{-9}
64	1	1.11×10^{-12}	6.87×10^{-9}	2.03×10^{-10}	7.49×10^{-10}
64	2	1.53×10^{-12}	1.21×10^{-8}	5.86×10^{-10}	1.77×10^{-9}
64	4	2.21×10^{-12}	1.27×10^{-7}	1.69×10^{-9}	1.28×10^{-8}

[LPS92] J. Laderman, V. Y. Pan, H. X. Sha, On Practical Algorithms for Accelerated Matrix Multiplication, *Linear Algebra and Its Applications*, **162–164**, 557–588, 1992.

[MP00] B. Mourrain, V. Y. Pan, Multivariate Polynomials, Duality and Structured Matrices, *J. of Complexity*, **16**, **1**, 110–180, 2000. (Proceedings Version in *STOC'98.*)

[MPR03] B. Mourrain, V. Y. Pan, O. Ruatta, Accelerated Solution of Multivariate Polynomial Systems of Equations, *SIAM J. on Computing*, **32, 2**, 435–454, 2003.

[OOT06] V. Olshevsky, I. V. Oseledets, E. E. Tyrtyshnikov, Tensor Properties of Multilevel Toeplitz and Related Matrices, *Linear Algebra and Its Applications*, **412**, 1–21, 2006.

[P72] V. Y. Pan, On Schemes for the Evaluation of Products and Inverses of Matrices (in Russian), *Uspekhi Matematicheskikh Nauk*, **27**, **5 (167)**, 249–250, 1972.

[P84] V. Y. Pan, How Can We Speed up Matrix Multiplication? *SIAM Review,* **26**, **3**, 393–415, 1984.

[P90] V. Y. Pan, On Computations with Dense Structured Matrices, *Math. of Computation*, **55, 191**, 179–190, 1990.

[P92] V. Y. Pan, Parallel Solution of Toeplitz-like Linear Systems, *J. of Complexity*, **8**, 1–21, 1992.

[P93] V. Y. Pan, Concurrent Iterative Algorithm for Toepliz-like Linear Systems, *IEEE Transactions on Parallel and Distributed Systems*, **4, 5**, 592–600, 1993.

[P93a] V. Y. Pan, Decreasing the Displacement Rank of a Matrix, *SIAM Journal on Matrix Analysis and Applications*, **14**, **1**, 118–121, 1993.

[P98/01] V. Y. Pan, Numerical Computation of a Polynomial GCD and Extensions, *nformation and Computation*, **167, 2**, 71–85, 2001. Proc. version: "Approximate Polynomial Gcds, Pad Approximation, Polynomial Zeros, and Bipartite Graphs", in Proc. *9th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA 98)*, 68-77, ACM Press, New York, and SIAM Publications, Philadelphia (1998).

Table 12.8: Relative residual norms for a linear system of equations with MLDIVIDE(A,B)

n	**r**	**min**	**max**	**mean**	**std**
32	1	6.34×10^{-3}	7.44×10^{1}	1.74×10^{0}	7.53×10^{0}
32	2	2.03×10^{-2}	1.32×10^{1}	9.19×10^{-1}	1.62×10^{0}
32	4	4.57×10^{-2}	1.36×10^{1}	1.14×10^{0}	1.93×10^{0}
64	1	3.82×10^{-3}	9.93×10^{0}	1.03×10^{0}	1.66×10^{0}
64	2	1.96×10^{-2}	1.27×10^{2}	3.09×10^{0}	1.40×10^{1}
64	4	7.13×10^{-3}	6.63×10^{0}	8.23×10^{-1}	1.20×10^{0}

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

[P10] F. Poloni, A Note on the O(n)-Storage Implementation of the GKO Algorithm, *Numerical Algorithms*, **55**, 115–139, 2010.

[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*, **409**, **2**, 255–268, 2008.

[PIMR10] 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.

[PKRK06] V. Y. Pan, M. Kunin, R. Rosholt, H. Kodal, homotopic Residual Correction Processes, *Math. of Computation*, **75**, 345–368, 2006.

[PMQR09] V. Y. Pan, B. Murphy, G. Qian, R. E. Rosholt, Error-free Computations via Floating-Point Operations, *Computers and Mathematics with Applications*, **57**, 560–564, 2009.

[PQ10] V. Y. Pan, G. Qian, Randomized Preprocessing of Homogeneous Linear Systems of Equations, *Linear Algebra and Its Applications*, **432**, 3272–3318, 2010.

[PQa] V. Y. Pan, G. Qian, On Solving Linear System with Randomized Augmentation, Tech. Reports TRs 2009009 and 2010009, *Ph.D. Program in Computer Science*, *Graduate Center, the City University of New York*, 2009 and 2010.
Available at http://www.cs.gc.cuny.edu/tr/techreport.php?id=352

[PQZa] V. Y. Pan, G. Qian, A. Zheng, On Randomized Preprocessing versus Pivoting, Tech. Reports TRs 2009010 and 2010011, *Ph.D. Program in Computer Science*, *Graduate Center, the City University of New York*, 2009 and 2010.

Available at http://www.cs.gc.cuny.edu/tr/techreport.php?id=352.

[PQZC] V. Y. Pan, G. Qian, A. Zheng, Z. Chen, Matrix Computations and Polynomial Root-finding with Preprocessing, *Linear Algebra and Its Applications*, in print.

[PS91] V. Y. Pan, R. Schreiber, An Improved Newton Iteration for the Generalized Inverse of a Matrix, with Applications, *SIAM Journal on Scientific and Statistical Computing*, **12, 5**, 1109–1131, 1991.

[PVW04] V. Y. Pan, M. Van Barel, X. Wang, G. Codevico, Iterative Inversion of Structured Matrices, *Theoretical Computer Science*, 315, **2–3** (Special Issue on Algebraic and Numerical Computing, edited by I. Z. Emiris, B. Mourrain, and V. Y. Pan), 581–592, 2004.

[PW08] V. Y. Pan, X. Wang, Degeneration of Integer Matrices Modulo an Integer, *Linear Algebra and Its Applications*, **429**, 2113–2130, 2008.

[PY99/01] V. Y. Pan, Y. Yu, Certified Computation of the Sign of a Matrix Determinant, *Algorithmica*, **30**, 708–724, 2001; Proc. version in *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, 715–724, ACM Press, New York, and SIAM Publications, Philadelphia, 1999.

[R06] G. Rodriguez, Fast Solution of Toeplitz- and Cauchy-like Least Squares Problems, *SIAM J. Matrix Analysis and Applications*, **28, 3**, 724–748, 2006.

[S80] J. T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of ACM*, **27**, **4**, 701–717, 1980.

[S98] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decompositions*, SIAM, Philadelphia, 1998.

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

[ST02] D. Spielman, S.-H. Teng, Smoothed Analysis of Algorithms, *Proc. of the International Congress of Mathematicians* (Beijing 2002), Vol. I, 597–606, Higher ED. Press, Beijing, 2002.

[T90] W. F. Trench, A Note on a Toeplitz Inversion Formula, *Linear Algebra and Its Applications*, **29**, 55–61, 1990.

[TB97] L. N. Trefethen, D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[V99] M. Van Barel, A Superfast Toeplitz Solver, 1999.
Available at http://www.cs.kuleuven.be/˜ marc/software/index.html

[VBHK01] M. Van Barel, G. Heinig, P. Kravanja, A Stabilized Superfast Solver for Nonsymmetric Toeplitz Systems, *SIAM Journal on Matrix Analysis and Applications*, **23**, **2**, 494–510, 2001.

[VK98] M. Van Barel, P. Kravanja, A Stabilized Superfast Solver for Indefinite Hankel Systems, *Linear Algebra and its Applications*, **284**, **1–3**, 335–355, 1998.

[VVM07] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices: Linear Systems* (Volume 1), The Johns Hopkins University Press, Baltimore, Maryland, 2007.

[VVM08] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices: Eigenvalue and Singular Value Methods* (Volume 2), The Johns Hopkins University Press, Baltimore, Maryland, 2008.

[W04] M. Wschebor, Smoothed Analysis of $\kappa(a)$, *J. of Complexity*, **20**, 97–107, 2004.

[W07] X. Wang, Affect of Small Rank Modification on the Condition Number of a Matrix, *Computer and Math. (with Applications)*, **54**, 819–825, 2007.

[Z79] R. E. Zippel, Probabilistic Algorithms for Sparse Polynomials, *Proceedings of EUROSAM'79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.