2009

# TR-2009009: Solving Linear Systems with Randomized Augmentation

Victor Y. Pan

Guoliang Qian

# Solving Linear Systems with Randomized Augmentation *

Victor Y. Pan[1,2],[a] and Guoliang Qian[2],[b]

[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

**Abstract**

Our randomized preprocessing of a matrix by means of augmentation counters its degeneracy and ill conditioning, uses neither pivoting nor orthogonalization, readily preserves matrix structure and sparseness, and leads to dramatic speedup of the solution of general and structured linear systems of equations in terms of both estimated arithmetic time and observed CPU time.

**Key words:** Linear systems of equations, Random augmentation, Conditioning of random matrices, Toeplitz matrices

# 1 Introduction

## 1.1 Background and our approach

Solution of a homogeneous linear system of equations $A\mathbf{y} = \mathbf{0}$ is a fundamental problem, closely related to solving nonhomogeneous linear systems and to other central subjects of matrix computations (cf. our Section 6, [16, Sections 7.2 and 11.1], and [18]).

The solution vectors $\mathbf{y}$ are called *null vectors* of matrix $A$. They form the *null space* $N(A)$. If its basis is given by the columns of a matrix $B$, then we call $B$ a *null matrix basis* for a matrix $A$.

The customary algorithms compute such bases and null vectors by employing the SVD of the input matrix or its LU and QR factorizations with pivoting. These approaches are costly. "Pivoting usually degrades the performance" [8, page 119], readily destroys matrix structure and sparseness, and threatens or undermines application of block matrix algorithms, whereas computing SVD is even a more involved and costly task.

Our alternative *randomized augmentation* of the input matrix (briefly covered also in [15, Section 12] and [16, Section 4]) is pivoting-free and orthogonalization-free and readily preserves input structure and sparseness. We prove that with a probability close to one such an augmentation fixes degeneracy while creating no numerical problems. Furthermore, an nmb for the original input is readily computed via the solution of linear systems of equations with the new matrix expected to have full rank and to have the condition number of the same order as the rank deficient input matrix.

*Supported by PSC CUNY Awards 69330–0038 and 61406–0039.

The power of our approach is naturally extended to ill conditioned inputs, which are small norm perturbations of rank deficient matrices. We prove that with a probability close to one our augmentation maps a typical ill conditioned input into a well conditioned one, but the subsequent computations generally require high accuracy. This need is obvious, because neither augmentation nor any other techniques can eliminate numerical problems for an ill conditioned input. Our approach, however, enables randomized preconditioning, that is for a typical ill conditioned input $A$ we reduce our original task to solving linear systems of equations with a matrix $C$ expected to be well conditioned. We must solve these systems with high accuracy, but for a well conditioned matrix $C$ we first obtain its approximate inverse $X$ (or alternatively its LUP or QR factorization) by computing with the IEEE standard double precision, and then iterative refinement rapidly produces the desired accurate solutions. Recall that every refinement step is essentially reduced to multiplication of the matrices $C$ and $X$ by two vectors (by using quadratic number of flops for general input and nearly linear number for Toeplitz-like input). Every step produces about $p_{\text{double}} - \log_2 \operatorname{cond} C$ new correct bits per an output value [8], [15]. Overall this implies dramatic improvement versus the known algorithms, which require cubic arithmetic time for general ill conditioned unstructured linear systems and quadratic time in the case of Toeplitz-like input.

The results of our extensive tests (the contribution of the second author) are in quite good accordance with the formal study above. In particular they show our dramatic speedup also in terms of the CPU time. Versus the QR-based solution for $n \times n$ singular Toeplitz linear systems we observed the acceleration factor $a(n)$ where $a(512) > 15$, $a(1024) > 90$, and $a(2048) > 350$ (see Table 9.1).

The first author has advanced in a variety of further efffective applications of the approach, part of which is reported in [15]–[18].

### 1.2 Organization of the paper and further extensions

We organize our paper as follows. We devote the next section to the definitions and to estimating the ranks and condition numbers of random matrices and their products with fixed matrices. In Sections 3 and 4 we present our randomized augmentation techniques and in Section 5 the related techniques of randomized post-multiplication. In Section 6 and in the Appendix we study the reduction of nonhomogeneous linear systems to homogeneous ones. In Section 7 we estimate the speedup that we achieve with our solutions versus the known algorithms, and in Section 8 we comment on application of our approach to special matrices. In Section 9 we cover our numerical tests.

**Acknowledgement.** Marc Van Barel's pointer to his Toeplitz solver in [21] was most helpful for our tests.

## 2 Definitions and basic facts

### 2.1 General and Toeplitz matrices, nmbs and annihilators

We use and extend the customary definitions of matrix computations (cf. [8], [19]). $\mathbb{C}$ (resp. $\mathbb{R}$) is the field of complex (resp. real) 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 $A$.) $(B_1, \dots, B_k) = (B_j)_{j=1}^k$ is a $1 \times k$ block matrix with blocks $B_1, \dots, B_k$. $\operatorname{diag}(B_1, \dots, B_k) = \operatorname{diag}(B_j)_{j=1}^k$ is a $k \times k$ block diagonal matrix with diagonal blocks $B_1, \dots, B_k$. $I_n$ or just $I$ denote the $n \times n$ identity matrix. $\mathbf{e}_i$ is its $i$th column vector, $i = 0, 1, \dots, n-1$. 0 denotes a matrix filled with zeros. A matrix $U$ is unitary or orthonormal if $U^H U = I$. $M(n)$ and $I(n)$ flops suffice to multiply an $n \times n$ matrix by a vector and (if possible) to invert it, respectively. $M(n) = 2n^2 - n$, $I(n) \le (2/3)n^3 + O(n^2)$ for general matrices; $M(n) < 20n \log n$, $I(n) = O(n^2)$ for Toeplitz and Hankel matrices. We also have $I(n) \le c_{\text{huge}} n^{2.376}$ for general matrices; $I(n) \le c_{\text{large}} n \log^2 n$ for Toeplitz and Hankel matrices where $c_{\text{huge}}$ and $c_{\text{large}}$ are immense and large constants, respectively (cf. [1], [3]).

range$(A)$ is the linear space generated by its columns. $\rho = \operatorname{rank} A$ denotes its rank, $\operatorname{nul} A = n - \rho$ its nullity, and $N(A)$ its null space. $\mathbf{v}$ is its null vector if $A\mathbf{v} = \mathbf{0}$. A matrix $H$ is a *complete*

*annihilator* of a matrix $A$ if range $H = N(A)$. Such an annihilator is a *null matrix basis* if it has full column rank. We use the abbreviations nmb, ca, nmb($A$), and ca($A$). Given a ca($A$), we can rely on its LUP or QR factorization or on the following simple fact to compute a nmb($A$).

**Fact 2.1.** *[16]. Suppose $H$ is a* ca($A$). *Then*
*(a) $H$ is a* nmb($A$) *if and only if* $\operatorname{nul} H = 0$ *and*
*(b) $HY$ is a* nmb($A$) *if $X$ is a* ca($H$) *and if $(X, Y)$ is a nonsingular matrix.*

Hereafter we deal with $m \times n$ input matrices $A$ having at least as many rows as columns. Otherwise, for $m > n$, we can apply the alternative techniques in [16] or shift to the case $m \le n$ based on the equations $N(A) = N(A^H A)$ or $N(A) = \cap_{i=1}^{h} N(B_i)$ provided $A = \sum_{i=1}^{h} (0, B_i, 0)^T$, $B_i$ are $k_i \times n$ matrices for $i = 1, \dots, h$, and $\sum_{i=1}^{h} k_i \ge m$. [8, Theorem 12.4.1] simplifies the transition from nmbs of the $h$ matrices $(0, B_i, 0)^T$ to a nmb($A$).

$J = J_n = (j_{i,k})_{i,k=0}^{n-1}$ is the reflection matrix, $j_{i,k} = 1$ if $i+k = n-1$, $j_{i,k} = 0$ unless $i+k = n-1$. ($J^2 = I$.) Toeplitz matrix $T = (t_{i-j})_{i=0,j=0}^{m-1,n-1}$ (resp. Hankel matrix $H = (h_{i+j})_{i=0,j=0}^{m-1,n-1}$) is defined by its $m+n-1$ entries, e.g., by its first row and first (resp. last) column. ($TJ$ and $JT$ are Hankel matrices for a Toeplitz matrix $T$, whereas $HJ$ and $JH$ are Toeplitz matrices for a Hankel matrix $H$.) Circulant matrices $C = (c_{i-j})_{i,j=0}^{n-1}$, $c_k = c_{k+n}$ for all $k$ for which $c_k$ and $c_{k+n}$ are defined, form an algebra in the class of Toeplitz matrices. They are defined by their first column vectors $\mathbf{c} = (c_i)_{i=0}^{n-1}$ and can be multiplied together and inverted in $O(n \log n)$ flops based on FFT [2].

## 2.2 Matrix norm, SVD, inverses, and condition number

$A = S_A \Sigma_A T_A^H$ is *SVD* or *full SVD* of an $m \times n$ matrix $A$ of a rank $\rho$ if $S_A S_A^H = S_A^H S_A = I_m$, $T_A T_A^H = T_A^H T_A = I_n$, $\Sigma_A = \operatorname{diag}(\widehat{\Sigma}_A, 0_{m-\rho,n-\rho})$, and $\widehat{\Sigma}_A = \operatorname{diag}(\sigma_j)_{j=1}^{\rho}$. Here $\sigma_j = \sigma_j(A) = \sigma_j(A^H)$ is the $j$th largest singular value of a matrix $A$, $j = 1, \dots, \rho$, equal to the distance from this matrix to a nearest matrix of rank $j-1$ for $j = 1, \dots, \rho$, $\sigma_1(A) = ||A|| = ||A^H||$ is the 2-norm of a matrix $A = (a_{i,j})_{i,j=1}^{m,n}$, and $||A||/\sqrt{mn} \le \max_{i,j=1}^{m,n} |a_{i,j}| \le ||A||$.

The 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 $A$.

$\operatorname{cond} A = \sigma_1(A)/\sigma_\rho(A)$ is the condition number of a matrix $A$ of a rank $\rho$. Such a matrix is *ill conditioned* if $\sigma_1(A) \gg \sigma_\rho(A)$ and is *well conditioned* otherwise. The concepts "large", "ill" and "well conditioned" are quantified in the context of the computational task and computer environment.

**Theorem 2.1.** *[17]. 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* $\operatorname{cond}(AB) \le (\operatorname{cond} A) \operatorname{cond} B$.

## 2.3 Random sampling and random matrices

$|\Delta|$ is the cardinality of a set $\Delta$. *Random sampling* of elements from a set $\Delta$ is their selection from this set at random, independently of each other, and under the uniform probability distribution on $\Delta$. A matrix is *random* if its entries are randomly sampled (from a fixed set $\Delta$). A matrix (resp. vector) is a *Gaussian random matrix* (resp. vector) with a mean $\mu$ and a variance $\sigma^2$ if it is filled with independent Gaussian random variables, all having the same mean $\mu$ and variance $\sigma^2$. If $\mu = 0$ and $\sigma^2 = 1$, this is a *standard Gaussian random matrix* (resp. vector).

**Lemma 2.1.** *[5]. For a set $\Delta$ of cardinality $|\Delta|$ (in a fixed ring), let a polynomial in $m$ variables have total degree $d$, let it not vanish identically on the set $\Delta^m$, and let the values of its variables be randomly sampled from the set $\Delta$. Then the polynomial vanishes with a probability of at most $d/|\Delta|$.*

**Corollary 2.1.** *An $m \times n$ matrix with entries sampled at random from a set $\Delta$ has full rank with a probability of at least $1 - r/|\Delta|$ for $r = \min\{m, n\}$.*

*Proof.* The determinant of a $r \times r$ matrix is a nonvanishing polynomial of degree $r$ in its entries. It remains to apply Lemma 2.1 to $r \times r$ submatrices of the input matrix. □

**Definition 2.1.** $F_X(y) = \text{Probability}\{X \le y\}$ *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* $l = \min\{m, n\}$. $F_{\mu,\sigma}(y) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{y} \exp(-\frac{(x-\mu)^2}{2\sigma^2})dx$ *is the CDF for a Gaussian random variable with a mean* $\mu$ *and a variance* $\sigma^2$. $\Phi_{\mu,\sigma}(y) = F_{\mu,\sigma}(\mu + y) - F_{\mu,\sigma}(\mu - y)$ *for* $y \ge 0$.

## 2.4 The extreme singular values of random matrices and of their products with fixed matrices

Standard Gaussian random matrices (cf. Definition 2.1) are well conditioned with a hugh probability [4], [7], and even perturbations by such random matrices $A$ is expected to make well conditioned any matrix having the norm of the order of $||A||$ [20]. Next we recall some relevant estimates. In Section 8 we comment on the respective properties of random special matrices.

**Theorem 2.2.** *Assume an* $m \times n$ *matrix* $A$ *filled with* $d$ *random variables* $X_1, \dots, X_d$. ($d = m+n-1$ *for random Toeplitz and Hankel matrices* $A$, $d = mn$ *for random general matrix* $A$.) *Write* $F_-(y) = \min_{i=1}^{d} F_{|X_i|}(y)$ *for* $y \ge 0$. *Then (a)* $F_-(y) = \Phi_{\mu,\sigma}(y)$ *where* $X_1, \dots, X_d$ *are Gaussian random variables with a mean* $\mu$ *and a variance* $\sigma$, *whereas (b)* $F_-(y) = y/a$ *(resp.* $F_-(y) = (y/a)^2$*) where* $0 \le y \le a$ *and the random variables* $X_1, \dots, X_d$ *are uniformly distributed on the real line segments* $[-a, a]$ *or* $[0, a]$ *(resp. the circle* $\{x : |x| \le a\}$ *on the complex plane or a sector of this circle). Furthermore we have the following lower and upper estimates for the CDF of the norm* $||A||$,
*c)* $1 - F_{||A||}(y) \le (1 - F_-(y/\sqrt{mn}\,))d$, *which is a trivial bound unless* $F_-(y/\sqrt{mn}\,) > 1 - 1/d$, *and*
*d)* $F_{||A||}(y) \ge (F_-(y/\sqrt{mn}\,))^d$ *if the* $d$ *random variables are independent of each other.*

**Theorem 2.3.** *(See [6, Theorem II.7].) Suppose* $A \in \mathbb{R}^{n \times n}$ *is a Gaussian random matrix with mean zero and variance* $\sigma^2$. *Then* $F_{||A||}(y) \ge 1 - e^{-x^2/2}$ *for* $x = y/\sigma - 2\sqrt{n}$.

**Theorem 2.4.** *(See [20, 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* $\sigma^2$, $W = \bar{U}(A + M)\bar{V}$, $l = \min\{m, n\}$, *and* $y \ge 0$. *Then* $F_W(y) \le 2.35y\sqrt{l}/\sigma$.

Combining the two latter theorems implies the following result.

**Corollary 2.2.** *(See [20, Theorem 3.1].) Under the assumptions of Theorem 2.4, let* $||M|| \le \sqrt{l}$. *Then* $F_{\text{cond}\,W}(y) \ge 1 - (14.1 + 4.7\sqrt{(2\ln y)/n})n/(y\sigma)$ *for all* $y \ge 1$.

On a further improvement of this bound by the factor of $\sqrt{\log n}$, see [24].

**Theorem 2.5.** *[17]. Suppose that* $G \in \mathbb{R}^{q \times m}$, $H \in \mathbb{R}^{n \times r}$, *and a random matrix* $W \in \mathbb{R}^{m \times n}$ *have full rank* $\rho$ *with probability one. Write* $r_G = \operatorname{rank} G$ *and* $r_H = \operatorname{rank} H$. *Then* $F_{GW}(y) \le F_W(y/\sigma_{r_G}(G))$ *if* $r_G = m$, *whereas* $F_{WH}(y) \le F_W(y/\sigma_{r_G}(H))$ *if* $r_H = n$.

$\text{cond}(AB)$ can be arbitrarily large even for $m \times r$ unitary matrices $A$ and $B^H$ if $m > r$, and so we cannot merely drop the above assumptions that $r_G \ge m$ and $r_H \ge n$, but the next theorem (employed in the next sections) circumvents this problem.

**Theorem 2.6.** *[17]. Suppose* $G \in \mathbb{R}^{r_G \times m}$, $H \in \mathbb{R}^{n \times r_H}$, $W \in \mathbb{R}^{m \times n}$, $\operatorname{rank} G = r_G < m$, $\operatorname{rank} H = r_H < n$, *and the assumptions of Theorem 2.4 hold. Then (a)* $F_{GW}(y) \le 2.35my\sqrt{l}/(\sigma_{r_G}(G)\sigma)$ *and (b)* $F_{WH}(y) \le 2.35y\sqrt{l}/(\sigma_{r_H}(H)\sigma)$.

# 3 A nmb of a matrix via northern augmentation

The following algorithm computes a nmb of an $m \times n$ matrix $A$ (see an alternative in Section 5).

**Algorithm 3.1. A nmb via northern augmentation.**

INPUT: *Three positive integers $\rho$, $m$, and $n$, $n \geq m \geq \rho$, an $m \times n$ matrix $A$ of a rank $\rho$, and a set $\Delta \in \mathbb{C}$ of a large cardinality $|\Delta|$.*

OUTPUT: *FAILURE or a matrix $B = \mathrm{nmb}(A)$.*

COMPUTATIONS:

1. *Write $r = n - \rho$. If $r = 0$, output the empty $\mathrm{nmb}(A)$. Otherwise generate a random matrix $V \in \Delta^{r \times n}$ such that $||V|| \approx ||A||$.*
2. *Output FAILURE and stop if the matrix $C = \begin{pmatrix} V \\ A \end{pmatrix}$ is rank deficient.*
3. *Otherwise compute a left inverse $C^{(I)}$. Compute and output the matrix $B = C^{(I)} \begin{pmatrix} I_r \\ 0 \end{pmatrix}$.*

*Correctness proof.* Let $Y = \mathrm{nmb}(A) \in \mathbb{C}^{n \times r}$ and write $B = C^{(I)} \begin{pmatrix} I_r \\ 0 \end{pmatrix}$. Then $CY = \begin{pmatrix} VY \\ 0 \end{pmatrix}$, $Y = C^{(I)} \begin{pmatrix} VY \\ 0 \end{pmatrix}$, and so $N(A) = \mathrm{range}\, Y \subseteq \mathrm{range}\, B$. It follows that $\mathrm{range}\, B = N(A)$ because $\mathrm{rank}\, B = \mathrm{rank}\, Y = r$.

**Theorem 3.1.** *The matrix $C$ in Algorithm 3.1 has full rank with a probability of at most $r/|\Delta|$.*

*Proof.* Let a $\rho \times n$ submatrix $A_{\rho,n}$ of the matrix $A$ have full rank $\rho$ and let $C_{n,n} = \begin{pmatrix} V \\ A_{\rho,n} \end{pmatrix}$. Clearly, $\det C_{n,n}$ is a polynomial of a degree of at most $r$ in the entries of the matrix $V$. It does not vanish identically in these entries (because the matrix $A_{\rho,n}$ has full rank), and so it vanishes with a probability of at most $r/|\Delta|$ for random matrix $V$ (in virtue of Lemma 2.1). □

Next we estimate $\mathrm{cond}\, C$.

**Theorem 3.2.** *Let the matrix $C$ in Algorithm 3.1 be scaled so that $||C|| \leq 1$. Let $\mathrm{rank}\, C = n$, $\mathrm{rank}\, A = \rho$, and so $\mathrm{rank}\, V = r = n - \rho$. Let $A = S_A \Sigma_A T_A^H$ be a full SVD of the matrix $A$ (where $\Sigma_A = \mathrm{diag}(\widehat{\Sigma}_A, 0)$ and $\widehat{\Sigma}_A$ is a $\rho \times \rho$ diagonal matrix of the singular values). Write*

$$\mathrm{diag}(I_r, S_A^H) C T_A = \begin{pmatrix} M \\ 0 \end{pmatrix}, \ M = \begin{pmatrix} V_0 & V_1 \\ \widehat{\Sigma}_A & 0 \end{pmatrix}. \tag{3.1}$$

*Then $\mathrm{cond}\, C \leq \frac{3}{\sigma_m(A)\sigma_r(V_1)}$.*

*Proof.* We have $(V_0, V_1) = VT_A$, and so $||(V_0^T, V_1^T)^T|| = ||V||$. Furthermore $\sigma_j(M) = \sigma_j(C)$ for $j = 1, \ldots, n$, $\sigma_m(A) \leq 1$, $\sigma_r(V_1) \leq ||V_1|| \leq 1$, $||V_0|| \leq 1$, and $M^{-1} = \begin{pmatrix} 0 & \widehat{\Sigma}_A^{-1} \\ V_1^{-1} & -V_1^{-1} V_0 \widehat{\Sigma}_A^{-1} \end{pmatrix}$, so that $||M^{-1}|| \leq ||\widehat{\Sigma}_A^{-1}|| + ||V_1^{-1}|| + ||\widehat{\Sigma}_A^{-1}||\ ||V_1^{-1}||$. Substitute $||\widehat{\Sigma}_A^{-1}|| = \frac{1}{\sigma_m(A)}$, $||V_1^{-1}|| = \frac{1}{\sigma_r(V_1)}$, and $||M^{-1}|| = \frac{1}{\sigma_n(M)} = \frac{1}{\sigma_n(C)}$ and obtain that $\mathrm{cond}\, C \leq \frac{1}{\sigma_n(C)} \leq \frac{1}{\sigma_m(A)} + \frac{1}{\sigma_r(V_1)} + \frac{1}{\sigma_m(A)\sigma_r(V_1)} \leq \frac{3}{\sigma_m(A)\sigma_r(V_1)}$. □

The theorem implies that the value $\mathrm{cond}\, C$ is within roughly the factor of $\sigma_r(V_1) = 1/||V^{-1}||$ from $\mathrm{cond}\, A$. Next we assume that $V$ is Gaussian random matrix and estimate the values $\sigma_r(V_1)$.

**Theorem 3.3.** *Let the assumptions of Theorem 2.6 hold for* $W = V$, $H = T_A \begin{pmatrix} 0 \\ I_r \end{pmatrix}$, $l = r_H = r$. *Then* $F_{V_1}(y) \le cy\sqrt{r}/\sigma$.

*Proof.* Observe that $V_1 = WH$ and $\sigma_{r_H}(H) = 1$ and apply part (b) of Theorem 2.6. □

**Corollary 3.1.** *Let the assumptions of Theorems 3.2 and 3.3 hold. Then* $F_C(y) \le cy\sqrt{r}/(\sigma_\rho(A)\sigma)$.

**Remark 3.1.** *In the SVD-free computation of a* nmb$(A)$ *in Algorithm 3.1 the matrix* $V$ *is defined independently of the matrix* $T_A$ *of the singular vectors. This suggests that the matrix* $V_1 = VT_A$ *behaves as random (and thus tends to be well conditioned) even where* $V$ *is a* weakly random *(e.g., structured) matrix defined by* $O(r)$ *random parameters.*

**Remark 3.2.** *To decrease the above upper bounds on the values* $1/\sigma_n(C)$ *and* $\operatorname{cond} C$ *one can scale the matrices* $A$ *and* $V$ *to maximize their norms subject to the bound* $||C|| \le 1$. *Suppose* $||V|| \approx ||A|| \approx 1/\sqrt{2}$. *Then having Gaussian random matrix* $V$ *we can expect (in virtue of Corollary 3.1) that* $\operatorname{cond} C$ *has the same order as* $\operatorname{cond} A = \sigma_1(A)/\sigma_\rho(A)$, $\rho = \operatorname{rank} A = m - r$.

**Remark 3.3.** *In Algorithm 3.1 we assume that the rank* $\rho$ *and the integer* $r = m - \rho$ *are given to us, but we can extend the algorithm by including the search for* $r$ *as the smallest integer for which the matrix* $C$ *has full rank or alternatively as the integer such that the matrix* $C$ *has full rank and* $AC^{(I)} \begin{pmatrix} I_r \\ 0 \end{pmatrix} = 0$.

**Remark 3.4.** *Assume an* $n \times n$ *ill conditioned input matrix* $\tilde{A} = A + E$ *of full rank that approximates an unknown well conditioned rank deficient matrix* $A$. *Now under the assumptions of Corollary 3.1, suppose* $C = \begin{pmatrix} V \\ A \end{pmatrix}$, $\tilde{C} = \begin{pmatrix} V \\ \tilde{A} \end{pmatrix}$, $V$ *is Gaussian random matrix scaled so that* $||V|| \approx ||A||$, *and the norm* $||E||$ *is a small fraction of the smallest positive singular value of the matrix* $A$. *Then the condition number* $\operatorname{cond} \tilde{C} \approx \operatorname{cond} C$ *and has the order of* $\operatorname{cond} A$.

# 4 Northwestern augmentation

## 4.1 Cas and nmbs via norhwestern augmentation: an algorithm

Next we define western augmentation and combine it with aggregation to compute a complete annihilator of a rank deficient matrix.

**Algorithm 4.1. A ca via northwestern augmentation.**

INPUT: *Two positive integers* $m$ *and* $n$, $m \le n$, *a matrix* $A \in \mathbb{C}^{m \times n}$, *and a finite set* $\Delta \in \mathbb{C}$ *of a large cardinality* $|\Delta|$.

OUTPUT: *The nullity* $r = \operatorname{nul} A$ *and a* ca$(A)$.

COMPUTATIONS:

1. *(The case of full rank input.) If the matrix* $A$ *has full rank, then either output* $r = 0$ *and the empty* nmb$(A)$ *if* $m = n$ *or otherwise, for* $m < n$, *apply Algorithm 3.1 to compute and to output the nullity* $r = \operatorname{nul} A = n - m$ *and an* $n \times r$ nmb$(A)$*; then stop.*
2. *(*Western augmentation.*) If the matrix* $A$ *is rank deficient, recursively generate* $m \times k$ *matrices* $B_k = (\mathbf{b}_i)_{i=1}^k$ *for* $k = 1, 2, \dots, q$ *by sampling their entries at random from the set* $\Delta$. *Stop where the block matrix* $(B_q, A)$ *has full rank* $m$. *(This implies that* $q \ge r = \operatorname{nul} A$.*)*
3. *(*Aggregation.*) Apply Algorithm 3.1 to compute a matrix* $\bar{Z} = \begin{pmatrix} Z_0 \\ Z_1 \end{pmatrix} = \text{nmb}((B_q, A))$ *where* $Z_0 \in \mathbb{C}^{q \times s}$, $Z_1 \in \mathbb{C}^{n \times s}$, *and* $q \le s \le q + r$.

4. *(*Auxiliary nmb.*) Apply Algorithm 4.1 to the matrix $Z_0$ to compute the nullity $r = \operatorname{nul} Z_0$ and an $s \times r$ matrix $X = \operatorname{nmb}(Z_0)$.*

5. *(*Disaggregation.*) Compute and output the $n \times r$ matrix $Y = Z_1 X = \operatorname{ca}(A)$.*

*Correctness proof* (cf. also Theorems 4.4 and 4.5 and Corollary 4.2). By the definition of the matrices $\bar{Z}$ and $X$, we have $B_q Z_0 + A Z_1 = 0$ and $B_q Z_0 X = 0$. Therefore $AY = AZ_1 X = 0$. Conversely, if $A\tilde{\mathbf{y}} = \mathbf{0}$, then $(B_q, A)\begin{pmatrix}\mathbf{0}\\ \tilde{\mathbf{y}}\end{pmatrix} = \mathbf{0}$. It follows that $\bar{Z}\tilde{\mathbf{x}} = \begin{pmatrix}\mathbf{0}\\ \tilde{\mathbf{y}}\end{pmatrix}$ for some vector $\tilde{\mathbf{x}}$ because $\bar{Z} = \operatorname{nmb}((B_q, A))$. Therefore $Z_0\tilde{\mathbf{x}} = 0$ and $Z_1\tilde{\mathbf{x}} = \tilde{\mathbf{y}}$.

**Remark 4.1.** *Having $Z_1 X = \operatorname{ca}(A)$ available, we can obtain a $\operatorname{nmb}(A)$ by applying LUP, QR factorization or Fact 2.1 to the matrix $H = Z_1 X$.*

**Remark 4.2.** *The correctness proof applies to any $q \geq r$ such that matrix $(B_q, A)$ has rank $m$, but it is desired to yield progress for smaller integers $q$ to deal with matrices $Z_0$ of a smaller size, for which the task of the ca computation is simpler. In view of Theorem 4.1 we can expect to yield a matrix $(B_q, A)$ of rank $m$ for $q = m - \operatorname{rank} A$.*

**Remark 4.3.** *Algorithm 3.1 combines the northern and western augmentations into northwestern augmentation $A \to K = \begin{pmatrix}W & V\\ B & A\end{pmatrix}$, which we study in Section 4.4. Alternatively we can combine western augmentation with postmultiplication (see Section 5).*

## 4.2 Aggregation interpretation of western augmentation

Stages 3–5 of the algorithm exemplify *aggregation methods*, which successively perform the following stages.

(a) Aggregate an input matrix $M$ into a matrix $M_1$ (in Algorithm 4.1, $M_1 = Z_0$).

(b) Compute the solution $Y_1$ for a given task, but for the aggregated input $M_1$ (in Algorithm 4.1, the task is the computation of a nmb, and we have $Y_1 = X$). At this stage, one can recursively reapply aggregation.

(c) Disaggregate the aggregated solution $Y_1$ to produce the solution $Y$ for the original input $M$.

Among many other examples of aggregation we recall the Schur Aggregation in [15], the hierarchial aggregation processes in [13], which in the 1980s served as the springboard for Algebraic Multigrid, and trilinear aggregating in [10], [12], and [14], largely responsible for designing both asymptotically and practically fastest algorithms for matrix multiplication. Fact 2.1 defines aggregation of a matrix $A$ into its complete annihilator $H$, that is $\mathbb{I} = A$, $\mathbb{I}_1 = H$ at stage (a), whereas $\mathbb{I} = H$, $\mathbb{I}_1 = X^H$ at stage (b).

## 4.3 Analysis of western augmentation

**Theorem 4.1.** *Assume that $m \leq n$ and $r = m - \operatorname{rank} A$. Then (a) the matrices $C = (B_k, A)$ in Algorithm 4.1 are rank deficient for $k < r$, whereas (b) for $k \geq r$ the matrices $B_k$ and $(B_k, A)$ are rank deficient with a probability of at most $r/|\Delta|$.*

*Proof.* $\operatorname{rank}(B_k, A) \leq \operatorname{rank} B_k + \operatorname{rank} A \leq k + \operatorname{rank} A$. This implies part (a). If $k \geq m - \operatorname{rank} A$ and the entries of the matrix $B_k$ are indeterminates, then clearly the matrices $B_k$ and $(B_k, A)$ have full rank. Now part (b) of the theorem follows from Corollary 2.1. □

Theorem 4.1 implies that $q = r$ with a probability of at least $1 - r/|\Delta|$. Next we estimate $\operatorname{cond} C$.

**Theorem 4.2.** *Suppose the matrix $C = (B_q, A)$ in Algorithm 4.1 has been scaled so that $||C|| \leq 1$. Let $\operatorname{rank} C = m$ and $\operatorname{rank} A = m - q$, so that $\operatorname{rank} B_q = q$. Let $A = S_A \Sigma_A T_A^H$ be a full SVD of the matrix $A$ (where $\Sigma_A = \operatorname{diag}(\widehat{\Sigma}_A, 0)$ and the matrix $\widehat{\Sigma}_A$ is nonsingular) and write*

$$S_A^H C \operatorname{diag}(I_q, T_A) = \begin{pmatrix}\bar{B}_0 & \widehat{\Sigma}_A & 0\\ \bar{B}_1 & 0 & 0\end{pmatrix}. \tag{4.1}$$

*Then* $\operatorname{cond} C \le \frac{3}{\sigma_{m-q}(A)\sigma_q(\bar{B}_1)}$.

*Proof.* Delete the last $n-m$ vanishing columns of the matrix in (4.1) and denote by $M$ the resulting nonsingular $m \times m$ matrix. We have $\begin{pmatrix} \bar{B}_0 \\ \bar{B}_1 \end{pmatrix} = S_A \begin{pmatrix} B_0 \\ B_1 \end{pmatrix}$, $||(\bar{B}_0^T, \bar{B}_1^T)|| = ||B_q||$, $\sigma_j(M) = \sigma_j(C)$ for $j = 1, \dots, m$. Moreover $\sigma_{m-q}(A) \le ||A|| \le 1$, $||\bar{B}_0|| \le 1$, and $\sigma_q(\bar{B}_1) \le ||\bar{B}_1|| \le 1$ because $||C|| \le 1$. Invert equation (4.1) to obtain $M^{-1} = \begin{pmatrix} 0 & \bar{B}_1^{-1} \\ \widehat{\Sigma}_A^{-1} & -\widehat{\Sigma}_A^{-1}\bar{B}_0\bar{B}_1^{-1} \end{pmatrix}$ and deduce that $||M^{-1}|| \le ||\widehat{\Sigma}_A^{-1}|| + ||\bar{B}_1^{-1}|| + ||\widehat{\Sigma}_A^{-1}||\ ||\bar{B}_1^{-1}||$. Substitute $\widehat{||\Sigma_A^{-1}||} = \frac{1}{\sigma_{m-q}(\widehat{\Sigma}_A)} = \frac{1}{\sigma_{m-q}(A)}$, $||\bar{B}_1^{-1}|| = \frac{1}{\sigma_q(\bar{B}_1)}$, and $||M^{-1}|| = \frac{1}{\sigma_m(C)}$ and obtain that $\operatorname{cond} C \le \frac{1}{\sigma_m(C)} \le \frac{1}{\sigma_{m-q}(A)} + \frac{1}{\sigma_q(\bar{B}_1)} + \frac{1}{\sigma_{m-q}(A)\sigma_q(\bar{B}_1)} \le \frac{3}{\sigma_{m-q}(A)\sigma_q(\bar{B}_1)}$. □

The theorem implies that the value $\operatorname{cond} C$ is within roughly the factor of $\sigma_r(\bar{B}_1) = 1/||V^{-1}||$ from $1/\sigma_{m-q}(A)$. Next we assume that $B_q$ is Gaussian random matrix and estimate the value $\sigma_r(\bar{B}_1)$.

**Theorem 4.3.** *Suppose* $\operatorname{rank} A = m - q$ *and the assumptions of Theorem 2.6 hold for* $W = B_q$, $G = (0, I_q)S_A^H$ *and* $r_G = l = q \le m$. *Then* $F_{\bar{B}_1}(y) \le cy\sqrt{q}/\sigma$.

*Proof.* Observe that $\bar{B}_1 = GW$ and $\sigma_{r_G}(G) = 1$ and apply part (a) of Theorem 2.6. □

**Corollary 4.1.** *Let the assumptions of Theorems 4.2 and 4.3 hold.*
*Then* $F_C(y) \le cy\sqrt{q}/(\sigma_{m-q}(A)\sigma)$.

Remarks 3.1–3.4 can be readily extended to the case of Algorithm 4.1, but we only specify the extension of Remarks 3.1 and 3.2.

**Remark 4.4.** *Recall that in the SVD-free computation of a* $\operatorname{nmb}(A)$ *in Algorithm 2 the matrix* $B_q$ *is defined independently of the matrix* $S_A$ *of the left singular vectors of the matrix* $A$. *This suggests that the matrix* $\bar{B}_1 = S_A^H B_q$ *behaves as random (and thus tends to be well conditioned) even where* $B_q$ *is a* weakly random *(e.g., random structured) matrix defined by* $O(r)$ *random parameters.*

**Remark 4.5.** *One can decrease the above upper bounds on the values* $1/\sigma_m(C)$ *and* $\operatorname{cond} C$ *by scaling the matrices* $A$ *and* $B_q$ *to maximize their norms subject to the bound* $||C|| \le 1$. *E.g., let* $||B_q|| \approx ||A|| \approx 1/\sqrt{2}$. *Then* $\operatorname{cond} \tilde{C} \approx \operatorname{cond} C$, *and by combining Theorems 4.2 and 4.3 and Corollary 4.1 we deduce that the latter value is expected to be of the order of* $\operatorname{cond} A = \sigma_1(A)/\sigma_\rho(A)$, $\rho = \operatorname{rank} A = m - q$.

## 4.4 Analysis of northwestern augmentation

**Theorem 4.4.** (a) *Assume six positive integers* $m$, $n$, $q$, $r$, $\tilde{r}$, *and* $\rho$ *such that* $\rho \le \min\{m, n\}$ *and* $\tilde{r} = \min\{m + r, n + q, \rho + q + r\}$, *a set* $\Delta$ *of cardinality* $|\Delta|$ *in a ring* $\mathcal{R}$, *and five matrices* $A \in \mathcal{R}^{m\times n}$ *of rank* $\rho$, $V$ *in* $\mathcal{R}^{r\times n}$, $B$ *in* $\mathcal{R}^{m\times q}$, $W$ *in* $\mathcal{R}^{r\times q}$, *and* $K = \begin{pmatrix} W & V \\ B & A \end{pmatrix} \in \mathcal{R}^{(m+r)\times(n+r)}$. *Then we have* $\operatorname{rank} K \le \tilde{r}$.

(b) *In addition suppose that either the entries of the three matrices* $B$, $V$ *and* $W$ *have been randomly sampled from the set* $\Delta$ *or* $B = V$ *and the entries of the matrices* $V$ *and* $W$ *have been randomly sampled from this set. Write* $l = \min\{m, n\}$, $d = \min\{m - q, n - r\}$. *Then* $\operatorname{rank} K = \tilde{r}$ *with a probability of at least* $1 - \frac{r+q}{|\Delta|}$ *if* $\rho \le d$ *and with a probability of at least* $1 - \frac{r+q+d-\rho}{|\Delta|}$ *if* $\rho \ge d$.

**Theorem 4.5.** *Under the assumptions of part (a) of Theorem 4.4, suppose*

$$K^{(I)}K = I_{n+q} \text{ and } Y^{(I)}Y = I_r \tag{4.2}$$

*for some matrices* $K^{(I)}$ *and* $W^{(I)}$*. Then*

$$N(A) \subseteq \text{range}((0, I_n)K^{(I)} \begin{pmatrix} 0 \\ B \end{pmatrix}). \tag{4.3}$$

*Furthermore if* $\text{rank}\, B \le \text{nul}\, A$*, then*

$$(0, I_n)K^{(I)} \begin{pmatrix} 0 \\ B \end{pmatrix} = \text{nmb}(A). \tag{4.4}$$

*Proof.* Let $\mathbf{y} \in N(A)$ and $\mathbf{x} \in \mathbb{C}^r$. Then $K \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} W\mathbf{x} + V\mathbf{y} \\ B\mathbf{x} \end{pmatrix}$. Substitute $\mathbf{x} = -W^{(I)}V\mathbf{y}$ and obtain that $K \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ B\mathbf{x} \end{pmatrix}$. Therefore $\mathbf{y} = (0, I_n)K^{(I)} \begin{pmatrix} \mathbf{0} \\ B\mathbf{x} \end{pmatrix}$. This proves claim (4.3), which implies claim (4.4) if $\text{rank}\, B \le \text{nul}\, A$ because $\text{rank}((0, I_n)K^{(I)} \begin{pmatrix} 0 \\ B \end{pmatrix}) \le \text{rank}\, B$. □

**Corollary 4.2.** *Under the assumptions of part (a) of Theorem 4.4, suppose equation (4.2) holds and write* $Y = (0, I_n)K^{(I)} \begin{pmatrix} 0 \\ B \end{pmatrix}$*. Then (a)* $YZ$ *is a* $\text{ca}(A)$ *if* $Z$ *is a* $\text{ca}(AY)$ *and furthermore (b)* $Z$ *is a* $\text{ca}(AY)$ *if* $YZ$ *is a* $\text{ca}(A)$ *and if* $\text{rank}\, Y = r$*.*

*Proof.* (a) Clearly $A(YZ) = (AY)Z = 0$ if $Z$ is a $\text{ca}(AY)$. Conversely let $A\mathbf{u} = \mathbf{0}$. Then $\mathbf{u} = Y\mathbf{v}$ for some vector $\mathbf{v}$ in virtue of (4.3). Therefore $AY\mathbf{v} = A\mathbf{u} = \mathbf{0}$. It follows that $\mathbf{v} = Z\mathbf{z}$ for some vector $\mathbf{z}$ because $Z$ is a $\text{ca}(AY)$. Consequently $\mathbf{u} = Y\mathbf{v} = YZ\mathbf{z}$.

(b) Surely $(AY)Z = A(YZ) = 0$ if $WZ$ is a $\text{ca}(A)$. Conversely let $AY\mathbf{u} = A(Y\mathbf{u}) = \mathbf{0}$. Then $Y\mathbf{u} = YZ\mathbf{v}$ for some vector $\mathbf{v}$ because $YZ$ is a $\text{ca}(A)$. Therefore $\mathbf{u} = Z\mathbf{v}$ since $\text{rank}\, Y = r$. □

Let us estimate $\text{cond}\, K$ provided the matrix $K$ is nonsingular.

**Theorem 4.6.** *In addition to the assumptions of Theorem 4.5, suppose* $||K|| \le 1$*,* $m+q = n+r$*, the matrix* $K$ *is nonsingular, and* $A = S_A \Sigma_A T_A^H$ *is a full SVD of the matrix* $A$ *where* $\Sigma_A = \text{diag}(\widehat{\Sigma}_A, 0)$*, the matrix* $\widehat{\Sigma}_A$ *is nonsingular, and* $\text{rank}\, A = \text{rank}\, \Sigma_A = \text{rank}\, \widehat{\Sigma}_A = \rho$*. Write*

$$M = \text{diag}(I_r, S_A^H)K\, \text{diag}(I_q, T_A) = \begin{pmatrix} W & V_0 & V_1 \\ \bar{B}_0 & \widehat{\Sigma}_A & 0 \\ \bar{B}_1 & 0 & 0 \end{pmatrix}, \tag{4.5}$$

$q = \text{rank}\, \bar{B}_1 = m - \rho$*, and* $r = \text{rank}\, V_1 = n - \rho$*. (Here* $\bar{B}_1$ *(resp.* $V_1$*) is a dummy empty matrix if* $\rho = m$ *(resp.* $\rho = n$*).) Then* $\sigma_{n+r}(K) \ge \frac{1}{8}\sigma_\rho(A)\sigma_q(\bar{B}_1)\sigma_r(V_1)$ *and so* $\text{cond}\, K \le \frac{1}{\sigma_{n+r}(K)} \le \frac{8}{\sigma_\rho(A)\sigma_q(\bar{B}_1)\sigma_r(V_1)}$*.*

*Proof.* By assumption the matrices $K$ and thus also $M$, $\bar{B}_1$, $\Sigma_A$, and $V_1$ are nonsingular. Equation (4.5) implies that $M^{-1} = \begin{pmatrix} 0 & 0 & \bar{B}_1^{-1} \\ 0 & \widehat{\Sigma}_A^{-1} & -\widehat{\Sigma}_A^{-1}\bar{B}_0\bar{B}_1^{-1} \\ V_1^{-1} & -V_1^{-1}V_0\widehat{\Sigma}_A^{-1} & V_1^{-1}(V_0\widehat{\Sigma}_A^{-1}\bar{B}_0 - W)\bar{B}_1^{-1} \end{pmatrix}$, $||(\bar{B}_0^T, \bar{B}_1^T)|| = ||B||$, $(V_0, V_1) = S_A^H V$, $||(V_0, V_1)|| = ||V||$, $\sigma_j(M) = \sigma_j(K)$ for $j = 1, \dots, n+r = m+q$. Moreover $||A|| \le 1$, $||B|| \le 1$, $||V|| \le 1$, and $||W|| \le 1$ because $||K|| = ||M|| \le 1$.

Deduce that $||K^{-1}|| \le f_1(A, \bar{B}_1, V_1) + f_2(A, \bar{B}_1, V_1) + f_3(A, \bar{B}_1, V_1)$ where $f_1(A, \bar{B}_1, V_1) = ||\widehat{\Sigma}_A^{-1}|| + ||\bar{B}_1^{-1}|| + ||V_1^{-1}||$, $f_2(A, \bar{B}_1, V_1) = ||\widehat{\Sigma}_A^{-1}||\, ||\bar{B}_1^{-1}|| + ||\widehat{\Sigma}_A^{-1}||\, ||V_1^{-1}|| + ||\bar{B}_1^{-1}||\, ||V_1^{-1}||$, $f_3(A, \bar{B}_1, V_1) = ||\widehat{\Sigma}_A^{-1}||\, ||\bar{B}_1^{-1}||\, ||V_1^{-1}||$.

Substitute $||\widehat{\Sigma}_A^{-1}|| = \frac{1}{\sigma_\rho(\widehat{\Sigma}_A)} = \frac{1}{\sigma_\rho(A)}$, $||\bar{B}_1^{-1}|| = \frac{1}{\sigma_q(\bar{B}_1)}$, $||V_1^{-1}|| = \frac{1}{\sigma_r(V_1)}$, and $||K^{-1}|| = \frac{1}{\sigma_{\tilde{r}}(K)}$ for $\tilde{r}$ in Theorem 4.4, and obtain that $\frac{1}{\sigma_{\tilde{r}}(K)} \le f_1(\sigma) + f_2(\sigma) + f_3(\sigma)$ where $f_1(\sigma) = \frac{1}{\sigma_\rho(A)} + \frac{1}{\sigma_q(\bar{B}_1)} + \frac{1}{\sigma_r(V_1)}$, $f_2(\sigma) = \frac{1}{\sigma_\rho(A)\sigma_q(\bar{B}_1)} + \frac{1}{\sigma_\rho(A)\sigma_r(V_1)} + \frac{1}{\sigma_q(\bar{B}_1)\sigma_r(V_1)}$, and $f_3(\sigma) = \frac{1}{\sigma_\rho(A)\sigma_q(\bar{B}_1)\sigma_r(V_1)}$. It follows that $\sigma_{n+r}(K) \ge \frac{1}{7}\sigma_\rho(A)\sigma_q(\bar{B}_1)\sigma_r(V_1)$. □

Theorems 3.3 and 4.3 bound the values $\sigma_q(\bar{B}_1)$ and $\sigma_r(V_1)$ provided $B$, $V$, and $W$ are Gaussian random matrices.

Remarks 3.1–3.4 can be readily extended to the case of northwestern augmentation, but we only specify the extension of Remark 3.2.

**Remark 4.6.** *To decrease the above upper bounds on the values* $1/\sigma_m(A)$ *and* $\operatorname{cond} C$*, we can scale the matrices* $A$*,* $B$ *and* $V$ *to maximize their norms subject to the bound* $||K|| \le 1$*. E.g., let* $||A|| \approx ||B|| \approx ||(W, V)|| \approx 1/2$*. Then by combining the estimates of Theorems 3.3, 4.3, and 4.6 we obtain that* $\operatorname{cond} K$ *is expected to have the order of* $\operatorname{cond} A = \sigma_1(A)/\sigma_\rho(A)$ *(cf. Remarks 3.4 and 4.5).*

**Remark 4.7.** *For a Hermitian matrix* $A$*, we can yield a Hermitian matrix* $K$ *by choosing* $B = V^H$ *and* $W = W^H$*. One can readily extend the results of the present section to this case.*

# 5 Post-multiplication as an alternative to the northern augmentation

Suppose we seek a $\operatorname{nmb}(\tilde{A})$ of an $m \times \tilde{n}$ matrix $\tilde{A} = (\tilde{A}_w, \tilde{A}_e)$ where $m \le \tilde{n}$ and the $m \times m$ western block $\tilde{A}_w$ is nonsingular. (E.g., a matrix $\tilde{A} = (B_q, A)$ for $\tilde{n} = n + q$ can be obtained in Stage 2 of Algorithm 4.1). Then we can immediately compute a $\operatorname{nmb}(\tilde{A}) = \begin{pmatrix} -\tilde{A}_w^{-1}\tilde{A}_e \\ I_{\tilde{n}-m} \end{pmatrix}$.

If the block $\tilde{A}_w$ is singular but the matrix $\tilde{A}$ has full rank $m$, we can try to yield a nonsingular $m \times m$ matrix $\tilde{A}S$ for an appropriate $\tilde{n} \times m$ matrix $S$ and then obtain a $\operatorname{nmb}(\tilde{A})$. The following algorithm employs these observations.

**Algorithm 5.1. A nmb via post-multiplication.**

INPUT *and* OUTPUT *as in Algorithm 3.1, except that we have an* $m \times \tilde{n}$ *input matrix* $\tilde{A}$ *rather than* $A \in \mathbb{C}^{m \times n}$ *for* $m \le \tilde{n}$*.*

COMPUTATIONS:

1. *Generate a nonsingular* $\tilde{n} \times \tilde{n}$ *matrix* $W = (S, T)$ *where* $S \in \mathbb{C}^{\tilde{n} \times m}$*.*
2. *Compute the* $m \times m$ *matrix* $\tilde{A}S$*. If it is singular, then output FAILURE and stop.*
3. *Otherwise compute and output the matrix* $W \begin{pmatrix} -(\tilde{A}S)^{-1}\tilde{A}T \\ I_{\tilde{n}-m} \end{pmatrix}$*, a* $\operatorname{nmb}(A)$*.*

Next assume that the input matrix $A$ of full rank has singular western block $A_w$ and show some relevant choices of the matrix $S$.

**Theorem 5.1.** *[8]. Let the matrix* $\tilde{A}$ *in Algorithm 5.1 have full rank and write* $S = \tilde{A}^H$*. Then the matrix* $\tilde{A}S$ *is nonsingular and* $\operatorname{cond}(\tilde{A}S) = (\operatorname{cond}\tilde{A})^2$*.*

**Theorem 5.2.** *Assume that* $m \le \tilde{n}$*, an* $m \times \tilde{n}$ *matrix* $\tilde{A}$ *has full rank, and* $S$ *is an* $\tilde{n} \times m$ *Toeplitz (resp. general) matrix with* $m + \tilde{n} - 1$ *(resp.* $m\tilde{n}$*) random entries sampled from a finite set* $\Delta$ *of cardinality* $|\Delta|$*. Then the matrix* $\tilde{A}S$ *is nonsingular (in which case the matrix* $S$ *has full rank* $m$*) with a probability of at least* $1 - m/|\Delta|$*.*

*Proof.* $\det(\tilde{A}S)$ is a polynomial of degree at most $m$ in the entries of the matrix $S$. The polynomial does not vanish identically in these entries because the matrix $\tilde{A}$ has full rank. Now the theorem follows from Lemma 2.1. □

**Theorem 5.3.** *Assume that* $m \le \tilde{n}$*, an* $m \times \tilde{n}$ *matrix* $\tilde{A}$ *has full rank and the* $m \times m$ *matrix* $\tilde{A}S$ *is nonsingular. Then* $\sigma_m(\tilde{A}S) \ge \sigma_m(\tilde{A})\sigma_m(S)$ *and consequently* $\operatorname{cond}(\tilde{A}S) \le (\operatorname{cond}\tilde{A})\operatorname{cond} S$*.*

*Proof.* The theorem follows from Theorem 2.1. □

**Theorem 5.4.** *If the assumptions of Theorem 2.6 hold for $G$ replaced by $\tilde{A}$, $X$ by $S$, $r_G$ and $l$ by $m$, and $m$ by $\tilde{n}$, then $F_{\tilde{A}S}(y) \le cy\sqrt{m}/(\sigma_m(\tilde{A})\sigma)$.*

*Proof.* The theorem follows from part (a) of Theorem 2.6. □

By avoiding northern augmentation, Algorithm 5.1 saves some flops and memory space versus Algorithm 3.1. We can achieve further saving provided the matrix $\tilde{A} = (B_q, A)$ has been output by Algorithm 4.1. In this case we can choose $W = \mathrm{diag}(I_q, G)$ where either $G = A^H$ or $G$ is a random $n \times m$ matrix scaled so that $||G|| \approx 1$. One can readily modify Theorems 5.1–5.4 to cover this case.

# 6 The solution of a nonhomogeneous linear system of equations

In the previous sections we reduced the solution of a homogeneous linear system $A\mathbf{y} = \mathbf{0}$ to the solution of nonhomogeneous ones, $CW = U$. Conversely, we can readily reduce the solution of a nonhomogeneous and nonsingular linear system $A\mathbf{y} = \mathbf{b}$ to computing the null vector $\mathbf{z} = \begin{pmatrix} 1/\eta \\ \mathbf{y} \end{pmatrix}$ for the matrix $M = (-\eta\mathbf{b}, A)$ and a nonzero scalar $\eta$.

**Claim 6.1.** *Suppose $C = (-\mathbf{b}, A)$, $A = S_A\Sigma_A T_A^H$ is a full SVD of an $n \times n$ matrix $A$, $S_A^H S_A = S_A S_A^H = T_A^H T_A = T_A T_A^H = I_n$, $\Sigma = \mathrm{diag}(\sigma_i)_{i=1}^n$, $\sigma_i = \sigma_i(A)$ for all $i$, $\mathbf{f} = (f_i)_{i=1}^n = -S^H\mathbf{b}$, $||A|| = ||\mathbf{b}|| = ||\mathbf{f}|| = 1$, $\sigma_{n-1} - 2cn\sqrt{n}\sigma_n > 0$ and $1/|f_n| \le cn$ for a constant $c$. (We can expect that the latter bound holds where $\mathbf{b}$ is a random vector independent of the matrix $S$ and normalized by scaling.) Then $\mathrm{cond}\, C = \sigma_n(C) \le 2cn\sqrt{n}/(\sigma_{n-1} - 2cn\sqrt{n}\sigma_n)$.*

*Proof.* Write $G = (\mathbf{f}, \Sigma)(I_n, 0)$, that is we define the matrix $G$ by zeroing the $(n, n+1)$st entry $\sigma_n$ of the matrix $(\mathbf{f}, \Sigma)$. Observe that the matrices $(\mathbf{f}, \Sigma)$ and $G$ have a common right inverse $F = W\,\mathrm{diag}(1/f_n, I_{n-1})Z_1$, that is $(\mathbf{f}, \Sigma)F = GF = I_n$, provided $W = \begin{pmatrix} 1 & \mathbf{0}^H \\ \mathbf{g} & \tilde{\Sigma} \\ 0 & \mathbf{0}^H \end{pmatrix}$, $\mathbf{g} = (-f_i/\sigma_i)_{i=1}^{n-1}$, $\tilde{\Sigma} = \mathrm{diag}(1/\sigma_i)_{i=1}^{n-1}$, and $Z_1 = (z_{i,j})_{i,j=0}^{n-1}$ is the $n \times n$ matrix of cyclic permutation, $z_{i,j} = 1$ where $j = i - 1 \mod n$ and $z_{i,j} = 0$ for the other pairs $(i, j)$. Further observe that $||F||_\infty \le \max_{i=1}^{n-1}(1 + |f_i|)/(|f_n|\sigma_i) \le 2/(|f_n|\sigma_{n-1})$. Therefore $\mathrm{cond}\, G = 1/\sigma_n(G) \le ||F|| \le \sqrt{n}||F||_\infty \le 2\sqrt{n}/(|f_n|\ \sigma_{n-1}) \le 2cn\sqrt{n}/\sigma_{n-1}$. We have $||(\mathbf{f}, \Sigma)|| = 1$, $\sigma_n(\mathbf{f}, \Sigma) \ge \sigma_n(G) - \sigma_n$ since $||G - (\mathbf{f}, \Sigma)|| = \sigma_n$. It follows that $\mathrm{cond}\, C = \mathrm{cond}(\mathbf{f}, \Sigma) \le 2cn\sqrt{n}/(\sigma_{n-1} - 2cn\sqrt{n}\sigma_n)$. □

To compute a null vector of the matrix $C = (-\eta\mathbf{b}, A)$ we can apply the algorithms in the previous sections to this matrix expecting that its condition number $\mathrm{cond}\, C$ has the order of $\sigma_1/\sigma_{n-1}$.

# 7 Our acceleration can be dramatic

Suppose we are dealing with an ill conditioned matrix $A$ and well conditioned matrix $C$, such that the ratio $\sigma_1/\sigma_n$ is large, whereas the ratio $\sigma_1/\sigma_{n-q}$ is not large. Furthermore assume that the integer $q$ is small. (This covers a large input class of ill conditioned matrices $A$ excluding just the ones lying on or near an algebraic variety of smaller dimension (cf. Remark 4.3).) Then the computations with the matrix $A$ require high acccuracy to avoid large output errors, but we can ensure high output accuracy in solving the auxiliary linear systems with the matrix $C$ by applying iterative refinement [8], [9], [15], [19]. From these linear systems we obtain the $q \times s$ matrix $Z_0$, and in the case of smaler integers $q$ this stage dominates the cost of the subsequent computations.

Now let us specify the resulting decrease of the computational cost. Suppose we have computed an approximate inverse $X \approx C^{-1}$ (or PLU or QR factorization for the matrix $C$) with $p$-bit precision by using $I(n)$ flops with double precision, where $I(n)$ is a function cubic in $n$ for general and quadratic for Toeplitz-like matrices $C$. Every step of iterative refinement takes $M(n)$ double precision flops (where $M(n)$ is in $O(n^2)$ for general and in $O(n \log n)$ for Toeplitz-like matrices $C$) and produces

about $p-\log_2 \operatorname{cond} C$ new correct bits per an output value. To obtain the solution with the required output precision $p_t$ we need about $h = p_t/(p - \log_2 \operatorname{cond} C)$ steps, that is $I(n) + hM(n)$ double precision flops. This is an improvement roughly by the factor $I(n)/M(n)$ (that is by the order of magnitude) versus the standard solution, which require $I(n)$ flops with the high precision $p_t$. Our tests in Section 9.1 show a dramatic acceleration of the solutions of Toeplitz linear systems also in terms of the CPU time.

# 8 Augmentation of special input matrices

We showed the effeciency of our approach where it is applied for smaller integers $q$, for both general and structured matrices. Furthermore for structured matrices $A$ and for smaller $q$ the augmentation $A \to (B_q, A)$ causes only a limited deterioration of the structure (i.e., the displacement rank grows by at most $q$). An attractive feature of the augmentation, however, is the option of keeping the structure intact completely. In such a structured augmentation, we use fewer random parameters. Are they sufficient to support our proof of the expected preconditioning power of the augmentation? All our theorems are immediately extended except for the results from [20] that random matrices are expected to be well conditioned. The extension of these results to structured matrices is a known open problem. Remarks 3.1 and 4.4 suggest that our augmentation is expected to work as preconditioning even if the latter extension is invalid. Our tests in Section 9.1 empirically support this informal argument. The tests in [17] show that random Toeplitz matrices tend to be well conditioned, almost as much as random general matrices.

# 9 Numerical tests

In a series of numerical experiments performed in the Graduate Center of the City University of New York, we tested our algorithms for computing nmbs and null vectors of general and Toeplitz matrices. We conducted the tests 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 $\{x : \ 0 \le x < 1\}$. To shift to the range $\{y : \ b \le y \le a + b\}$ for fixed real $a$ and $b$, we applied the linear transform $x \to y = ax + b$. CPU time was measured with the mclock function. We computed QR factorizations and SVDs by applying the LAPACK procedures DGEQRF and DGESVD, respectively.

## 9.1 Computations with Toeplitz matrices

**a) Generation of singular Toeplitz matrices**

To generate an $n \times n$ singular Toeplitz matrix, we first sampled $2n-2$ random entries $a_{i,j}$ for $j = 1$, $i = 1, \dots, n-1$ and for $i = 1$, $j = 2, \dots, n$ in the range $[-1, 1)$, then defined the $(n-1)^2$ entries $a_{i+1,j+1} = a_{i,j}$ for $i, j = 1, \dots, n-1$, and finally set $a_{n,1} = 0$. We arrived at an $n \times n$ Toeplitz matrix $A_0 = (a_{i,j})_{i,j=1}^{n}$, computed the entry $y_{n,1}$ of its inverse $A_0^{-1} = (y_{i,j})_{i,j=0}^{n-1}$, and changed the $(1,n)$th entry of the matrix $A_0$ into $a_{n,1} = -1/y_{n,1}$. (As we expected in virtue of Lemma 2.1, we always had $y_{n,1} \det A_0 \neq 0$ in our tests. Had we had $y_{n,1} = 0$, we could have regenerated the matrix $A_0$, whereas had it been singular, we would have stopped the computations and output it.)

The resulting matrix $A = (a_{i,j})_{i,j=1}^{n}$ had nullity one. Indeed it was a rank-one A-modification of a nonsingular matrix $A_0$, whereas $A\mathbf{y} = \mathbf{0}$ for $\mathbf{y} = A_0^{-1}\mathbf{e}_0$ because $A_0\mathbf{y} = \mathbf{e}_0$, $A = A_0 - \frac{1}{y_{n,1}}\mathbf{e}_0\mathbf{e}_{n-1}^T$, and $\mathbf{e}_{n-1}^T\mathbf{y} = y_{n,1}$.

**b) Augmentation of singular Toeplitz matrices and the computation of their null vectors**

We embedded our $n \times n$ singular Toeplitz matrix $A = (a_{i,j})_{i,j=1}^{n}$ into an $(n+1) \times (n+1)$ Toeplitz matrix $K = (a_{i,j})_{i,j=0}^{n} = \begin{pmatrix} w & \mathbf{v}^T \\ \mathbf{b} & A_0 \end{pmatrix}$ for $w = a_{0,0}$, $\mathbf{b} = (a_{i,0})_{i=1}^{n}$, and $\mathbf{v} = (a_{0,j})_{j=1}^{n}$. We defined the entries $a_{i,0}$ and $a_{0,j}$ for $i, j = 0, 1, \ldots, n-1$ by applying the equations $a_{i,j} = a_{i+1,j+1}$ and sampled the two entries $a_{n,0}$ and $a_{0,n}$ at random in the range $[-1, 1)$. For such a matrix $K$ we applied Theorem 4.5 for $r = 1$, to compute a null vector of the matrix $A$ given by the vector $(0, I_n)K^{-1}\begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix}$. This amounted to solving a nonsingular Toeplitz linear systems of equations with the matrix $K$. For this task we applied the code in [21], based on the algorithms in [11], [22], [23]. For comparison we also obtained the null vectors of the same matrices $A$ based on computing their QR factorizations and SVDs. We have a little decreased the CPU time by using QR (rather than QRP) factorization. The latter one, that is QR factorization with pivoting (performed by LAPACK procedures DGEQPF and DGEQP3) is recommended for dealing with ill conditioned inputs [8, Section 5.5], but we avoided them in our tests.

**c) Output data in the tests with Toeplitz matrices**

We use the abbreviations "n.-w.a.", "QR", and "SVD" as our pointers to the northwestern augmentation (based on Theorem 4.5), QR factorization, and SVD, respectively. Table 9.1 covers our computation of null vectors for Toeplitz input matrices. It shows the CPU time of this computation for each of the three methods as well as the ratios of these data for the QR-based and SVD-based solutions versus the algorithm based on the northwestern augmentation and Theorem 4.5. The ratios are displayed in the last two columns of the table. The CPU time is measured in terms of the CPU cycles. They can be converted into seconds by dividing them by a constant CLOCKS_PER_SEC, which is 1000 on our platform.

In all our tests the computed approximate null vectors $\mathbf{y}$ had relative residual norms $\frac{||A\mathbf{y}||}{||A||\,||\mathbf{y}||}$ of the order of $10^{-17}$.

All data are average over 100 tests for each input size $2^k$ from 256 to 8192. The table entries are marked by a hyphen "-" where the tests required too long runtime and were not completed.

Table 9.1: CPU time for computing null vectors of Toeplitz matrices

| size | n.-w.a. | QR | SVD | QR/n.-w.a. | SVD/n.-w.a |
|---|---|---|---|---|---|
| 256 | 3.8 | 18.4 | 317.8 | 4.8 | 83.6 |
| 512 | 8.0 | 148.0 | 5242.1 | 18.5 | 655.3 |
| 1024 | 16.1 | 1534.2 | 87371.2 | 97.0 | 5522.6 |
| 2048 | 33.6 | 11750.3 | – | 357.7 | – |
| 4096 | 79.5 | – | – | – | – |
| 8192 | 169.5 | – | – | – | – |

## 9.2 Computations with unstructured matrices

**a) Generation of input matrices**

We first fixed pairs of $n$ and $k$ for $n = 64, 128$ and $k = 7$. Then for every pair $(n, k)$ we generated $m = 100$ instances of matrices $A$, $B$, $V_0$, and $V_1$ and vectors $\mathbf{b}$ as follows.

The matrices $A$ have been computed as the error-free products $S\Sigma T^H$ where $S$ and $T$ were $n \times n$ random orthonormal matrices (generated with double precision) and $\Sigma = \mathrm{diag}(\sigma_j)_{j=1}^{n}$, $\sigma_{n-j} = 10^{j-17}$ for $j = 1, \ldots, k$, and $\sigma_{n-j} = 1/(n-j)$ for $j = k+1, \ldots, n-1$ (cf. [9, Section 28.3]).

$B$ was random $n \times k$ matrix with $||B|| = ||A||$.

$V$ was $k \times (n+k)$ matrix $V = (V_0, V_1)$ where and $V_0$ was the $k \times k$ identity matrix $I_k$ and $V_1 = B^T$.

For every choice of these matrices we performed preconditioning tests and the solution tests as follows.

**b) Preconditioning tests**

We computed $m$ ratios $\frac{\text{cond } A}{\text{cond } M}$ for $M = \begin{pmatrix} V_0 & V_1 \\ B & A \end{pmatrix}$.

Table 9.2 displays the average (mean), minimum, maximum, and standard deviation for the $m$ ratios for $n = 64$ and $n = 128$.

Table 9.2: ratios $\frac{\text{cond } A}{\text{cond } M}$

| **matrix size** | **min** | **max** | **mean** | **std** |
|---|---|---|---|---|
| $64 \times 64$ | $3.29 \times 10^{9}$ | $1.65 \times 10^{13}$ | $2.49 \times 10^{12}$ | $2.60 \times 10^{12}$ |
| $128 \times 128$ | $8.27 \times 10^{8}$ | $2.56 \times 10^{12}$ | $5.51 \times 10^{11}$ | $6.44 \times 10^{11}$ |

**c) The solution tests**

In the solution tests we solved nonsingular linear systems $A\mathbf{y} = \mathbf{b}$ where $A$ was the matrix generated above, $\mathbf{b}$ was a random vector, and we scaled them to have $||\mathbf{b}|| = ||A|| = 1$. We first computed the null vector $\mathbf{z}$ of the matrix $(-\mathbf{b}, A)$, then scaled it to obtain the vector $(1, \mathbf{y})^H$, and finally output the solution vector $\mathbf{y}$.

Tables 9.3 and 9.4 display the average (mean), minimum, maximum, and standard deviation for the relative residual norms $\frac{||A\mathbf{y}-\mathbf{b}||}{||\mathbf{y}||}$ in our tests for $n = 64$ and $n = 128$, respectively. For each input instance we computed the solution in two ways, that is by performing two iterations of the extended iterative refinement and with no such iteratiion.

Table 9.3: relative residual norms in the solution tests with $64 \times 64$ inputs

| **refinement** | **min** | **max** | **mean** | **std** |
|---|---|---|---|---|
| 2 iterations | $7.89 \times 10^{-48}$ | $8.26 \times 10^{-44}$ | $1.40 \times 10^{-45}$ | $8.47 \times 10^{-45}$ |
| no iteration | $1.43 \times 10^{-31}$ | $7.30 \times 10^{-28}$ | $1.69 \times 10^{-29}$ | $9.12 \times 10^{-29}$ |

Table 9.4: relative residual norms in the solution tests with $128 \times 128$ inputs

| **refinement** | **min** | **max** | **mean** | **std** |
|---|---|---|---|---|
| 2 iterations | $1.31 \times 10^{-46}$ | $1.37 \times 10^{-43}$ | $4.11 \times 10^{-45}$ | $1.67 \times 10^{-44}$ |
| no iteration | $8.57 \times 10^{-31}$ | $1.92 \times 10^{-27}$ | $5.12 \times 10^{-29}$ | $2.55 \times 10^{-28}$ |

# Appendix

## A Solution of a linear matrix equation

Suppose we seek the solution of a matrix equation $AY = B$ for $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{n \times k}$. We can solve $k$ linear systems $A\mathbf{y}_i = \mathbf{e}_i$ for $i = 1, \ldots, k$, concurrently or successively, but in some cases

we can save flops if we reduce the task to computing a nmb for the matrix $(-\frac{||A||}{||B||}B, A)$. Let us specify the recovery of the solution $Y$ from a nmb$(-\frac{||A||}{||B||}B, A)$. To simplify the notation assume that $||A|| = ||B||$.

**Algorithm A.1. Solution to a linear matrix equation given a nmb.**

INPUT: *four positive integers $k$, $m$, $n$, and $\nu$, such that $k \le m \le n$, $k \le \nu$, an $m \times k$ matrix $B$ of full rank $k$, an $m \times n$ matrix $A$, $\widehat{Y} = \text{nmb}(\widehat{A}) \in \mathbb{C}^{(n+k)\times\nu}$ for the matrix $\widehat{A} = (-B, A)$, and a Subroutine FULL·RANK, which tests whether a given matrix has full rank.*

OUTPUT: *an $n \times k$ matrix $Y$ satisfying the matrix equation $AY = B$ or INCONSISTENT if the equation has no solution.*

COMPUTATIONS:

*Write $\widehat{Y} = \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix}$ where $Y_0$ and $Y_1$ are $k \times \nu$ and $n \times \nu$ matrices, respectively. If the matrix $Y_0$ is rank deficient, output INCONSISTENT. Otherwise compute a nonsingular matrix $Y_0^-$ satisfying the matrix equation $Y_0 Y_0^- = I_k$. Then compute and output the $n \times k$ matrix $Y = Y_1 Y_0^-$.*

Correctness of the algorithm is readily verified.

**Remark A.1.** *We can generalize the augmentation trick in Section 6 and extend Algorithm A.1 to obtain the solution matrix $Y$ from a nmb$((B_q, A))$ where the last $k$ columns of the matrix $B_q$ form the matrix $B$ and the other columns are sampled at random and properly scaled. In numerical implementation of Algorithm A.1 and the latter extension we can fail if the matrix $Y_0$ is nonsingular but ill conditioned, but then we can reduce our task to solving $h$ matrix equations $AY^{(i)} = B^{(i)}$, $i = 1, \dots, h$, where $\tilde{B} = (B^{(i)})_{i=1}^h$, $Y = (Y^{(i)})_{i=1}^h$, $h > 1$, and $\tilde{B} = B$ or $\tilde{B} = B_q$.*

# References


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

[2] R.E. Cline, R.J. Plemmons, and G. Worm, Generalized inverses of certain Toeplitz matrices. *Linear Algebra and Its Applications,* **8**, 25–33, 1974.

[3] D. Coppersmith, S. Winograd, Matrix Multiplicaton via Arithmetic Progressions. *J. of Symbolic Computation*, **9 3**, 251–280, 1990.

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

[5] R. A. Demillo, R. J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Processing Letters*, **7**, **4**, 193–195, 1978.

[6] K. R. Davidson, S. J. Szarek, Local Operator Teory, 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.

[7] A. Edelman, Eigenvalues and Condition Numbers of Random Matrices, PhD Thesis (106 pages), Math Dept., MIT, 1989 and *SIAM J. on Matrix Analysis and Applications*, **9**, **4**, 543–560, 1988.

[8] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, Maryland, 1996.

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

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

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

[12] 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.

[13] W. L. Miranker, V. Y. Pan, Methods of Aggregations, *Linear Algebra and Its Applications*, **29**, 231–257, 1980.

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

[15] V. Y. Pan, D. Grady, B. Murphy, G. Qian, R. E. Rosholt, A. Ruslanov, Schur Aggregation for Linear Systems and Determinants, *Theoretical Computer Science*, *Special Issue on Symbolic–Numerical Algorithms* (D.A. Bini, V.Y. Pan, J. Verschelde editors), **409**, **2**, 255–268, 2008.

[16] V. Y. Pan, G. Qian, Randomized Preprocessing of Homogeneous Linear Systems, Tech. Report TR 2009014, *Ph.D. Program in Computer Science*, *Graduate Center, the City University of New York*, 2009. Available at http://www.cs.gc.cuny.edu/tr/techreport.php?id=352

[17] V. Y. Pan, G. Qian, A. Zheng, Randomized Preconditionng versus Pivoting, Tech. Report TR 2009010, *Ph.D. Program in Computer Science*, *Graduate Center, the City University of New York*, 2009. Available at http://www.cs.gc.cuny.edu/tr/techreport.php?id=352

[18] V. Y. Pan, X. Yan, Additive Preconditioning, Eigenspaces, and the Inverse Iteration, *Linear Algebra and Its Applications*, **430**, 186–203, 2009.

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

[20] 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.

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

[22] 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.

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

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