An Incremental Method for Computing Dominant Singular Spaces*

***Younes Chahlaoui*†, *Kyle A. Gallivan*‡, *and Paul Van Dooren*§**

1 Introduction

In many problems one needs to compute a projector on the dominant subspace of a given data matrix A of dimension $m \times n$. One can interpret the columns of the matrix A as *data vectors* with some *energy* equal to their 2-norm. Finding the dominant space of dimension $k < \min(m, n)$ amounts to finding the k first columns of the matrix U in the singular value decomposition of A:

$$A = U\Sigma V^T, \quad U^T U = I_n, \quad VV^T = V^T V = I_n, \quad \Sigma = \mathrm{diag}\{\sigma_1, \ldots \sigma_n\}, \qquad (1)$$

and where the diagonal elements σ_i of Σ are non negative and non increasing. This decomposition in fact expresses that the orthogonal transformation V applied to the columns of A yields a new matrix $AV = U\Sigma$ with orthogonal columns of non increasing norm. The *dominant* columns of this transformed matrix are obviously the k leading ones. A block version of this decomposition makes this more explicit :

$$A = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_{1,1} & \\ & \Sigma_{2,2} \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T, \qquad (2)$$

where U_1 and V_1 have k columns and $\Sigma_{1,1}$ is $k \times k$. An orthogonal basis for the corresponding space is then clearly given by U_1 which is also equal to $AV_1\Sigma_{1,1}^{-1}$. The

*This work was supported in part by the National Science Foundation through grants CCR-9796315 and ASC-9872140, and by the Belgian Programme on Inter-university Poles of Attraction.

†Department of Mathematical Engineering, Université catholique de Louvain, Belgium, E-mail: chahloui@csam.ucl.ac.be

‡School of Computational Science and Information Technology, The Florida State University, Tallahassee, FL, 32306, E-mail: gallivan@csit.fsu.edu.

§Department of Mathematical Engineering, Université catholique de Louvain, Belgium, E-mail: vdooren@csam.ucl.ac.be

cost of this decomposition including the construction of U is $14mn^2 + O(n^3)$. For an additional $O(n^3)$ operations it is also possible to compute an orthogonal basis for the columns of V_1, which is required in several applications.

In this paper we are interested in problems where m is very large, and $m >> n >> k$, and where column operations on A or on the basis U are not only costly in computations but also involves swapping data from the main memory. For such problems, computing the entire decomposition and then truncating to k basis vectors is unacceptable. We would like the complexity to be essentially linear in the size of the data or $O(mnk)$.

In addition, we assume that the matrix A is produced incrementally, i.e., all of the columns are not available simultaneously. Several applications have this property. For example, representing in this fashion a sequence of large images via the approximation of A where each column of A is an image is essentially the Karhunen-Loeve compression technique [6]. Such an approximation is also used in the context of observation-based model reduction for dynamical systems. The so-called **proper orthogonal decomposition** (POD) approximation uses the dominant left space of a matrix A where a column consists of a time instance of the solution of an evolution equation, e.g., the flow field from a fluid dynamics simulation. Since these flow fields tend to be very large, only a small number can be stored efficiently during the simulation and therefore an incremental approach is useful. As each time step is solved the basis for the space is updated to track the dominant left space. Each vector is used as a discrete approximation of a basis function of space only to approximate the state $x(t) \approx U_k a(t)$. The evolution equation $\dot{x} = F(x)$ is replaced by a reduced order equation $\dot{a} = U_k^H F(U_k a) = f(a)$. State information is recovered by integrating the reduced order equation rather than interpolating between saved states, thereby trading space for computation. The cost of the production of the reduced order system is dependent on the form of the differential equation which strongly influences the efficiency of applying the technique. Finally, the dominant space approximation is also used in text retrieval to encode document/term information, and avoid certain types of semantic noise. The incremental form is required when documents are added or when the entire matrix is not available at one point in time and space.

In this paper, we summarize an algorithm that yields an approximation to one or both of the dominant singular spaces by working incrementally on the columns of A. It requires $8mnk + O(nk^3)$ operations if only the left space is tracked and $10mnk + O(nk^3)$ if both spaces are tracked.

2 An incremental procedure

In this section, we summarize an incremental procedure to estimate the dominant subspaces of a given matrix A. The procedure processes a sliding window across the columns of the matrix. Each iteration consists of two steps. On the first step, we add l columns to the current window of k vectors into the matrix A. The second step deflates the $k + l$ vectors to k. At the end of each iteration, we have a factorization that yields bases of the left and right dominant singular spaces. In the description

of the algorithm that follows, we assume for simplicity $l = 1$.

To start the procedure, we assume that we have a QR factorization of the first k columns of A denoted $A(:, 1:k) = QR$ (using MATLAB notation). We initialize the right space basis to $V^T = I_k$. The vectors e_{i+1}^T and e_{k+1} are appended to expand the $k \times i$ matrix V^T by a row and column. The next column of A, denoted a, is used to expand Q and R via a Gram-Schmidt procedure:

$$\begin{aligned} r &= Q^T a \\ \hat{a} &= a - Qr \\ \rho &= \|\hat{a}\| \\ \hat{a} &= q\rho. \end{aligned}$$

This produces a new factorization

$$\hat{Q}\hat{R}\hat{V}^T. \tag{3}$$

The structure of the expand step is shown in Figure 1 for $l = 1$.

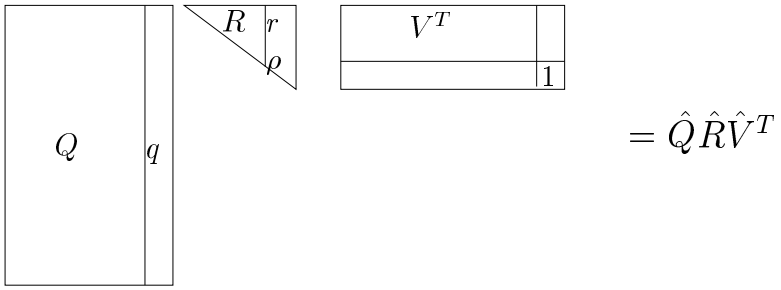

Figure 1. *The expand step for* $l = 1$.

The deflation step uses knowledge of the smallest singular value and the corresponding left singular vector of $\hat{R}$ in order to define transformations that place the factorization into a form that isolates the approximate bases for the dominant left and right singular spaces. We first compute the smallest singular value μ_{k+1} and corresponding singular vector u_{k+1} of the $(k+1) \times (k+1)$ matrix $\hat{R}$. An orthogonal matrix, G_u^T, constructed such that

$$G_u^T u_{k+1} = e_{k+1},$$

is applied to $\hat{R}$. Of course the matrix $G_u^T \hat{R}$ is not in triangular form, so we restore the triangular form by constructing an orthogonal matrix G_v such that

$$R_{up} = G_u^T \hat{R} G_v$$

is upper triangular. In order to isolate the dominant spaces, the matrices G_u and G_v must be used appropriately to deflate μ_{k+1} from R_{up}.

Given that R_{up} is upper triangular and $G_u^T u_{k+1} = e_{k+1}$ we have

$$\begin{aligned} e_{k+1}^T R_{up} e_{k+1} &= \mu_{k+1} \\ G_v^T v_{k+1} &= e_{k+1} \\ R_{up} e_{k+1} &= \mu_{k+1} e_{k+1}. \end{aligned}$$

It follows immediately from this that

$$R_{up} = \begin{bmatrix} R_+ & 0 \\ 0 & \mu_{k+1} \end{bmatrix} \tag{4}$$

where R_+ is upper triangular. Applying G_u and G_v in (3) yields

$$\begin{aligned} \hat{Q}\hat{R}\hat{V}^T &= (\hat{Q}G_u)(G_u^T \hat{R} G_v)(G_v^T \hat{V}^T) \\ &= (\hat{Q}G_u) R_{up} (G_v^T \hat{V}^T) \\ &= \bar{Q} R_{up} \bar{V}^T \end{aligned}$$

whose structure is shown in Figure 2. The column $\bar{q}$, row $\bar{v}^T$, and the last column and row in R_{up} are deleted to yield Q_+, R_+, and V_+^T, which are $m \times k$, $k \times k$, and $k \times i$ respectively after the i-th step. These are used as the factorization that is expanded by the $i+1$-st column of A during the next iteration.

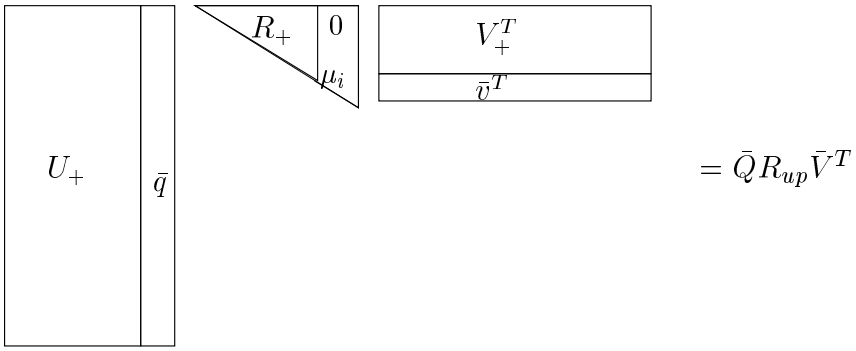

Figure 2. *The deflate step for $l = 1$.*

All columns of A are *passed* through once and *compared* with the current best estimate of the dominant subspace before deflating to maintain the k vectors in each basis. The derivations of the algorithms are very empirical but, in fact, good bounds can be obtained for the quality of the bases. These are summarized in Section 4.

3 Computational cost

The one-sided algorithm that produces a basis for the left dominant singular subspace (Q_+) and an estimate of the singular values (from R_+) has a complexity

of $8mnk + O(nk^3)$. The Gram-Schmidt expansion of Q requires approximately $4mk$ per iteration. In the deflate step, the computation of $\hat{Q}G_u$ also requires approximately $4mk$ operations if G_u is formed from Householder transformations or modified Givens rotations. The construction of G_v via a QR factorization, the computation of u_{k+1} and μ_{k+1}, and the computation of $G_u^T \hat{R} G_v$ together require $O(k^3)$ operations.

The two-sided algorithm that tracks Q_+,R_+, and V_+ requires more careful consideration to achieve our $O(mnk)$ target. As before the Gram-Schmidt portion contributes $4mk$ per iteration. If the QR factorization-based formation of G_v is used then the computation of $G_v^T \hat{V}^T$ requires $O(ik^2)$ per iteration resulting in an overall complexity term of $O(n^2k^2)$ which is unacceptable. However, using Givens rotations implies that, given u_{k+1} and μ_{k+1}, G_u and G_v can be determined and $G_u^T \hat{R} G_v$ computed in $O(k^2)$ operations. The matrices G_u and G_v are computed and applied simultaneously to a matrix, T, formed by appending u_{k+1} to $\hat{R}$ as the final column:

$$G_u^T T G_v = P_k P_{k-1} \cdots P_1 T Z_1 \cdots Z_{k-1} Z_k$$

where P_j is a rotation of rows j and $j+1$ and Z_j is a rotation of columns j and $j+1$. The nonzero elimination and fill pattern is shown in Figure 3 for $k = 3$. ρ is used to mark the positions of original elements of $\hat{R}$ that are updated, η_j is used to mark the element of u_{k+1} that is eliminated by P_j (with the exception of η_4), and ϕ_j is used to denote the fill-in caused by P_j and eliminated by Z_j.

$$P_3 P_2 P_1 \begin{pmatrix} \rho & \rho & \rho & \rho & \eta_1 \\ \phi_1 & \rho & \rho & \rho & \eta_2 \\ 0 & \phi_2 & \rho & \rho & \eta_3 \\ 0 & 0 & \phi_3 & \rho & \eta_4 \end{pmatrix} Z_1 Z_2 Z_3 = \begin{pmatrix} \rho & \rho & \rho & 0 & 0 \\ 0 & \rho & \rho & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & \mu_4 & 1 \end{pmatrix}$$

Figure 3. *The elimination and fill-in structure for the two-sided algorithm with* $k = 3$.

The computation of $\hat{Q}G_u$ requires the application of k Givens rotations and $6km$ operations and the computation of $G_v^T \hat{V}^T$ requires the application of k Givens and $6ki$ operations on the i-th iteration. Including the $O(k^3)$ on each step to find u_{k+1} and μ_{k+1}, the total operation count is $10mnk + 3kn^2 + O(nk^3)$. This is the most efficient algorithm for tracking dominant singular spaces of which we are aware. The closest approach is that of Chandrasekaran et al. [2].

4 Accuracy and orthogonality

In [1], the effect of truncating the SVD on the spaces and singular values as well as the consequences of finite precision are analyzed in detail. The results of those analyses are summarized in this section.

The analysis of the effect of the truncation on each iteration starts with the observation that there exists an orthogonal column transformation G_n that relates

A and the intermediate results of the algorithm :

$$AG_n = \left[\begin{array}{cccc} Q_{(n)}R_{(n)} & \nu_{(k+1)}z_{k+1} & \cdots & \nu_{(n)}z_n \end{array} \right]. \tag{5}$$

G_n consists of the product of the G_v matrices from each iteration and appropriately chosen permutations. $Q_{(n)}$ and $R_{(n)}$ are the basis for the left dominant space and the $k \times k$ triangular matrix whose singular values are used as approximations to the dominant singular values of A. The $\nu_{(j)}$ and z_j are the value μ_{k+1} and vector $\bar{q}$ dropped on the j-th iteration.

Using the singular value decomposition of $R_{(n)}$:

$$R_{(n)} = \hat{U}_n \hat{\Sigma} \hat{V}_n^T$$

we construct:

$$AG_n \left[\begin{array}{cc} \hat{V}_n & 0 \\ 0 & I \end{array} \right] = \left[\begin{array}{cc} Q_{(n)}\hat{U}_n & Q_{(n)}^{\perp} \end{array} \right] \left[\begin{array}{cc} \hat{\Sigma} & A_{1,2} \\ 0 & A_{2,2} \end{array} \right] = WM,$$

where the columns of $A_2 \doteq \left[\begin{array}{c} A_{1,2} \\ A_{2,2} \end{array} \right]$ have 2-norms $\nu_{(i)}$ and the Frobenius norm of this submatrix is $\| \left[\nu_{(k+1)}, \ldots, \nu_{(n)} \right] \|_2$.

The singular values of A are also those of

$$M \doteq \left[\begin{array}{cc} \hat{\Sigma} & A_{1,2} \\ 0 & A_{2,2} \end{array} \right].$$

Bases of the true left dominant subspace and the approximated subspace (that is generated by setting $A_2 = 0$ in M) can be normalized so the subspaces are $\mathrm{Im} \left[\begin{array}{c} I_k \\ P_u \end{array} \right]$ for the true subspace and $\mathrm{Im} \left[\begin{array}{c} I_k \\ 0 \end{array} \right]$ for the approximation. The largest canonical angle θ_k between the subspaces satisfies $\tan \theta_k = \|P_u\|$ [7]. We therefore want $\|P_u\| \approx 0$. A similar construction is possible for the right space and using results from [4] the following is shown in [1].

Theorem 2. *Let $\hat{\mu} = \max_i \nu_{(i)}$, σ_i be the singular values of A, $\hat{\sigma}_i^{(n)}$ be the singular values of $R_{(n)}$,*

$$\hat{M} = \left[\begin{array}{cc} \hat{\Sigma} & 0 \\ 0 & 0 \end{array} \right], \quad M = \left[\begin{array}{cc} \hat{\Sigma} & A_{1,2} \\ 0 & A_{2,2} \end{array} \right], \quad \mu \doteq \| \left[\begin{array}{c} A_{1,2} \\ A_{2,2} \end{array} \right] \|_2.$$

The left and right canonical angles θ_k and ϕ_k and the singular values satisfy the following bounds.

$$|\sigma_i - \hat{\sigma}_i^{(n)}| \approx \frac{\hat{\mu}^2}{(\sigma_i + \hat{\sigma}_i^{(n)})} \leq \frac{\hat{\mu}^2}{2\hat{\sigma}_i^{(n)}}.$$

If $\mu < \hat{\sigma}_k^{(n)}/\sqrt{3}$ then

$$\tan \theta_k \leq \tan \hat{\theta}_k = \frac{\hat{\mu}^2}{(\hat{\sigma}_k^{(n)})^2 - \hat{\mu}^2},$$

and if $\mu < 7(\hat{\sigma}_k^{(n)})^2/16\|A\|_2$ *then*

$$\tan\phi_k \leq \tan\hat{\phi}_k = \frac{\hat{\mu}\hat{\sigma}_1^{(n)}}{(\hat{\sigma}_k^{(n)})^2 - \hat{\mu}^2}.$$

Note that the estimates are all quadratic in $\hat{\mu}$ and should give quite accurate results if $\hat{\mu} << \hat{\sigma}_i^{(n)}$. This is the case when the gap, γ, between σ_k and σ_{k+1} is large.

We therefore have that in exact arithmetic the algorithm can track well the dominant spaces given a sufficient gap in the spectrum. The algorithm, however, has the flavor of classical Gram-Schmidt and therefore raises concerns about the numerical reliability of the algorithm. We have shown that the concerns are unfounded [1]. A backward error analysis yields the following theorem.

Theorem 3. *The incremental algorithm produces approximate matrices* $\bar{V}_{(i)}, \bar{Q}_{(i)}$ *and* $\bar{R}_{(i)}$ *that satisfy exactly the perturbed equation*

$$[A(:,1\ :i) + E]\bar{V}_{(i)} = \bar{Q}_{(i)}\bar{R}_{(i)}, \quad (\bar{V}_{(i)} + F)^T(\bar{V}_{(i)} + F) = I_k,$$

with the bounds (up to $O(\epsilon_{unit}^2)$ *terms where* ϵ_{unit} *is the unit roundoff) :*

$$\|E\|_F \leq \epsilon_e\|A\|_2, \ \ \epsilon_e \leq 26k^{3/2}n\epsilon_{unit}, \quad \|F\|_F \leq \epsilon_f \leq 9k^{3/2}n\epsilon_{unit}.$$

and in practice

$$\epsilon_e \leq 26k^2\epsilon_{unit}, \quad \epsilon_f \leq 9k^2\epsilon_{unit}.$$

Note these bounds do not depend on m, largest dimension of A, and in practice they do not depend strongly on n and therefore the result scales well for the large problems encountered in the applications discussed above.

Given the backward error result, we have also considered the implications for the loss of orthogonality in the bases. If the loss is proportional to the condition number of A, the results could be disastrous. If, however, the loss is proportional to the condition number of A restricted to the dominant space, the algorithms are satisfactory in practice. This is in fact the case, and using results from [3] and [5] the following theorem can be proven.

Theorem 4. *Let (a given matrix)* $\bar{V} \in \mathcal{R}^{n\times k}$ *select* k *columns of the matrix* $A \in \mathcal{R}^{m\times n}$*, and let*

$$A\bar{V} = QR, \quad Q^TQ = I_k,$$

with R *upper triangular, be its exact* QR *factorization. Let*

$$A\bar{V} + G = \bar{Q}\bar{R}, \quad \|G\|_F = \epsilon_g\|A\|_2 \approx u\|A\|_2, \tag{6}$$

be a computed version, where $\bar{Q} = Q + \Delta_Q$, $\bar{R} = R + \Delta_R$. *Then under a mild assumption, we can bound the loss of orthogonality in* $\bar{Q}$ *as follows:*

$$\|\bar{Q}^T\bar{Q} - I_k\|_F \leq \sqrt{2}\epsilon_g\kappa_2(R)\kappa_R(A\bar{V}) \leq 2\epsilon_g\kappa_2^2(R), \quad \epsilon_g \approx u.$$

$\kappa_R(M)$ refers to the refined condition number of [3]. The implication is that there is no disastrous loss of orthogonality and this is consistent with our empirical observations [1].

5 Example

We have tested the algorithms with random matrices of dimension $m = 1000$ and $n = 50$. The matrices were normalized so that the singular values were all between 0 and 1. The number of dominant singular values and vectors was taken to be $k = 5$. We illustrate the potential of the incremental algorithm with a single experiment with a fairly large gap of $\gamma = 0.7448$. The complete set of experiments is given in [1]. In Figure 4, we plot the true singular values σ_i as a solid line, the approximations, $\sigma_i^{(n)}$, of the k leading singular values as stars, and the dismissed singular values $\nu_{(i)}$ as circles. Details of the predicted and actual singular values and canonical angles are presented in Tables 1 and 2.

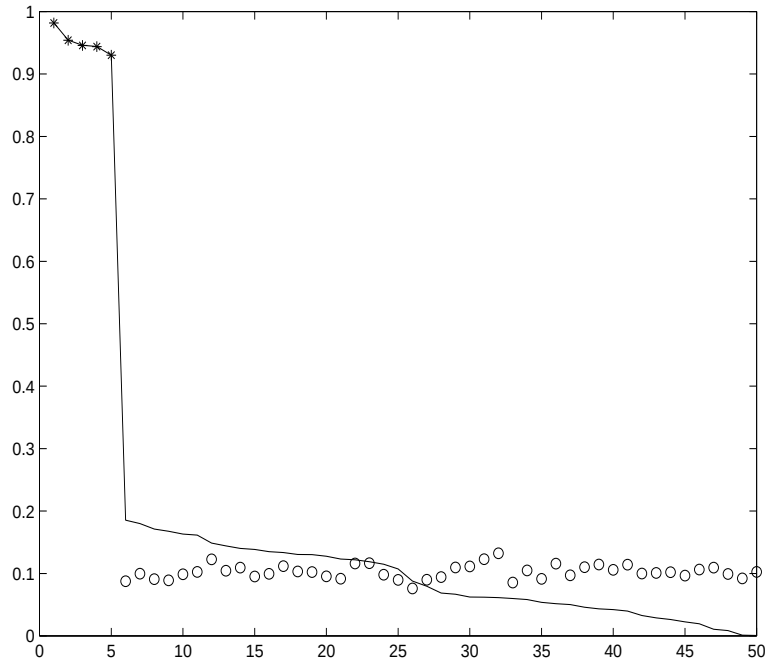

Figure 4. —- *true sv's* $\sigma_i(A)$, $*$ *approximated sv's* $\hat{\sigma}_1^{(n)}, \ldots, \hat{\sigma}_k^{(n)}$,$\circ$ *dismissed sv's* $\nu_{(k+1)}, \ldots, \nu_{(n)}$.

σ_i	$\hat{\sigma}_i^{(n)}$	$\lvert\sigma_i - \hat{\sigma}_i\rvert$	$\hat{\mu}^2/(2\hat{\sigma}_i^{(n)})$	$\cos\theta_i$	$\cos\phi_i$
0.9820	0.9817	0.0003	0.0091	1.0000	0.9999
0.9544	0.9541	0.0003	0.0096	1.0000	0.9999
0.9461	0.9458	0.0003	0.0098	1.0000	0.9999
0.9442	0.9440	0.0003	0.0098	1.0000	0.9996
0.9302	0.9301	0.0002	0.0101	1.0000	0.9992

Table 1. *Errors in singular values, predicted bound, and true canonical cosines.*

$\mu = 0.1857$	$\hat{\mu} = 0.1323$
$\|P_u\|_2 = 0.0047$	$\|\hat{P}_u\|_2 = 0.0413$
$\|P_v\|_2 = 0.0396$	$\|\hat{P}_v\|_2 = 0.3066$
$\cos\hat{\theta}_k = 0.9991$	$\cos\hat{\phi}_k = 0.9561$

Table 2. *Predicted cosines, and true and predicted tangents, true and predicted first dismissed singular value.*

6 Conclusions

In this paper we have summarized our recent work on the design and analysis of incremental algorithms for determining dominant singular spaces. The proposed one- and two-sided algorithms are efficient and effective. They are reasonably robust with acceptable loss of orthogonality and accuracy related to the gap in the spectrum. The algorithms seem to have potential for large problems due to the practical error bounds that are essentially independent of m and n given certain simplifying assumptions.

Bibliography

[1] Y. Chahlaoui, K. A. Gallivan, and P. Van Dooren, *Recursive Calculation of Dominant Singular Subspaces*, submitted for publication.

[2] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler and H. Zhang, *An Eigenspace Update Algorithm for Image Analysis*, submitted for publication.

[3] X.-W. Chang, C. Paige and G. Stewart, *Perturbation Analyses for the QR Factorization*, SIAM J. Matr. Anal. Appl., 18 (1997), pp. 775–791.

[4] S. Geman, *A Limit Theorem for the Norm of Random Matrices*, Annals of Probability, 8 (1908), pp. 252–261.

[5] N. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM Publications, Philadelphia, 1996.

[6] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.

[7] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, 1990.