UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Analysis and Applications of Stewart's Pivoted QLP Decomposition (Ph.D. Thesis)

David Andrew Huckaby

May 2002

CAM Report 02-30

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

UNIVERSITY OF CALIFORNIA

Los Angeles

Analysis and Applications of Stewart's Pivoted QLP Decomposition

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

David Andrew Huckaby

2002

The dissertation of David Andrew Huckaby is approved.

Christopher Anderson

Ming Gu

Kung Yao

Tony F. Chan, Committee Chair

University of California, Los Angeles

2002

To my savior and love,

the Lord Jesus Christ ...

in whom, through whom, and for whom are all things.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Professional: Thank you to Dr. Chan for expert advice and skillful guidance. He is essentially a coauthor. Thanks also to Ming and to Per Christian Hansen for helpful discussions.

Personal: I could never have graduated without the love and support of my family, especially my parents. Thank you. Thanks again to Dr. Chan for his patience throughout my illness. Thanks to the many others who were supportive as well, including many brothers and sisters in the Lord at UPC and also Christian brother David Kan in the Math Department.

πάντα ἰσχύω ἐν τῳ ἐνδυναμοῦντί με.

VITA

Nov. 19, 1972	Born, Fort Worth, Texas, USA.
1995	B.S. (Mathematics), The University of Texas at Austin.
1997	M.A. (Mathematics), UCLA, Los Angeles, California.
1995–2002	Teaching Assistant and Research Assistant, Mathematics Department, UCLA.

ABSTRACT OF THE DISSERTATION

Analysis and Applications of Stewart's Pivoted QLP Decomposition

by

David Andrew Huckaby

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2002

Professor Tony F. Chan, Chair

The pivoted QLP decomposition, introduced by G. W. Stewart, represents the first two steps in an algorithm which approximates the SVD. The matrix $A\Pi_0$ is first factored as $A\Pi_0 = QR$, and then the matrix $R^T\Pi_1$ is factored as $R^T\Pi_1 = PL^T$, resulting in $A = Q\Pi_1 L P^T \Pi_0^T$, with Q and P orthogonal, L lower-triangular, and Π_0 and Π_1 permutation matrices. Stewart noted that the diagonal elements of L approximate the singular values of A with surprising accuracy. We provide mathematical justification for this phenomenon. We demonstrate that if there is a gap in the singular values, the relative error in all of the singular values is quadratic in the gap ratio. The worst case is at the gap, where the absolute error is cubic in the singular values on either side of the gap. One order is due to the rank-revealing pivoting in the first step; then, two more orders are achieved in the second step. Our analysis assumes that $\Pi_1 = I$, that is, that pivoting need be done only on the first step. The algorithm can be continued beyond the first two steps, and we make some observations concerning the asymptotic convergence. For example, we point out that repeated singular values can accelerate convergence of individual elements. This, in addition to the relative convergence

to all of the singular values being quadratic in the gap ratio, indicates that the QLP decomposition can be powerful even when the ratios between neighboring singular values are close to one.

We also follow Stewart in considering truncating and interleaving the algorithm, which are possible because of the nonnecessity of pivoting in the second step, with an eye toward obtaining an efficient approximation to the truncated SVD for low-rank problems. We show that the convergence results of the full decomposition hold also in this truncated version. For a matrix having numerical rank r, the truncated pivoted QLP decomposition can be computed in $\mathcal{O}(mnr)$ time, making it ideal for accurate SVD approximations for low-rank problems.

We also look at two applications in which the pivoted QLP decomposition can substitute for the SVD, namely condition number estimation and latent semantic indexing.

CHAPTER 1

Introduction

1.1 The Singular Value Decomposition

One of the most important tools for analyzing a matrix A is the Singular Value Decomposition (SVD): If A is a real m-by-n matrix, then there exist orthogonal matrices

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m \times m} \text{ and } V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$U^T A V = diag(\sigma_1, \ldots, \sigma_p) \equiv \Sigma \in \mathbb{R}^{m \times n} \; p = \min\{m, n\}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ [GL96, Theorem 2.5.2].

The SVD shows us that any matrix $A = U\Sigma V^T$ is equivalent to a diagonal matrix, when supplied with the correct change of basis on the left and the correct change of basis on the right. The vectors u_i and v_i are called, respectively, the ith *left singular vector* and ith *right singular vector* of A. The σ_i are called the *singular values* of A.

Since the rank of a diagonal matrix is the number of its nonzero entries, we see that the rank of Σ is the number r of nonzero singular values. Hence, the matrix A can be written

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

which expresses A as a sum of r rank-one matrices. Since U and V are orthogonal, the rank of A is thus r. A basis for the range of A is given by $[u_1, \ldots, u_r]$, and a basis for the nullspace of A is given by $[v_{r+1}, \ldots, v_n]$. Note that since U and V are orthogonal, we have $\|A\|_2 = \|\Sigma\| = \sigma_1$.

That the rank of a matrix A can be obtained by inspecting the singular values is definitely a selling point for the SVD. However, the decomposition says much more about A, and this can be seen most easily by viewing things geometrically. The image of the unit sphere under any matrix A is a hyperellipse. The left singular vectors u_i give the direction of the axes of this hyperellipse, and the σ_i give their corresponding lengths. The right singular vectors v_i are the vectors in the original unit sphere that map to the u_i. So the SVD tells us in which directions A is "active" (from the singular vectors) and how much (from the singular values). This is made precise by the following result [GL96, Theorem 2.5.3]: Let $k < r = \text{rank}(A)$, and let

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T.$$

Then

$$\min_{\text{rank}(B)\leq k} = \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}. \tag{1.1}$$

So A_k is the best rank-k approximation to A, and the error is σ_{k+1}.

This result really brings to the fore the true power of the SVD, that the decomposition tells us so much about the dimensionality of the problem represented by A. First of all, as noted above, it provides the rank. By equation (1.1), if $\sigma_{k+1} = 0$ and $\sigma_k \neq 0$, then the rank of A is k. Similarly, if σ_{k+1} is numerically zero and σ_k is not, then A has numerical rank k. But we can generalize: If there is a large gap in the singular values, say σ_k is much larger than σ_{k+1}, we see that $A_{k+1} = A_k + \sigma_{k+1} u_{k+1} v_{k+1}^T$ is not a much better approximation to A than is A_k, relatively speaking. Hence, gaps in the singular values allow us to break up A into subspaces, in each of which A is "acting" roughly the same amount. Geometrically, the kth axis of the hyperellipse to which A maps the unit sphere is so much larger than the $(k+1)$st axis that taking the first k axes provides a good approximation to the entire hyperellipse. Practically, we can "chop off" the SVD at this point, letting $A = A_k$, with relative impunity. The next logical place to reduce the dimension would be at the next gap, say between σ_l and σ_{l+1}. For $k+1 \leq i \leq l$, the input directions v_i and output directions u_i are far less important than for $i \leq k$ but far more important than for $i \geq l+1$

While the SVD gives a lot of information about A, it requires a lot of computation time. What we would like to have is a good approximation to the SVD which is cheaper to compute.

1.2 Prelude in QR

Let us survey some matrix factorizations which were around before the pivoted QLP decomposition. Each gives a portion of the information that the SVD provides, and we will look at them in order of increasing informativeness.

One attempt at gaining SVD-type information is, of course, the QR factorization [GL96, Section 5.2]: Let $A \in \mathbb{R}^{m \times n}$ have rank n. Then A can be written uniquely in the form $A = QR$, where Q is an $m \times n$ matrix having orthonormal columns and R is an $n \times n$ upper triangular matrix with positive diagonal elements. This can be made into a full blown decomposition which handles rank-deficient A. For details, see [Ste98, Chap. 4, Sec. 1].

The QR factorization is very cheap to compute relative to the SVD. Since Q is orthogonal, we know that $\|A\|_2 = \|R\|_2$, so that the singular values of R are the same as those of A. As we have effectively reduced a dense matrix A to an upper triangular matrix R, we might hope that the diagonal elements of R are approximations to the singular values of A.

This is clearly not true in general. For example, if the norm of the first column of A is small compared to the norms of the rest of the columns, then r_{11} is small and comes nowhere close to $\sigma_1 = \|A\|_2$. All that is readily apparent is that $r_{11} \leq \sigma_1$ and $r_{nn} \geq \sigma_n$.

We can introduce column pivoting into the computation of the QR factorization [Gol65], and this makes a huge difference. Pivoting ensures that the diagonal elements in the computed R-factor are in sorted order, and these are often rough approximations of the corresponding singular values. They can also be used for gap revelation, which of course requires less accuracy. Pivoting also provides at least one bound that its unpivoted brother could not promise. The fact that r_{11}

equals the norm of the largest column of A gives rise to the bound $r_{11} \geq \sigma_1/\sqrt{n}$ (see page 19).

Note that although the column pivoted QR factorization provides this useful bound for the error in approximating σ_1 by r_{11}, only an exponential bound exists for σ_n and r_{nn} [Dem97]. So although in practice the diagonal elements of R are rough approximations to the singular values, this need not always be the case. (Indeed, the Kahan matrix provides a well-known counterexample. [Kah66])

This is one consideration that motivates the *rank revealing QR factorization* (RRQR) [Fos86, Cha87]. If R is partitioned as

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

an RRQR algorithm tries to maximize the smallest singular value of R_{11} and/or minimize the largest singular value of R_{22} [CI94]. This essentially means making, respectively, $\|R_{11}\|$ as large as possible and $\|R_{22}\|$ as small as possible. From the interlacing property of singular values, $\sigma_{min}(R_{11}) \leq \sigma_k(A)$ and $\sigma_{max}(R_{22}) \geq \sigma_{k+1}(A)$. So an RRQR factorization provides bounds on the singular values of A in terms of the norms of the blocks.

In terms of tracking the singular values, the RRQR algorithms tend to perform about as well as the pivoted QR factorizations, and they come with guarantees. Whereas the pivoted QR factorization can completely fail (like on Kahan's example), an RRQR algorithm is guaranteed to work within the bounds it provides. So for example, one algorithm which tries to minimize $\|R_{22}\|$ can promise that $r_{nn} \leq \sqrt{n}\,\sigma_n$, a bound that the column pivoted QR factorization cannot provide.

Stewart [Ste99] introduced another candidate for SVD-quality information with minimal cost having the QR factorization as its only building block. The pivoted QLP decomposition requires only the work of two QR factorizations, and one of them need not even be pivoted. Yet despite its simplicity and speed, the decomposition provides approximations to the singular subspaces of A and gives excellent approximations to *all* of the singular values of A. In this dissertation we will study this fascinating decomposition.

1.3 Overview

The dissertation is organized as follows. To start off Chaper 2 we discuss the QLP decomposition and illustrate just how good it is at tracking the singular values. We also consider taking it beyond just two successive QR factorizations, which leads to an algorithm that asymptotically calculates the SVD. In discussing the convergence of this algorithm in the absence of pivoting, we point out the connection to the QR algorithm and mention past work done by Mathias and Stewart [MS93] and Chandrasekaran and Ispsen [CI95].

In § 2.2 in Chapter 2 we study the convergence of the QLP decomposition. The decomposition is obtained by performing a pivoted QR factorization of A as $A = QR$, where we have incorporated the pivoting into the matrix A, and then a pivoted QR factorization of R^T as $R^T = PL$. If there is a gap between σ_k and σ_{k+1}, partition the matrix L into diagonal blocks L_{11} and L_{22} and off-diagonal block L_{21}, where L_{11} is k-by-k. We show that the convergence of $(\sigma_j(L_{11})^{-1} - \sigma_j^{-1})/\sigma_j^{-1}$ for $j = 1, \ldots, k$, and of $(\sigma_j(L_{22}) - \sigma_{k+j})/\sigma_{k+j}$, for $j = 1, \ldots, n-k$ are all quadratic in the gap ratio σ_{k+1}/σ_k. The worst case is therefore at the gap, where the absolute errors $\|L_{11}^{-1}\| - \sigma_k^{-1}$ and $\|L_{22}\| - \sigma_{k+1}$ are thus cubic in σ_k^{-1} and σ_{k+1}, respectively. It turns out that one order is due to the rank-

revealing pivoting in the first step; then, the pivoting has provided a springboard so that two more orders are achieved in the second step. Our analysis assumes that $\Pi_1 = I$, that is, that pivoting is done only on the first step. That we have relative convergence quadratic in the gap ratio for *all* of the singular values helps show why the QLP decomposition can work so well even when most of the ratios between neighboring singular values are close to one. It only takes one significant gap for the decomposition to work very well. We provide numerical experiments to illustrate the results.

In § 2.3 in Chapter 2 we present some observations concerning the asymptotic convergence of the algorithm. The off-diagonal elements converge in interesting patterns, and from them we can discern the asymptotic rates of convergence of the diagonal elements. From this perspective of individual element convergence, we again see the phenomenon of repeated singular values (or singular values that are close) accelerating convergence. The results are obtained by analyzing the actions of the Givens rotations used in computing the QR factorizations, and numerical experiments are again provided to illustrate the theory.

In Chapter 3 we look at truncating and interleaving the QLP decomposition, which gives a result analagous to the truncated SVD and is ideal for low-rank problems. Truncating is computing only that part of the decomposition pertaining to the larger singular values, including approximations to these singular values and their respective singular vectors. Interleaving is a tool used to identify a gap in the singular values, presumably in order to truncate at the gap. As the computation proceeds down the diagonal of A, we alterate between steps one and two, i.e., between computing rows of R and columns of L, using the diagonal elements of R to probe for a gap and the diagonal elements of L to confirm an alleged gap. When a gap is confirmed, the computation can stop, for at this point

the truncated pivoted QLP decomposition has been computed.

Stewart suggested truncating and interleaving, and we show that they actually work. We extend our theory for the full QLP decomposition to the truncated version. In particular, we show that if there is a gap between σ_k and σ_{k+1}, again the convergence of $(\sigma_j(L_{11})^{-1} - \sigma_j^{-1})/\sigma_j^{-1}$ for $j = 1, \ldots, k$, are all quadratic in the gap ratio σ_{k+1}/σ_k—this without having to compute the rest of the matrix L. We also point out that the truncated QLP runs in $\mathcal{O}(mnk)$ time, where truncation occurs after k rows.

We then consider two applications of the pivoted QLP decomposition in situations where it can substitute for the SVD. In Chapter 4 we look at the problem of estimating condition number. We first discuss the idea of the condition of a problem, illustrating the need for a condition number estimate. We then derive bounds on the underapproximation ratio. A condition estimator should run in $\mathcal{O}(n^2)$ time. So using ideas related to QLP as inspiration, we discuss techniques that provide estimates better than that of the pivoted QR factorization but run in $\mathcal{O}(n^2)$ or even $\mathcal{O}(n)$ time. We finish the chapter with some numerical experiments that verify the ability of the QLP and some of these other methods to provide good condition number estimates.

In Chapter 5 we use the QLP to do latent semantic indexing. Following the exposition of Barry, Drmač, and Jessup in [BDJ99], we first discuss the idea of representing a database as a matrix and performing queries. We then discuss the idea of reducing the rank of the matrix by using the QR factorization or the SVD and point out that the rank-reduced representation of the database reveals its semantic content better than the original full-rank matrix. We then show how the truncated and interleaved QLP decomposition is ideal for this application and apply it to a small example. A concluding chapter provides a summary of

the dissertation.

CHAPTER 2

Convergence Theory

2.1 QLP: An Approximate SVD

2.1.1 The Pivoted QLP Decomposition

The QLP decomposition was introduced by G. W. Stewart [Ste99], who observed its potency in rank revelation, singular value approximation, and gap revelation. Let us call the diagonal elements in the R matrix of a QR factorization of A the R-values of A. Noting that the R-values are rough approximations of the singular values, Stewart suggested taking the pivoted QR factorization and then triangularizing on the right, obtaining the factorization $A = Q\Pi_1 L P^T \Pi_0^T$. If we include the permutation matrices Π_1 and Π_0^T as part of A and R^T, we have $A = QLP^T$, called the pivoted QLP decomposition of A. Note that the second step is equivalent to performing a QR factorization on R^T, obtaining $R^T = PL^T$. Also note that L is lower-triangular. So the decomposition amounts to taking two pivoted QR factorizations and thus factorizing A into the product of an orthogonal matrix, a lower-triangular matrix, and another orthogonal matrix. The diagonal elements of L are called the L-values of A.

Stewart showed empirically that the L-values track the singular values surprisingly well—far better than the R-values. See Figure 2.1.

Note from Figure 2.1 that not only do the L-values identify the gap far better

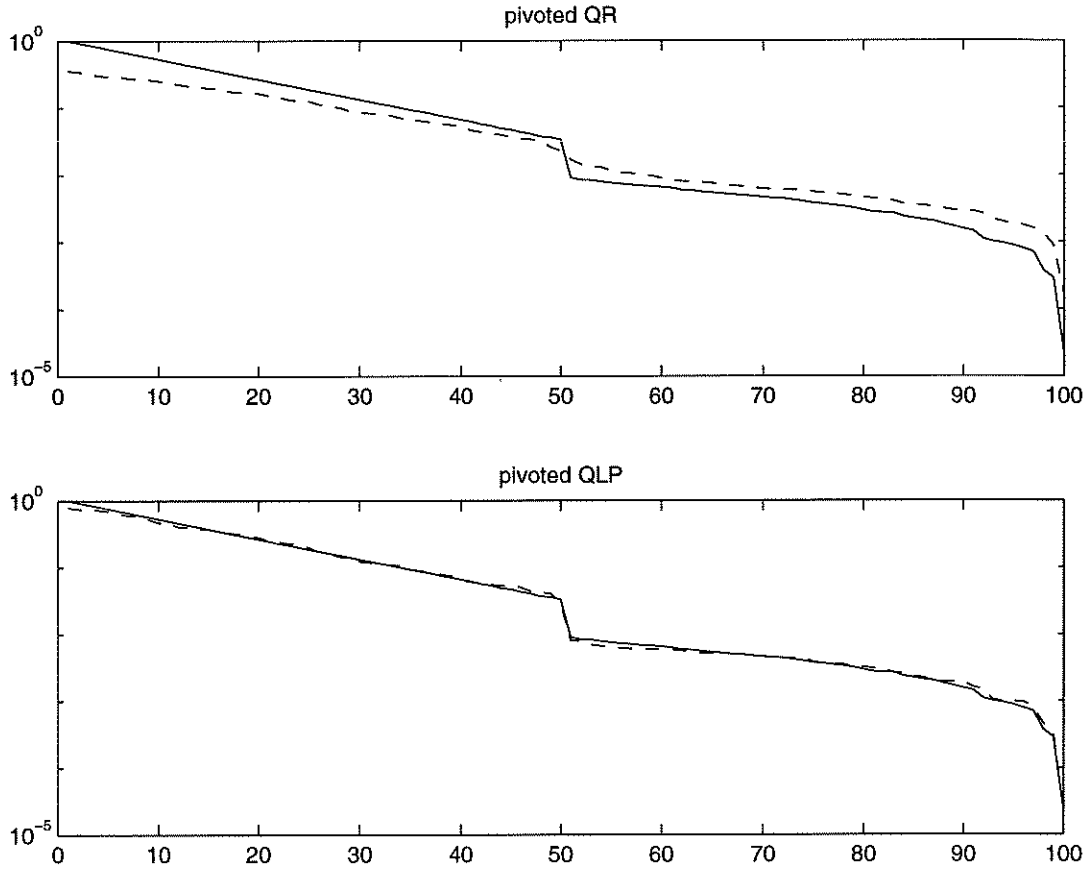

Figure 2.1: QR vs QLP. Here the solid lines are the singular values of a 100-by-100 matrix that has a gap between σ_{50} and σ_{51}. The dotted line represents the R-values in the first plot and the L-values in the second.

than the R-values, they also approximate the singular values. So with only the extra cost of one more QR factorization, we get very good information—almost SVD-quality information in many situations. (Note that the QLP decomposition is a special case of the ULV decomposition, also introduced by Stewart [Ste92, Ste93]. The usual ULV and URV decompositions are also rank-revealing but do not attempt to approximate the singular values. Their main selling point is that they are easily updated.)

2.1.2 The QLP Iteration

Stewart points out that without pivoting, the decomposition represents the first two steps in an iterative algorithm that actually computes the SVD [MS93]. Let us call this iterative algorithm the *QLP iteration.* In each step after the first, we just compute the QR factorization of the transpose of the R factor produced by the last step. Here is the algorithm:

1. Compute the QR factorization of A, obtaining $A = Q_0 R_0$.
2. Compute the QR factorization of R_0^T, obtaining $R_0^T = Q_1 R_1$.
3. Compute the QR factorization of R_1^T, obtaining $R_1^T = Q_2 R_2$.
4. Continue in the same way.

Notice that if we stop after the second step and perform the QR factorizations in steps 1 and 2 with pivoting, then this is just the QLP decomposition. If on the other hand we do not pivot and do not stop after the second step but continue on, we obtain something akin to the QR algorithm for computing eigenvalues and eigenvectors, which follows [TB97]:

$A^{(0)} = A$

for $k = 1, 2, \ldots$

$Q^{(k)} R^{(k)} = A^{(k-1)}$ QR factorization of $A^{(k-1)}$

$A^{(k)} = R^{(k)} Q^{(k)}$ Switch factors

The $A^{(k)}$ converge to a diagonal matrix whose elements are the eigenvalues of A in decreasing order.

To see how the QLP iteration relates to the QR algorithm, define the matrix $A^{(i)} = R_{2i}^T R_{2i}$. Now from the QLP iteration, $R_{2i}^T = Q_{2i+1} R_{2i+1}$, so that $A^{(i)} =$

$Q_{2i+1}R_{2i+1}R_{2i}$. Since the product of two upper-triangular matrices is upper-triangular, this equation represents a QR factorization of $A^{(i)}$, the first step of the QR algorithm. To perform the second step, we switch the two factors [MS93]:

$$\begin{aligned} A^{(i+1)} &= R_{2i+1}R_{2i}Q_{2i+1} \\ &= R_{2i+1}R_{2i+1}^T \\ &= R_{2i+1}Q_{2i+2}^T Q_{2i+2}R_{2i+1}^T \\ &= R_{2i+2}^T R_{2i+2}. \end{aligned}$$

We see that every two steps of the QLP iteration (excluding the first) on the R_i are equivalent to one step of the QR algorithm on the $R_{2i}^T R_{2i}$. Since the $A^{(i)}$ converge to a diagonal matrix whose elements are the eigenvalues of $A^{(0)} = R_0^T R_0$ in decreasing order, the R_{2i} converge to a diagonal matrix whose elements are the singular values of R_0 in decreasing order. (A similar argument can be used for the R_i when i is odd.)

Some convergence results for the unpivoted version of the iteration were given by Mathias and Stewart [MS93] and Chandrasekaran and Ipsen [CI95]. Let R be the upper-triangular matrix at one step of the iteration and let R' be the upper-triangular matrix at the next step. Partition the n-by-n matrices as

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \quad \text{and} \quad R' = \begin{pmatrix} R'_{11} & R'_{12} \\ 0 & R'_{22} \end{pmatrix},$$

where R_{11} and R'_{11} are k-by-k. Mathias and Stewart showed that if $\rho = \|R_{22}\|/\inf(R_{11}) < 1$, then $\|R'_{12}\| \leq \rho\|R_{12}\|$. Chandrasekaran and Ipsen

studied not only the convergence of the R matrices, but also the convergence of the singular vectors of the R matrices as well. They provided some monotonic convergence results and some asymptotic convergence results and also suggested preceeding the iteration with an RRQR algorithm to make ρ as small as possible.

We would like to point out that the QLP iteration also computes the singular vectors. At each iteration, an orthogonal matrix is computed as part of the QR factorization. A product of these accumulates on each side of the $R^{(i)}$, the product on the left converging to U and the product on the right converging to V^T. In the case of the QLP decomposition, that is, stopping after two iterations, there is only one matrix on each side, Q and P. Error bounds on how well they approximate the singular subspaces are available. See [MS93] and [Ste98, Chap. 5, Thm. 2.3]. Our focus will be on the convergence of the triangular matrix to the singular values.

2.2 Convergence of the QLP Decomposition

2.2.1 The Extreme Singular Values

First we will see how well the QLP decomposition approximates the extremal singular values, σ_1 and σ_n. In studying the QLP decomposition, we let the notation reflect the fact that we view the decomposition as the first two steps of the QLP iteration. Thus we use $Q^{(0)}$ instead of Q, $Q^{(1)}$ instead of P, and $(R^{(1)})^T$ instead of L. Stewart pointed out that the pivoting in the first step was crucial, but necessary in the second step only to avoid "certain contrived counterexamples." So to simplify the analysis, we will assume no pivoting on the second step. This is fine, since the results can only get better with pivoting.

We are not here seeking precise bounds, but only rough estimates sufficient to

explain the good convergence after only two steps. In analyzing the first step, we note that all rank-revealing-type algorithms perform roughly equally most of the time. This frees us to use bounds from any RRQR algorithm in our derivations, while still tacitly assuming a pivoted QR decomposition in the first step. We will use the bounds provided for "Hybrid-III" by Chandrasekaran and Ipsen [CI94].

Since we are interested in bounding the error in approximating σ_n, we let $k = n - 1$. In this case, the relevant bounds are:

$$|r_{nn}^{(0)}| \leq \sqrt{n}\,\sigma_n, \tag{2.1}$$

$$\inf(R_{11}^{(0)}) \geq \frac{1}{\sqrt{2(n-1)}}\,\sigma_n. \tag{2.2}$$

(The bound (2.1) is originally due to Golub and Van Loan [GL96].)

After the first step, we no longer have a full matrix A but an upper-triangular matrix $R^{(0)}$, so we may employ the analysis of Mathias and Stewart [MS93]. Their paper is divided into two parts. In the first, they studied the unpivoted iteration on block upper-triangular matrices. They showed [MS93, Theorem 2.1] that under the assumption

$$\rho^{(i)} = \frac{\|R_{22}^{(i)}\|}{\inf(R_{11}^{(i)})} < 1, \tag{2.3}$$

the following hold:

$$\|R_{12}^{(i+1)}\| \leq \rho^{(i)}\|R_{12}^{(i)}\|, \tag{2.4}$$

$$\sigma_j(R_{22}^{(i+1)}) \leq \sigma_j(R_{22}^{(i)}), \qquad j = 1, \ldots, n-k, \tag{2.5}$$

$$\sigma_j(R_{11}^{(i+1)}) \geq \sigma_j(R_{11}^{(i)}), \qquad j = 1, \ldots, k. \tag{2.6}$$

In the second part of their paper, they used these results to approximate the singular values of a block upper-triangular matrix R by the singular values of a block diagonal matrix $\hat{R}$. Partition R and $\hat{R}$ as

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \quad \text{and} \quad \hat{R} = \begin{pmatrix} R_{11} & 0 \\ 0 & R_{22} \end{pmatrix}.$$

In general, the singular values of $\hat{R}$ differ from those of R by no more than $\|R_{12}\|$ [GL96, Cor. 8.6.2]. Mathias and Stewart show that if

$$\rho = \frac{\|R_{22}\|}{\inf(R_{11})} < 1, \tag{2.7}$$

then

$$\sigma_j(R)/\sigma_j(\hat{R}) = 1 + \mathcal{O}\left(\frac{\|R_{12}\|^2}{(1-\rho^2)[\inf(R_{11})]^2}\right).$$

We will assume that (2.3) holds with $i = 0$, that is, after the first step. This is not an unreasonable assumption, since the pivoted QR factorization of the first

step roughly orders the embryonic singular values. Note that if A has j zero singular values, then this initial pivoting will put zeros as the last j elements on the diagonal of $R^{(0)}$. The matrix can then be deflated (the last j rows and j columns of $R^{(0)}$ discarded), and the iteration continued with $R^{(0)}$ having all non-zero singular values. We may therefore assume that there are no zero singular values.

Now we are ready to state our first theorem.

Theorem 2.2.1 *Let A be an m-by-n matrix and let $\sigma_{n-1}(A) > \sigma_n(A)$. Let $R^{(0)}$ be the R-factor in the pivoted QR factorization of A, $A\Pi = Q^{(0)}R^{(0)}$ and let $R^{(1)}$ be the R-factor in the unpivoted QR factorization of $(R^{(0)})^T$, $(R^{(0)})^T = Q^{(1)}R^{(1)}$. Assume that the bounds (2.1) and (2.2) hold and that $\rho^{(0)} = |r_{nn}^{(0)}|/inf(R_{11}^{(0)}) < 1$.*

Then

$$|r_{nn}^{(1)}| - \sigma_n \leq \frac{\sigma_n^3}{\sigma_{n-1}^2} \mathcal{O}\left(\frac{n^{\frac{5}{2}} \|(R_{12}^{(0)})\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2} \right). \tag{2.8}$$

Proof:

Since $\rho^{(0)} = |r_{nn}^{(0)}|/\inf(R_{11}^{(0)}) < 1$, by (2.5) and (2.6) we have $|r_{nn}^{(1)}| \leq |r_{nn}^{(0)}|$ and $\|R_{11}^{(1)}\| \geq \|R_{11}^{(0)}\|$, so that $\rho^{(1)} = |r_{nn}^{(1)}|/\inf(R_{11}^{(1)}) < 1$ also. Hence Mathias and Stewart's result [MS93, Theorem 3.1] gives

$$\frac{\sigma_n}{|r_{nn}^{(1)}|} \geq 1 - \mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2} \right),$$

or

$$|r_{nn}^{(1)}| - \sigma_n \leq |r_{nn}^{(1)}| \mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2)(\inf(R_{11}^{(1)})]^2}\right),$$

From this, the result follows:

$$\begin{aligned}
|r_{nn}^{(1)}| - \sigma_n &\leq |r_{nn}^{(1)}| \mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right) \\
&\leq |r_{nn}^{(1)}| \mathcal{O}\left(\frac{(\rho^{(0)})^2\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right) \qquad (2.9) \\
&\leq |r_{nn}^{(0)}| \mathcal{O}\left(\frac{(\rho^{(0)})^2\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right) \qquad (2.10) \\
&\leq \frac{|r_{nn}^{(0)}|^3}{\inf(R_{11}^{(0)})^2} \mathcal{O}\left(\frac{\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right) \\
&\leq \frac{\sigma_n^3}{\sigma_{n-1}^2} \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right). \qquad (2.11)
\end{aligned}$$

The inequality (2.9) follows from (2.4), (2.10) follows from (2.5), and (2.11) follows from (2.1) and (2.2). □

Note that here $R_{12}^{(1)}$ is a vector. Theorem 2.2.1 gives a bound on the error in approximating the smallest singular value σ_n of A by $|r_{nn}^{(1)}|$, the absolute value of the final element on the diagonal of $R^{(1)}$. We see that the error $|r_{nn}^{(1)}| - \sigma_n$ is cubic in σ_n. From equation (2.10), we see that one order comes from the $|r_{nn}^{(0)}|$ factor in front, which is bounded in the rank-revealing first step while two orders come from the $(\rho^{(0)})^2$ factor contributed in the second step. The inequality also shows that the relative error $(|r_{nn}^{(1)}| - \sigma_n)/\sigma_n$ is quadratic in σ_n/σ_{n-1}.

We illustrate Theorem 2.2.1 on a 30-by-30 matrix. We fix the largest 29 singu-

lar values to be spaced evenly from 1 to 10. We perform the QLP decomposition (with pivoting on the first step and no pivoting on the second step) five times, allowing the smallest singular value, σ_{30}, to take on the values $10^{-1}, 10^{-2}, \ldots, 10^{-5}$. We expect the absolute error $|r^{(1)}_{30,30}| - \sigma_{30}$ to decrease by 10^{-3} on each run and the relative error $(|r^{(1)}_{30,30}| - \sigma_{30})/\sigma_{30}$ to decrease by 10^{-2} on each run. This is verified in Figure 2.2.

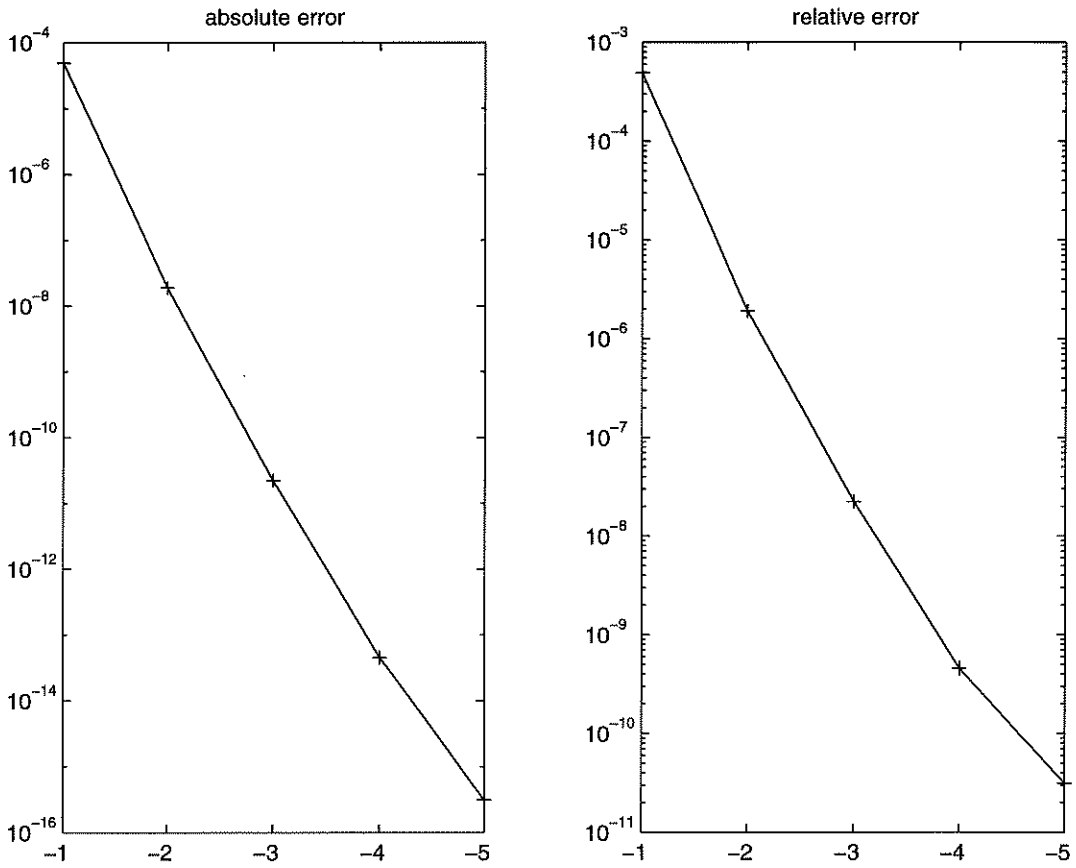

Figure 2.2: As σ_{30} decreases from 10^{-1} down to 10^{-5}, the absolute error $|r^{(1)}_{nn}| - \sigma_n$ decreases as the cube of σ_{30}, and the relative error $(|r^{(1)}_{nn}| - \sigma_n)/\sigma_n$ decreases as the square of $\sigma_{30}/\sigma_{29} = \sigma_{30}/1 = \sigma_{30}$.

There is an analogous result for $r^{(1)}_{11}$. Because the pivoted QR factorization makes $|r^{(0)}_{11}|$ equal to the norm of the largest column of A, $|r^{(0)}_{11}| = \max_{1 \le j \le n} \|Ae_j\|_2^2$,

where e_j is the j-th canonical vector, we do not need to borrow an RRQR bound for $|r_{11}^{(0)}|$. For any $A \in \mathbb{R}^{l \times m}$, we have

$$\max_{1 \le j \le n} \|Ae_j\|_2 \ge \frac{\|A\|_2}{\sqrt{n}}. \tag{2.12}$$

This is a standard result: $\|A\|_2^2 = \max_{\|x\|_2=1} \|Ax\|_2^2 \le \|A\|_F^2 \le n \max_{1 \le j \le n} \|Ae_j\|_2^2$. The first inequality is from, for example, Theorem 2.5 on p. 175 of [Ste73].

When we take the column pivoted QR factorization of a matrix A, the absolute value of the element r_{11} in the upper-left corner is equal to $\max_{1 \le j \le n} \|Ae_j\|_2$. Equation (2.12) therefore tells us that $|r_{11}^{(0)}|$ is greater than or equal to $\sigma_1/\sqrt{n}$. So here we have an RRRQ bound automatically. The one RRQR bound assumed for this case is [CI94]:

$$\|R_{22}^{(0)}\| \le \sqrt{2(n-1)}\,\sigma_2. \tag{2.13}$$

The theorem is very similar to Theorem 2.2.1.

Theorem 2.2.2 *Let A be an m-by-n matrix and let $\sigma_1(A) > \sigma_2(A)$. Let $R^{(0)}$ be the R-factor in the pivoted QR factorization of A, $A\Pi = Q^{(0)}R^{(0)}$ and let $R^{(1)}$ be the R-factor in the unpivoted QR factorization of $(R^{(0)})^T$, $(R^{(0)})^T = Q^{(1)}R^{(1)}$. Assume that the bound (2.13) holds and that $\rho^{(0)} = \|R_{22}^{(0)}\|/|r_{11}^{(0)}| < 1$.*

Then

$$|r_{11}^{(1)}|^{-1} - \sigma_1^{-1} \leq \frac{\sigma_2^2}{\sigma_1^3} \mathcal{O}\left(\frac{n^{\frac{5}{2}} \|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2} \right). \tag{2.14}$$

Proof:

Since $\rho^{(0)} = \|R_{22}^{(0)}\|/|r_{11}^{(0)}| < 1$, by (2.5) and (2.6) we have $\|R_{22}^{(1)}\| \leq \|R_{22}^{(0)}\|$ and $|r_{11}^{(1)}| \geq |r_{11}^{(0)}|$, so that $\rho^{(1)} = \|R_{22}^{(1)}\|/|r_{11}^{(1)}| < 1$. Hence Mathias and Stewart's result [MS93, Theorem 3.1] gives

$$\frac{|r_{11}^{(1)}|}{\sigma_1} \geq 1 - \mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2} \right).$$

or

$$|r_{11}^{(1)}|^{-1} - \sigma_1^{-1} \leq |r_{11}^{(1)}|^{-1} \mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2} \right).$$

From this, the result follows:

$$
\begin{aligned}
|r_{11}^{(1)}|^{-1} - \sigma_1^{-1} &\leq |r_{11}^{(1)}|^{-1} \mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2}\right) \\
&\leq |r_{11}^{(1)}|^{-1} \mathcal{O}\left(\frac{(\rho^{(0)})^2\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2}\right) \qquad (2.15) \\
&\leq |r_{11}^{(0)}|^{-1} \mathcal{O}\left(\frac{(\rho^{(0)})^2\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2}\right) \qquad (2.16) \\
&\leq \frac{\|R_{22}^{(0)}\|^2}{|r_{11}^{(0)}|^3} \mathcal{O}\left(\frac{\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2}\right) \\
&\leq \frac{\|R_{22}^{(0)}\|^2}{\sigma_1^3} \mathcal{O}\left(\frac{n^{\frac{3}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2}\right) \qquad (2.17) \\
&\leq \frac{\sigma_2^2}{\sigma_1^3} \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2](r_{11}^{(1)})^2}\right). \qquad (2.18)
\end{aligned}
$$

The inequality (2.15) follows from (2.4), (2.16) follows from 2.6, (2.17) follows from equation (2.12), and (2.18) follows from 2.13. □

Note that in Theorem 2.2.2, the quantities in question are reciprocals, and the convergence is cubic in $1/\sigma_1$. We illustrate the theorem using a 30-by-30 matrix as we did Theorem 2.2.2. This time we fix the smallest 29 singular values to be spaced evenly from 0.1 to 1. We allow σ_1 to increase from 10 to 10^5 over five runs, and we expect to see the absolute error decrease by a factor of 10^3 and the relative error decrease by a factor of 10^2 with each run. See Figure 2.3 for the results.

2.2.2 The Interior Singular Values

There are generalizations of the above bounds in terms of the norms of R_{11}^{-1} and R_{22}. That is, if R_{11} has dimension k and hence R_{22} dimension $n-k$, we can

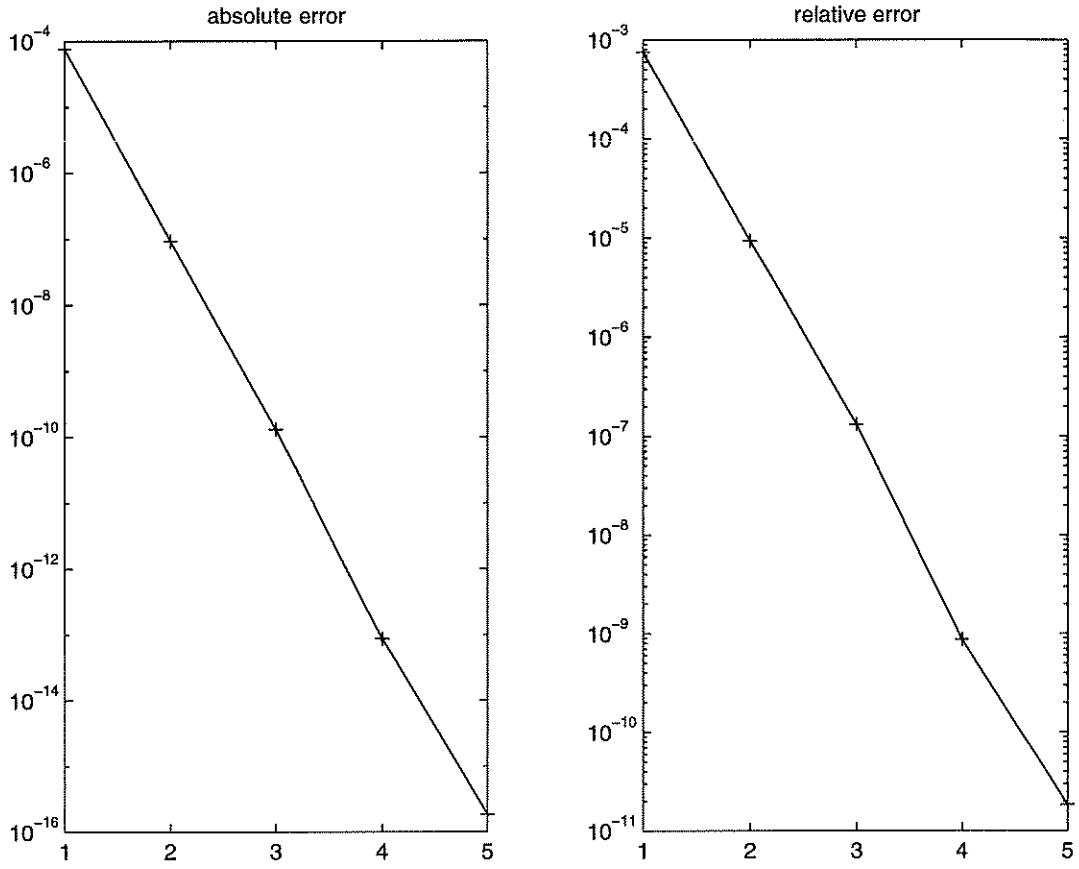

Figure 2.3: As σ_1 increases from 10^1 up to 10^5, the absolute error $|r_{11}^{(1)}|^{-1} - \sigma_1^{-1}$ decreases as the cube of σ_1^{-1}, and the relative error $(|r_{11}^{(1)}|^{-1} - \sigma_1^{-1})/\sigma_1^{-1}$ decreases as the square of $\sigma_2/\sigma_1 = 1/\sigma_1 = \sigma_1^{-1}$.

bound $\|(R_{11}^{(1)})^{-1}\| - \sigma_k^{-1}$ and $\|R_{22}^{(1)}\| - \sigma_{k+1}$.

For general k, the bounds provided for "Hybrid-III" by Chandrasekaran and Ipsen [CI94] are as follows:

$$\|R_{22}^{(0)}| \leq \sqrt{(k+1)(n-k)}\,\sigma_{k+1}, \tag{2.19}$$

$$\inf(R_{11}^{(0)}) \geq \frac{\sigma_k}{\sqrt{k(n-k+1)}}. \tag{2.20}$$

The general result is given in the following theorem.

Theorem 2.2.3 *Let A be an m-by-n matrix and let $\sigma_k(A) > \sigma_{k+1}(A)$. Let $R^{(0)}$ be the R-factor in the pivoted QR factorization of A, $A\Pi = Q^{(0)}R^{(0)}$ and let $R^{(1)}$ be the R-factor in the unpivoted QR factorization of $(R^{(0)})^T$, $(R^{(0)})^T = Q^{(1)}R^{(1)}$. Assume that the bounds (2.19) and (2.20) hold and that $\rho^{(0)} = \|R_{22}^{(0)}\|/inf(R_{11}^{(0)}) < 1$.*

Then for $j = 1, \ldots, n-k$,

$$\frac{\sigma_j(R_{22}^{(1)}) - \sigma_{k+j}}{\sigma_{k+j}} \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^2 \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right), \tag{2.21}$$

and for $j = 1, \ldots, k$,

$$\frac{\sigma_j(R_{11}^{(1)})^{-1} - \sigma_j^{-1}}{\sigma_j^{-1}} \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^2 \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right). \tag{2.22}$$

Proof: The proofs of inequalities (2.21) and (2.22) are similar to the proofs of Theorems 2.2.1 and 2.2.2, respectively. For (2.21), for example, Mathias and Stewart give us the first line

$$\sigma_j(R_{22}^{(1)}) - \sigma_{k+j} \leq \sigma_j(R_{22}^{(1)})\mathcal{O}\left(\frac{\|R_{12}^{(1)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right),$$

and the rest follows as in the proof of Theorem 2.2.1. □

Theorem 2.2.3 tells us that when there is a gap between σ_k and σ_{k+1}, the

relative error in each of the singular values is quadratic in the gap ratio σ_{k+1}/σ_k. For the singular values at the gap, the absolute error is cubic in the singular value, as the following corollary spells out.

Corollary 2.2.4 *Under the assumptions of Theorem 2.2.3, we have*

$$\|R_{22}^{(1)}\| - \sigma_{k+1} \leq \frac{\sigma_{k+1}^3}{\sigma_k^2} \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right), \tag{2.23}$$

and

$$\|(R_{11}^{(1)})^{-1}\| - \sigma_k^{-1} \leq \frac{\sigma_{k+1}^2}{\sigma_k^3} \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(1)})^2][\inf(R_{11}^{(1)})]^2}\right). \tag{2.24}$$

Note from the bounds in Theorem 2.2.3 that the corollary represents the worst case. That is, the bounds are better as we go away from the gap. Theorem 2.2.3 helps explain why the QLP decomposition does so well even when the ratios between neighboring singular values are close to one. As long as there is one substantial gap somewhere in the singular values, convergence will be fast for all of them.

So we see how the pivoting is so important to the QLP decomposition. The pivoting in the first step roughly orders the embryonic singular values. In the theorems, this allows us to make the reasonable assumption (2.3) for $i = 0$. This ordering then allows the quadractic convergence factor in the second step.

In addition to providing this ordering, the pivoting in the first step also gives us a linear convergence factor (in the form of the RRQR bounds). Thus the rank-revealing nature of the pivoting furnishes one order of convergence, while

the ordering nature of the pivoting sets the table for the second step to provide two additional orders of convergence.

Note that neither the assumption (2.3) nor the assumed RRQR bounds necessarily hold for the column pivoted QR decomposition. But the RRQR bounds often do, and (2.3) is more probable the larger the gap, so this analysis helps explain why QLP apparently does so well in practice. Without pivoting, the two-step algorithm not only fails to provide the linear convergence factor in the first step; more importantly, it often fails to provide the ordering needed for the quadratic convergence factor in the second step. Total failure is often seen with graded matrices. Pivoting is key.

We now illustrate Theorem 2.2.3 and Corollary 2.2.4 with a couple of examples. In each example we use a 100-by-100 matrix, let there be gap between σ_{50} and σ_{51}, and once again let the gap ratio σ_{51}/σ_{50} decrease from 10^{-1} initially to 10^{-5} in the fifth run.

The first example illustrates the bound (2.23). We fix the fifty largest singular values to be equally spaced from 1 to 10. The lowest fifty singular values are also equally spaced, but their range varies from 10^{-1} to 10^{-2} on the first run, from 10^{-2} to 10^{-3} on the second run, and so forth. As the gap ratio σ_{51}/σ_{50} thus decreases by a factor of 10 each run, we expect to see the absolute error $\|R_{22}^{(1)}\| - \sigma_{51}$ and the relative error $(\|R_{22}^{(1)}\| - \sigma_{51})/\sigma_{51}$ decrease by factors of 10^3 and 10^2, respectively. This is shown in Figure 2.4.

In the second example, we illustrate the bound (2.22). This time we fix the smallest fifty singular values to be equally spaced from 0.1 to 1. The largest fifty singular values range from 10 to 10^2, then 10^2 to 10^3, etc. As the gap ratio thus decreases by a factor of 10 each run, we expect to see the relative error in each of the first fifty singular values decrease by a factor of 10^2. This is borne out by

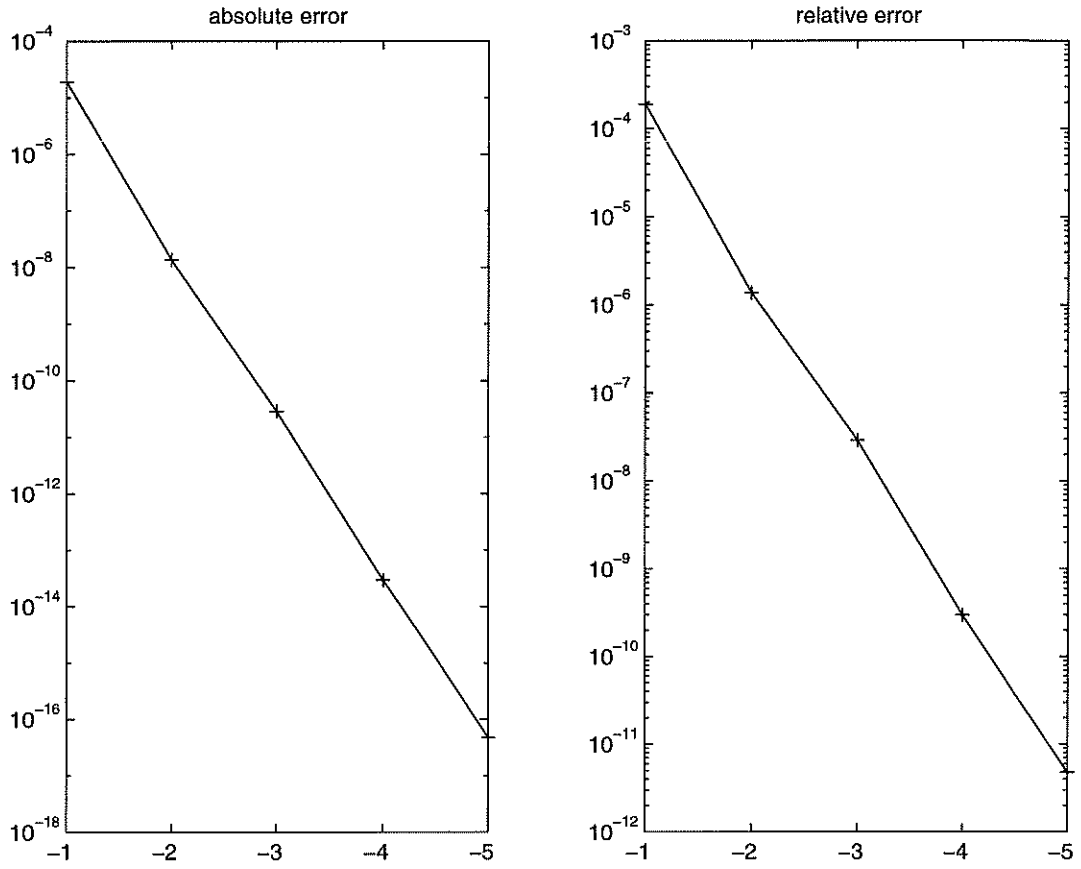

Figure 2.4: As σ_{51} decreases from 10^{-1} down to 10^{-5}, the absolute error $\|R_{22}^{(1)}\| - \sigma_{51}$ decreases as the cube of σ_{51}, and the relative error $(\|(R_{22}^{(1)}\| - \sigma_{51})/\sigma_{51}$ decreases as the square of $\sigma_{51}/\sigma_{50} = \sigma_{51}/1 = \sigma_{51}$.

Figure 2.5, in which we plot the relative error in σ_1^{-1} and σ_{40}^{-1}.

2.3 Convergence of the QLP Iteration

In section 2.1.2 we discussed taking the QLP decomposition beyond just the first two steps, resulting in an iteration (the QLP iteration) that converges to the singular values. In this section, we take two approaches to studying the convergence of the QLP iteration.

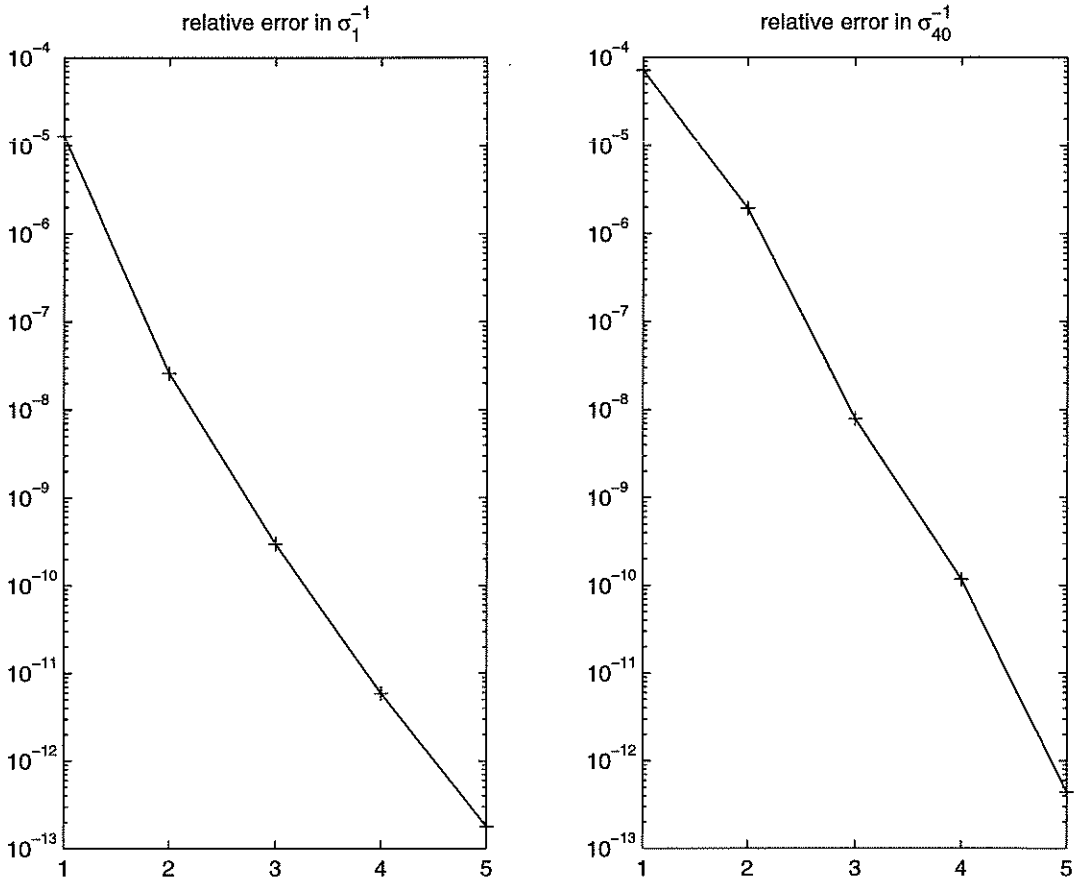

Figure 2.5: As σ_{50} increases from 10 up to 10^5, the relative error $(\sigma_1(R_{11}^{(1)})^{-1} - \sigma_1^{-1})/\sigma_1^{-1}$, plotted on the left, and the relative error $(\sigma_{40}(R_{11}^{(1)})^{-1} - \sigma_{40}^{-1})/\sigma_{40}^{-1}$, plotted on the right, both decrease as the square of of $\sigma_{51}/\sigma_{50} = 1/\sigma_{50}$.

2.3.1 More Iterations

In sections 2.2.1 and 2.2.2 we saw that with a gap between σ_k and σ_{k+1}, the QLP decomposition approximates all of the singular values with a relative error depending on the square of the gap ratio σ_{k+1}/σ_k. If we iterate beyond the first two steps (assuming that pivoting is used on the first step and no pivoting on each subsequent step), the error bounds improve by a quadratic factor in each step. To see this, we can simply apply the bound (2.4) at each iteration.

Theorem 2.3.1 *Let A be an m-by-n matrix and let $\sigma_k(A) > \sigma_{k+1}(A)$. Let $R^{(0)}$ be the R-factor in the pivoted QR factorization of A, $A\Pi = Q^{(0)}R^{(0)}$ and let $R^{(i)}$, $i \geq 1$, be the R-factor in the unpivoted QR factorization of $(R^{(i-1)})^T$, $(R^{(i-1)})^T = Q^{(i)}R^{(i)}$. Assume that the bounds (2.19) and (2.20) hold and that $\rho^{(0)} = \|R_{22}^{(0)}\|/inf(R_{11}^{(0)}) < 1$.*

Then for $j = 1, \ldots, n-k$,

$$\frac{\sigma_j(R_{22}^{(i)}) - \sigma_{k+j}}{\sigma_{k+j}} \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2i} \mathcal{O}\left(\frac{n^{\frac{4i+1}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(i)})^2][\inf(R_{11}^{(i)})]^2}\right), \tag{2.25}$$

and for $j = 1, \ldots, k$,

$$\frac{\sigma_j(R_{11}^{(i)})^{-1} - \sigma_j^{-1}}{\sigma_j^{-1}} \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2i} \mathcal{O}\left(\frac{n^{\frac{4i+1}{2}}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(i)})^2][\inf(R_{11}^{(i)})]^2}\right). \tag{2.26}$$

Proof:

The proofs are similar to those given before. For example, in proving the bound (2.25), the only real difference from the proof of Theorem 2.2.1 is inequality (2.9), different here because we are iterating. So the first two lines of the series of inequalities are as follows:

$$\begin{aligned}
\sigma_j(R_{22}^{(i)}) - \sigma_{k+j} &\leq \sigma_j(R_{22}^{(i)})\mathcal{O}\left(\frac{\|R_{12}^{(i)}\|^2}{[1-(\rho^{(i)})^2][\inf(R_{11}^{(i)})]^2}\right) \\
&\leq \sigma_j(R_{22}^{(i)})\mathcal{O}\left(\frac{(\rho^{(0)})^{2i}\|R_{12}^{(0)}\|^2}{[1-(\rho^{(i)})^2][\inf(R_{11}^{(i)})]^2}\right).
\end{aligned}$$

The rest of the proof proceeds as expected. □

2.3.2 Assymptotic Convergence of Individual Elements

We would now like to make some observations and conjectures on the rate of convergence of individual elements in the $R^{(i)}$ matrices as well as individual elements in the $U^{(i)}$ and $V^{(i)}$ matrices, where $R^{(i)} = U^{(i)}\Sigma(V^{(i)})^T$. Unfortunately, we cannot provide hard and fast proofs for everything, but we will describe what we see. It appears that the convergence of the diagonal elements of the $R^{(i)}$ depends on the convergence of the off-diagonal elements. We will prove a statement concerning this dependence. The convergence of the off-diagonal elements appears complex, and we will give some observations.

We already know that the $R^{(i)}$ converge to a diagonal matrix whose elements are the singular values of A in decreasing order. So in discussing asymptotic rates of convergence of elements in $R^{(i)}$, we may assume that the diagonal elements are close to the σ_j and that all of the off-diagonal elements of $R^{(i)}$ are very small. With pivoting in the first step, convergence often reaches the asymptotic range quickly, after the first two or three iterations. Note that whether pivoting is used (in the first step only or on every step) or not, there will come an iteration after which pivoting ceases anyway. Hence, it would seem that asymptotic rates of convergence would not depend on whether we pivot. This intuition is wrong, however, for the acceleration in convergence afforded by pivoting extends in some cases even to the asymptotic convergence rates. We will comment on this.

Either Householder transformations or Givens rotations can be used to triangularize each $(R^{(i)})^T$. We will assume the use of Givens rotations, since their action is easier to track and visualize, and because they illuminate the rates of

convergence for specific elements.

Let us look at how the elements of $(R^{(i)})^T$ could change while the matrix is being upper-triangularized via Givens rotations. The elements $(r^{(i)})^T_{21}$ through $(r^{(i)})^T_{n1}$ are first zeroed out, then the elements $(r^{(i)})^T_{32}$ through $(r^{(i)})^T_{n2}$ are zeroed out, and so on. We will assume that elements in each column are zeroed from top to bottom. Doing so gives us a useful analytical result. It is easy to verify that zeroing out the elements in this order leaves the submatrix consisting of the bottom $n-1$ rows and $n-1$ columns in lower-triangular form. This means that in zeroing out the second and subsequent columns, the mechanics look exactly the same.

When an element $(r^{(i)})^T_{jk}$ is zeroed out, the Givens rotation will change entries only in rows j and k. Take a look at Figure 2.6.

$$\begin{pmatrix} x & x & x & x & x & x \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{0} & \\ & & x & & & \\ & & x & x & & \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \\ & x & x & x & x & x \end{pmatrix} \qquad \begin{pmatrix} x & x & x & x & x & x \\ & \mathbf{a} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{0} & \\ & & x & & & \\ & & x & x & & \\ & \mathbf{h} & \mathbf{d_1} & \mathbf{d_2} & \mathbf{b} & \\ & x & x & x & x & x \end{pmatrix}$$

Figure 2.6: The element $(r^{(i)})^T_{52}$ is about to be zeroed out by a Givens rotation. On the left the elements which at this point are nonzero are marked with an x. The bold elements will be changed by the Givens rotation. On the right, depending on what type of change the Givens rotation will effect, the changing elements are assigned various letters.

In Figure 2.6 we have assigned different letters to the nonzero elements that change to indicate for each element what type of change the Givens rotation will effect. For example, the elements c_1 and c_2 will be changed in the same way. Figure 2.7 shows exactly how each element is changed.

First of all, note that with every Givens rotation that is applied, the diagonal

$$\begin{pmatrix} a & \cdots & c_s & \cdots & 0 \\ h & \cdots & d_s & \cdots & b \end{pmatrix} \longrightarrow \begin{pmatrix} \sqrt{a^2+h^2} & \cdots & \frac{ac_s+hd_s}{\sqrt{a^2+h^2}} & \cdots & \frac{hb}{\sqrt{a^2+h^2}} \\ 0 & \cdots & \frac{ad_s-hc_s}{\sqrt{a^2+h^2}} & \cdots & \frac{ab}{\sqrt{a^2+h^2}} \end{pmatrix}$$

Figure 2.7: A Givens rotation affects elements in two rows. Those elements are shown on the left, with letters indicating the type of change they will undergo. On the right, the elements appear as they are after the Givens rotation.

elements will always play the role of either a or b. In fact, the diagonal element $(r^{(i)})^T_{jj}$ will first play the role of b precisely $j-1$ times, namely when the $j-1$ elements to the left of it in row j are being zeroed. It will then play the role of a precisely $n-j$ times, namely when the $n-j$ elements below it in column j are being zeroed.

So the convergence of a diagonal element is completely determined by these $n-1$ Givens rotations. Looking at Figure 2.7, we see that whether the element is acting as a or as b for a given rotation, it receives an $\mathcal{O}(h^2)$ perturbation, where h is the off-diagonal element being zeroed by the given rotation. As all of these $n-1$ off-diagonal elements are converging to zero, it is obvious that asymptotically, only the largest of the $n-1$ elements will contribute significantly to the convergence of the diagonal element. The rest of the $n-1$ elements will only slightly alter the value of the diagonal element at each iteration. Since asymptotically, the largest of the $n-1$ elements will be the one(s) with the slowest asymptotic convergence rate, it is acceptable to ignore the contributions from all the other off-diagonal elements.

Let us make this more precise in the context of an example. We will write things in terms of a small quantity $\epsilon \ll \sigma_n$, and then we are free to discard lower order terms. Say we are looking at the essential limit $r_{11}^{(\infty)}$ of $r_{11}^{(i)}$, which will play the role of a for each of the $n-1$ Givens rotations that affect it each iteration. If $r_{k1}^{(i)}$ is asymptotically converging the slowest of all the elements in

the first column below $r_{11}^{(i)}$, then we ignore the contribution of all the others. (We will later consider the case when two or more of these elements share the slowest convergence rate.) Let us assume that $|r_{k1}^{(i)}| < \epsilon$ for all $i \geq i_0$, some i_0 and all the other $|r_{j1}^{(i)}|, j \neq 1$ are less than ϵ^2 for all $i \geq i_0$. Let $\lambda_1, \ldots, \lambda_{n-1}$ be the asymptotic convergence rates of each of the $r_{j1}^{(i)}, 1 < j \leq n$. Then for $i \geq i_0$ we assume that $r_{j1}^{(i)}$ is converging with a rate of $\lambda_j + \delta_i$, where $|\delta_i| < \epsilon$. We will require i_0 sufficiently large so that the largest (slowest) rate satisfies $\lambda_k + \epsilon < 1$. Letting $h = r_{k1}^{(i)}$, we look back at Figure 2.7 and see that for $i \geq i_0$,

$$\begin{aligned} r_{11}^{(i)} &= \sqrt{(r_{11}^{(i-1)})^2 + \sum_{j=2}^{n} (r_{j1}^{(i)})^2} \\ &= \sqrt{(r_{11}^{(i-1)})^2 + h^2} + \mathcal{O}(\epsilon^4), \end{aligned}$$

We see that of all the $n-1$ off-diagonal terms that could contribute to $r_{11}^{(i)}$, only $h = r_{k1}^{(i)}$ makes an $\mathcal{O}(\epsilon^2)$ contribution. All the rest effect only $\mathcal{O}(\epsilon^4)$ perturbations. So in a similar fashion we have

$$\begin{aligned} r_{11}^{(i+1)} &= \sqrt{(r_{11}^{(i-1)})^2 + h^2 + [(\lambda_k + \delta_{i+1})h]^2} + \mathcal{O}(\epsilon^4) \\ r_{11}^{(i+2)} &= \sqrt{(r_{11}^{(i-1)})^2 + h^2 + [(\lambda_k + \delta_{i+1})h]^2 + [(\lambda_k + \delta_{i+1})(\lambda_k + \delta_{i+2})h]^2} + \mathcal{O}(\epsilon^4) \\ &\cdots \\ r_{11}^{(\infty)} &= \sqrt{(r_{11}^{(i-1)})^2 + h^2 + \sum_{j=0}^{\infty} \left[\left(\prod_{a=i_0+1}^{i_0+1+j} (\lambda_k + \delta_a) \right) h \right]^2} + \mathcal{O}(\epsilon^4). \end{aligned} \tag{2.27}$$

In equation (2.27), there might be a question as to whether we can actually sweep the contributions of all of the $n-2$ other off-diagonal elements into the

$\mathcal{O}(\epsilon^4)$, since each of these elements contributes an infinite series similar to that of $h = r_{k1}^{(i)}$. The contribution from a generic one $\hat{h} = r_{b1}^{(i)}$ can be bounded as follows:

$$\begin{aligned}
\sum_{j=0}^{\infty}\left[\left(\prod_{a=i_0+1}^{i_0+1+j}(\lambda_b+\delta_a)\right)\hat{h}\right]^2 &\leq \sum_{j=0}^{\infty}\left[(\lambda_b \pm \epsilon)^{j+1}\hat{h}\right]^2 \\
&= \hat{h}^2(\lambda_b+\epsilon)^2\sum_{j=0}^{\infty}(\lambda_b \pm \epsilon)^{2j} \\
&= \hat{h}^2(\lambda_b \pm \epsilon)^2\frac{1}{1-(\lambda_b \pm \epsilon)^2} \\
&= \hat{h}^2(\lambda_b \pm \epsilon)^2(1+\mathcal{O}\left((\lambda_b \pm \epsilon)^2\right) \\
&= \mathcal{O}(\epsilon^4),
\end{aligned}$$

since $\hat{h} = \mathcal{O}(\epsilon^4)$ and $\lambda_b + \epsilon < \lambda_k + \epsilon < 1$ by assumption when $i \geq i_0$. So equation (2.27) holds.

Now we have a feel for the proof of the following theorem, which concerns the asymptotic convergence rates of the diagonal elements.

Theorem 2.3.2 *Let $\epsilon \ll \sigma_n$. Let i_0 be an integer sufficiently large so that for $i \geq i_0$ the following hold:*

1. *For each $1 \leq s \leq n$, denote by h_s the element having the slowest (i.e., largest) asymptotic convergence rate of all the elements in the same row or column as $r_{ss}^{(i)}$, namely, $(r^{(i)})_{s,1}^T, \ldots, (r^{(i)})_{s,s-1}^T$ and $(r^{(i)})_{s+1,s}^T, \ldots, (r^{(i)})_{n,s}^T$, and denote the asymptotic convergence rate of h_s by λ_s. If more than one of these $n-1$ elements share the same slowest asymptotic convergence rate, denote them by h_{s_1}, h_{s_2}, etc.*

2. *Assume the rate of convergence of each of these $n-1$ elements is within ϵ*

of its asymptotic convergence rate. So for example, the rate of the element h_s *at iteration* $i \geq i_0$ *is* $\lambda_s + \delta_i$, *where* $|\delta_i| < \epsilon$. *Similarly for the rest of these* $n-2$ *elements.*

3. *Assume all of the off-diagonal elements of the matrix* $R^{(i)}$ *are less than* ϵ^2 *in absolute value, except for the* $h_{s_j}, 1 \leq s \leq n$, *which need only be less than* ϵ *in absolute value.*

4. *Assume that each of the diagonal elements* $r_{ss}^{(i)}, 1 \leq s \leq n$ *is within* ϵ^2 *of* σ_s.

5. *Assume* $\lambda_s + \epsilon < 1$, *for* $1 \leq s \leq n$.

Then the convergence rate of $r_{ss}^{(i)}$ *is* $\lambda_s^2 + \mathcal{O}(\epsilon)$.

Proof: Without loss of generality, we assume that the diagonal elements of the $R^{(i)}$ are positive.

First let $s = 1$.

Say there is only one element h_1 among the $n-1$ off-diagonal elements in the first column of $R_{(i)}^T$ that converges with the slowest asymptotic rate λ_1. We have already derived formulae for $r_{11}^{(i)}$ and $r_{11}^{(\infty)}$ above. Let $P = \prod_{a=i_0+1}^{i_0+1+j}(\lambda_1 + \delta_a)$. We have

$$\begin{aligned} r_{11}^{(\infty)} - r_{11}^{(i)} &= \sqrt{(r_{11}^{(i-1)})^2 + h_1^2 + \sum_{j=0}^{\infty}(Ph_1)^2} - \sqrt{(r_{11}^{(i-1)})^2 + h_1^2} + \mathcal{O}(\epsilon^4) \\ &= |r_{11}^{(i-1)}| \left(1 + \frac{1}{2}\frac{h_1^2 + \sum_{j=0}^{\infty}(Ph_1)^2}{(r_{11}^{(i-1)})^2} - 1 - \frac{1}{2}\frac{h_1^2}{(r_{11}^{(i-1)})^2}\right) + \mathcal{O}(\epsilon^4) \\ &= \frac{1}{2}\frac{\sum_{j=0}^{\infty}(Ph_1)^2}{r_{11}^{(i-1)}} + \mathcal{O}(\epsilon^4). \end{aligned}$$

Similarly,

$$\begin{aligned} r_{11}^{(\infty)} - r_{11}^{(i+1)} &= \sqrt{(r_{11}^{(i-1)})^2 + h_1^2 + \sum_{j=0}^{\infty} (Ph_1)^2} \\ &\qquad - \sqrt{(r_{11}^{(i-1)})^2 + h_1^2 + ((\lambda_1 + \delta_{i_0+1})h_1)^2} + \mathcal{O}(\epsilon^4) \\ &= \frac{1}{2} \frac{\sum_{j=1}^{\infty} (Ph_1)^2}{r_{11}^{(i-1)}} + \mathcal{O}(\epsilon^4). \end{aligned}$$

So the asymptotic rate of convergence of $r_{11}^{(i)}$ is

$$\begin{aligned} \frac{r_{11}^{(\infty)} - r_{11}^{(i+1)}}{r_{11}^{(\infty)} - r_{11}^{(i)}} &= \frac{\frac{1}{2} \frac{\sum_{j=1}^{\infty} (Ph_1)^2}{r_{11}^{(i-1)}}}{\frac{1}{2} \frac{\sum_{j=0}^{\infty} (Ph_1)^2}{r_{11}^{(i-1)}}} + \mathcal{O}(\epsilon^4) \\ &= \frac{\sum_{j=1}^{\infty} (Ph_1)^2}{\sum_{j=0}^{\infty} (Ph_1)^2} + \mathcal{O}(\epsilon^4). \end{aligned}$$

To bound this, we can set all of the δ_a in $P = \prod_{a=i_0+1}^{i_0+1+j} (\lambda_1 + \delta_a)$ to their maximum values, obtaining

$$\begin{aligned} \frac{h_1^2 \sum_{j=1}^{\infty} (\lambda_1 \pm \epsilon)^{2j+4}}{h_1^2 \sum_{j=0}^{\infty} (\lambda_1 \pm \epsilon)^{2j+2}} + \mathcal{O}(\epsilon^4) &= \frac{(\lambda_1 \pm \epsilon)^2 \frac{1}{1-(\lambda_1 \pm \epsilon)^2}}{\frac{1}{1-(\lambda_1 \pm \epsilon)^2}} + \mathcal{O}(\epsilon^4) \\ &= (\lambda_1 \pm \epsilon)^2 + \mathcal{O}(\epsilon^4) \\ &= \lambda_1^2 + \mathcal{O}(\epsilon) \end{aligned}$$

Now consider the case when $s = n$. The only Givens rotations that affect $(r^{(i)})_{nn}^T$ are the ones that zero $(r^{(i)})_{n1}^T, \ldots, (r^{(i)})_{n,n-1}^T$. Again assume there is

precisely one of these elements, say $(r^{(i)})_{k1}^T$, which has the slowest asymptotic convergence rate. Set $h_n = (r^{(i)})_{k1}^T$, and let its asymptotic convergence rate be λ_n.

We can look at Figure 2.7 to see how things will work. The b in Figure 2.7 will be here $r_{nn}^{(i-1)}$, and at each iteration this will be multiplied by $a/\sqrt{a^2+h^2}$. The element a will be $r_{kk}^{(i)}$ and the element h will be h_n, both of which change at each iteration. All the elements in the last row of $(R^{(i)})^T$ other than h_n will cause only $\mathcal{O}(\epsilon^4)$ perturbations as before. This time we let $P = \prod_{a=i_0+1}^{i_0+j}(\lambda_n+\delta_a)$, and we use the convention that $\prod_{a=i_0+1}^{i_0}(\lambda_n+\delta_a) = 1$. We have

$$
\begin{aligned}
r_{nn}^{(\infty)} - r_{nn}^{(i)} &= r_{nn}^{(i-1)} \prod_{j=0}^{\infty} \frac{r_{kk}^{(i+j)}}{\sqrt{(r_{kk}^{(i+j)})^2 + (Ph_n)^2}} \\
&\qquad - r_{nn}^{(i-1)} \frac{r_{kk}^{(i)}}{\sqrt{(r_{kk}^{(i)})^2 + h_n^2}} + \mathcal{O}(\epsilon^4) \qquad (2.28) \\
&= r_{nn}^{(i-1)} \prod_{j=0}^{\infty} \left[1 - \frac{1}{2} \frac{h_n^2}{(r_{kk}^{(i+j)})^2} P^2 \right] \\
&\qquad - r_{nn}^{(i-1)} \left(1 - \frac{1}{2} \frac{h_n^2}{(r_{kk}^{(i)})^2} \right) + \mathcal{O}(\epsilon^4) \\
&= r_{nn}^{(i-1)} \left(1 - \frac{h_n^2}{2} \sum_{j=0}^{\infty} \frac{P^2}{(r_{kk}^{(i+j)})^2} - 1 + \frac{1}{2} \frac{h_n^2}{(r_{kk}^{(i)})^2} \right) + \mathcal{O}(\epsilon^4) \\
&= -r_{nn}^{(i-1)} \frac{h_n^2}{2} \sum_{j=1}^{\infty} \frac{P^2}{(r_{kk}^{(i+j)})^2} + \mathcal{O}(\epsilon^4).
\end{aligned}
$$

This leads to

$$\frac{r_{nn}^{(\infty)} - r_{nn}^{(i+1)}}{r_{nn}^{(\infty)} - r_{nn}^{(i)}} = \frac{-r_{nn}^{(i-1)} \frac{h_n^2}{2} \sum_{j=2}^{\infty} \frac{P^2}{(r_{kk}^{(i+j)})^2}}{-r_{nn}^{(i-1)} \frac{h_n^2}{2} \sum_{j=1}^{\infty} \frac{P^2}{(r_{kk}^{(i+j)})^2}} + \mathcal{O}(\epsilon^4). \tag{2.29}$$

To bound this, we once again set all of the δ_a in $P = \prod_{a=i_0+1}^{i_0+j}(\lambda_n + \delta_a)$ to their maximum values. We also make use of the fact that the diagonal elements are within ϵ^2 of the singular values to which they are converging. This gives us

$$\frac{\sum_{j=2}^{\infty} \frac{(\lambda_n \pm \epsilon)^{2j+2}}{\sigma_k \pm \epsilon^2}}{\sum_{j=1}^{\infty} \frac{(\lambda_n \pm \epsilon)^{2j}}{\sigma_k \pm \epsilon^2}} + \mathcal{O}(\epsilon^4) = \lambda_n^2 + \mathcal{O}(\epsilon). \tag{2.30}$$

Finally let $s = l$. The only Givens rotations that affect $(r^{(i)})_{ll}^T$ are the ones that zero $(r^{(i)})_{l1}^T, \ldots, (r^{(i)})_{l,l-1}^T$, and the ones that zero $(r^{(i)})_{l+1,l}^T, \ldots, (r^{(i)})_{n,l}^T$. Assume one of these has the slowest asymptotic convergence rate. Then the proof for either the $s = 1$ case or the $s = n$ case carries over.

If in any of the cases more than one of these elements converge at λ_s, the same proofs hold with minor modifications. For example, say that both $(r^{(i)})_{pl}^T$ and $(r^{(i)})_{lq}^T$ converge at the rate λ_s. Denote $r_{ql}^{(i-1)}$ by h_{s_1} and $r_{lp}^{(i-1)}$ by h_{s_2}. So looking once again at Figure 2.7, $(r^{(i)})_{ll}^T$ plays the role of b with $h = h_{s_1}$ and later plays the role of a with $h = h_{s_2}$. Hence $r_{ll}^{(\infty)}$, the analoge of $r_{nn}^{(\infty)}$ in equation (2.28), is

$$
\begin{aligned}
r_{ll}^{(i-1)} \prod_{j=0}^{\infty} & \sqrt{\left(\frac{r_{qq}^{(i+j)}}{\sqrt{(r_{qq}^{(i+j)})^2 + P_1^2}}\right)^2 + \left(\frac{P_2}{r_{ll}^{(i-1)}}\right)^2} + \mathcal{O}(\epsilon^4) \\
&= r_{ll}^{(i-1)} \prod_{j=0}^{\infty} \sqrt{1 - \left(\frac{P_1}{r_{qq}^{(i+j)}}\right)^2 + \left(\frac{P_2}{r_{ll}^{(i-1)}}\right)^2} + \mathcal{O}(\epsilon^4) \\
&= r_{ll}^{(i-1)} \left(1 + \frac{1}{2} \sum_{j=0}^{\infty} \left[\frac{P_2^2}{(r_{ll}^{(i-1)})^2} - \frac{P_1^2}{(r_{qq}^{(i+j)})^2}\right]\right) + \mathcal{O}(\epsilon^4).
\end{aligned}
$$

Here we used $P_1 = \left(\prod_{a=i_0+1}^{i_0+j}(\lambda_l + \delta_a)\right) h_{l_1}$ and $P_2 = \left(\prod_{a=i_0+1}^{i_0+j}(\lambda_l + \delta_a)\right) h_{l_2}$. The proof then continues as in the other cases. (At the step where we bound, as in going from (2.29) to (2.30), letting each $\delta_a = \pm\epsilon$ and $r_{qq}^{(i+j)} = \sigma_q \pm \epsilon^2$, everything inside the sum in the numerator except for $(\lambda_l \pm \epsilon)^{2j+2}$ factors out and cancels with the identical terms in the denominator.) For other cases in which multiple elements share the slowest asymptotic convergence rate, the proofs are similar. □

Theorem 2.3.2 basically says that the asymptotic rates of convergence of the diagonal elements are the squares of the rates of the off-diagonal elements. More precisely, the rate of a diagonal element $r_{ss}^{(i)}$ is equal to the square of the slowest rate among all the off-diagonal elements in the union of row s and column s. So we now know the asymptotic convergence rates of the diagonal elements given the asymptotic convergence rates of the off-diagonal elements, to which we now turn.

We can use Mathias and Stewart's results again here. Recall that under the assumption (2.3) we have

$$\|R_{12}^{(i+1)}\| \leq \rho^{(i)} \|R_{12}^{(i)}\|.$$

For sufficiently large i, if $\sigma_k \neq \sigma_{k+1}$, then (2.3) holds. So we know that every element in the off-diagonal block $R_{12}^{(i)}$ must converge asymptotically with a rate $\rho \leq \sigma_{k+1}/\sigma_k$. So we have the following bound on the convergence of off-diagonal elements.

Lemma 2.3.3 *For $s < t$ let λ_{st} be the asymptotic rate of convergence of $r_{st}^{(i)}$. If $\sigma_s \neq \sigma_t$ then $\lambda_{st} \leq \sigma_t/\sigma_s$.*

Combining Lemma 2.3.3 and Theorem 2.3.2 gives us the following bound.

Theorem 2.3.4 *Let λ_{ss} be the asymptotic convergence rate of $r_{ss}^{(i)}$. If $\sigma_{s-1} > \sigma_s > \sigma_{s+1}$, then $\lambda_{ss} \leq \max((\sigma_s/\sigma_{s-1})^2, (\sigma_{s+1}/\sigma_s)^2)$.*

This bound is tight. In addition, there are many times when the convergence is much faster. Examples given later will illustrate both of these facts.

Having achieved a bound on the convergence of off-diagonal (and therefore diagonal) elements, we can make a few observations on precise asymptotic convergence rates.

We can easily see what happens when A is a 2-by-2 matrix with distinct singular values, for the passage from $(R^{(i)})^T$ to $R^{(i+1)}$ is illustrated by Figure 2.7 (just look at the first and last columns). Since we are in the asymptotic range, h is very small, and the diagonal elements a and b are close to σ_1 and σ_2. They experience $\mathcal{O}(h^2)$ perturbations and thus remain close to σ_1 and σ_2. If we denote the off-diagonal element of $R^{(i+1)}$ by h', then the rate of convergence is $h'/h =$

$b/\sqrt{a^2+h^2}$. This is an $\mathcal{O}(h^2)$ perturbation of $b/|a|$, so we see that the asymptotic rate of convergence is σ_2/σ_1. Note that if the singular values are equal, the convergence is painfully slow.

The elements in a general n-by-n matrix are experiencing two types of effects from the various Givens rotations, and these can be seen once again in Figure 2.7. Once each iteration, a given off-diagonal element x will be zeroed out. That is, it will play the role of h in Figure 2.7. Let us say that x here receives a *zeroing contribution* to its convergence. Typically several times each iteration, the element x will be affected by a Givens rotation though it is not being zeroed out. That is, it will play the role of c_s or d_s in Figure 2.7. Let us say that x is here receiving a *nonzeroing contribution.* (The only off-diagonal element never to receive a nonzeroing contribution to its convergence is $r_{1n}^{(i)}$. It plays the role of h precisely once, with $a = r_{11}^{(i)}$ and $b = r_{nn}^{(i)}$, and therefore its asymptotic rate of convergence is σ_n/σ_1, the reciprocal of the condition number.)

We already understand the effect of a zeroing contribution. Indeed, if only zeroing contributions were in play, then the situation in an n-by-n matrix would be a simple generalization of the 2-by-2 case: $r_{st}^{(i)}$ would asymptotically converge at a rate of σ_t/σ_s if $\sigma_s > \sigma_t$ and painfully slowly if the singular values were equal. We know this is not the case from empirical observation. First, repeated singular values tend to speed up convergence, and speed it up a great deal. See Figure 2.8. As we know that this is not coming from zeroing contributions, we have our first hint that nonzeroing contributions not only sometimes play a significant role but can be agents of very fast convergence. Second, nonzeroing contributions sometimes play a significant role (and bring about fast convergence) even in matrices with all distinct singular values.

To get a feel for how this fast convergence might happen, let us take a closer

$$\begin{pmatrix} \underset{\frac{1}{4}}{\mathbf{10}} & \frac{1}{2} & \frac{1}{50} & \frac{1}{10} \\ & \underset{\frac{1}{4}}{\mathbf{5}} & \frac{1}{25} & \frac{1}{5} \\ & & \underset{\frac{1}{25}}{\mathbf{5}} & \frac{1}{5} \\ & & & \underset{\frac{1}{25}}{\mathbf{1}} \end{pmatrix}$$

Figure 2.8: This is one observed convergence pattern for a matrix having singular values 10, 5, 5, and 1, written in bold on the diagonal for reference. In place of each off-diagonal element is its asymptotic convergence rate. Beneath each diagonal element is its rate. (Note that the diagonal convergence is as Theorem 2.3.2 states.) We know that nonzeroing contributions must be significant simply because of the repeated singular values. But look at how they speed up the convergence. The rates of the elements $r_{13}^{(i)}$ and $r_{23}^{(i)}$ are faster than the ratio of any two singular values in the matrix. We would expect $r_{13}^{(i)}$ to converge at a rate of $5/10 = 1/2$ were zeroing contributions dominating. Note that its (much) faster rate here allows the diagonal element $r_{33}^{(i)}$ to have a faster rate as well, $(1/5)^2$ instead of $(1/2)^2$.

look at a nonzeroing contribution. Say an off-diagonal element in question plays the role of c_s in Figure 2.7 under a certain Givens rotation. If we denote its value after this rotation is applied by c_s', then the convergence rate is

$$\frac{c_s'}{c_s} = \frac{ac_s - hd_s}{c_s\sqrt{a^2+h^2}} \approx \pm 1 + \frac{hd_s}{|a|c_s}, \tag{2.31}$$

where the sign is the sign of a and is different from the sign of the fraction. (The signs of the elements is a topic we will not pursue here.) Keeping in mind that a is large, being on the diagonal, we see that the convergence rate will be about ± 1 if h, c_s, and d_s are all about the same size. That is, the Givens rotation will not significantly affect c_s. But what if in equation (2.31), the element c_s is much smaller than h and d_s? Then the ratio $hd_s/|a|c_s$ is no longer close to zero, and

the contribution from this Givens rotation is significant.

So where there are disparities in the sizes of off-diagonal elements, nonzeroing contributions can come into play. To see how they can speed convergence, again consider the asymptotic convergence rate of c_s given by equation (2.31). For this rate to be constant, c_s must converge at the product of the rates of h and d_s. (The diagonal element a is large and constant for our purposes here.) This is what we seem to observe with, for example, repeated singular values. See Figure 2.9.

$$\begin{pmatrix} \sim 10 & x & x & x \\ & a \sim 5 & \mathbf{c} & 0 \\ & & \sim 5 & \\ & h & d & b \sim 1 \end{pmatrix} \qquad \begin{pmatrix} \mathbf{10}_{\frac{1}{4}} & \frac{1}{2} & \frac{1}{50} & \frac{1}{10} \\ & \mathbf{5}_{\frac{1}{4}} & \frac{1}{25} & \frac{1}{5} \\ & & \mathbf{5}_{\frac{1}{25}} & \frac{1}{5} \\ & & & \mathbf{1}_{\frac{1}{25}} \end{pmatrix}$$

Figure 2.9: The example of Figure 2.8 revisited. In the matrix on the left, the last Givens rotation applied gave the element $r_{23}^{(i)}$ (now marked $\mathbf{c}$) almost no zeroing contribution to its convergence because of the repeated singular values 5 and 5. Now the element marked h is being zeroed, and $\mathbf{c}$ will receive a nonzeroing contribution, because in this example it is very small compared to d and h. From the discussion, $\mathbf{c}$ will converge at the product of the rates of h and d. In this example, h and d are being affected by primarily zeroing contributions, so they are each converging at the rate of $1/5$. So $\mathbf{c}$ should be converging at the rate of their product, $1/25$. The matrix on the right, which shows the rates again for reference, confirms this.

Before we state an observation concerning the asymptotic rate of convergence for off-diagonal elements, let us set some notation to handle repeated singular values. Say that $\sigma_k > \sigma_{k+1} = \cdots = \sigma_{k+p} > \sigma_{k+p+1}$. Set $\rho_{k+1}^{(u)} = \cdots = \rho_{k+p}^{(u)} \equiv \sigma_k/\sigma_{k+1}$, and set $\rho_{k+1}^{(l)} = \cdots = \rho_{k+p}^{(l)} \equiv \sigma_{k+p}/\sigma_{k+p+1}$.

We already know from Lemma 2.3.3 that if $s < t$, then the slowest the asymptotic rate of convergence of $r_{st}^{(i)}$ can be is σ_s/σ_t, assuming the singular values are not equal. For the general case, we here catalogue some of the possible asymp-

totic rates of convergence for the off-diagonal elements based on our empirical observations.

1. If neither σ_s nor σ_t is a repeated singular value, then $r_{st}^{(i)}$ often converges at the rate σ_s/σ_t, but can behave as if σ_s and/or σ_t were repeated, as described in the following.

2. If $\sigma_s = \sigma_t$ then $r_{st}^{(i)}$ often converges at either the rate $(\rho_s^{(u)})^2$ or the rate $(\rho_s^{(l)})^2$. Actually, a sequence of j equal singular values defines a j-by-j block, all of whose off-diagonal elements converge at the same rate.

3. If $\sigma_s \neq \sigma_t$ and one or both are repeated singular values, then $r_{st}^{(i)}$ often converges at the rate σ_s/σ_t, $(\rho_s^{(u)})^2\sigma_s/\sigma_t$, $(\rho_s^{(l)})^2\sigma_s/\sigma_t$, $(\rho_t^{(u)})^2\sigma_s/\sigma_t$, or $(\rho_t^{(l)})^2\sigma_s/\sigma_t$.

In Figure 2.10 are the asymptotic convergence rates for a few example matrices with given singular values, and (1) through (3) above are represented.

In the 3-by-3 matrix, off-diagonal elements receive only zeroing contributions, so the asymptotic convergence rate of $r_{st}^{(i)}$ is σ_t/σ_s. This is not so for the 6-by-6 matrix. Although it has all singular values distinct, the element $r_{45}^{(i)}$ is receiving nonzeroing contributions, converging at a rate of $1/50 \ll 1/2 = 5/10 = \sigma_5/\sigma_4$. The three 4-by-4 matrices have the same singular values, 10, 5, 5, and 1, but converge differently. From these three matrices, we see that asymptotic (and not just initial) convergence is dependent not only on the singular values but also on the entries in the original matrix. It is also dependent on whether pivoting is used or not. In the examples we looked at, pivoting on the first step of the QLP iteration tended to give faster asymptotic convergence rates than not pivoting. (For example, for 4-by-4 matrices having singular values 10, 5, 5, and 1, the rates on the right are far more likely to occur when pivoting is used on the first step.

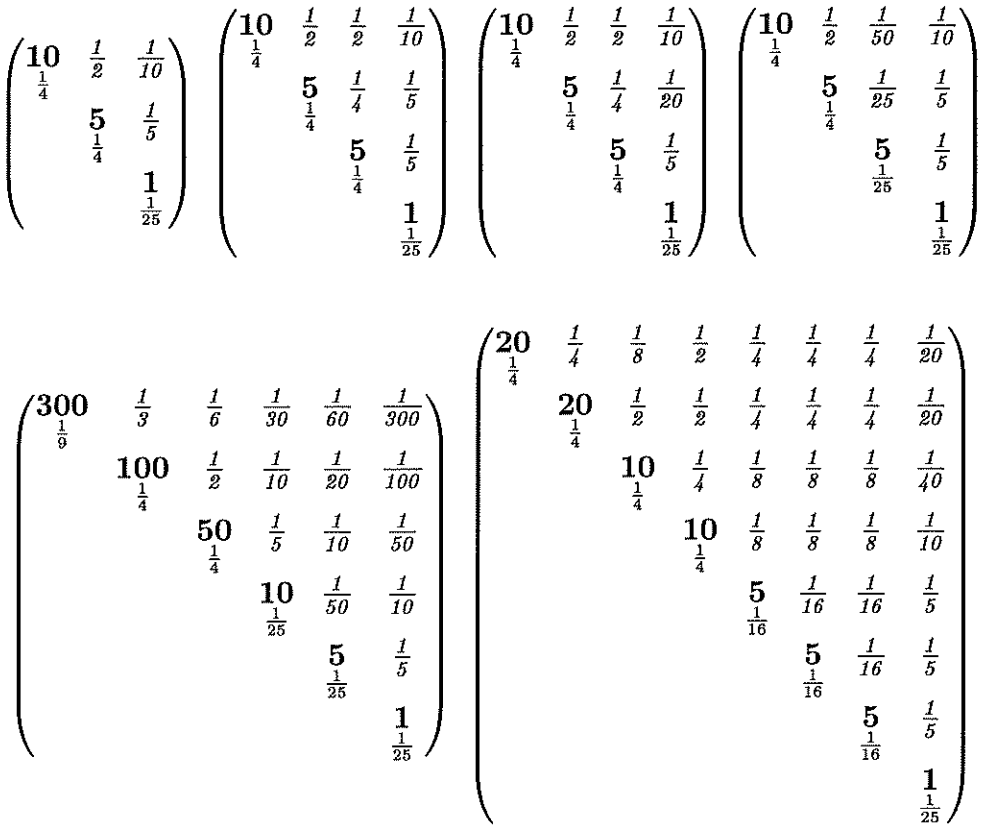

Figure 2.10: Here are six sample matrices with their singular values given in bold on the diagonal. Off-diagonal elements are replaced by their asymptotic convergence rates in italics, and diagonal elements have their rates written below them. Concerning the off-diagonal rates, see the text for some discussion and also notice that each of (1) through (3) in the catalogue on page 44 is represented. Concerning the diagonal rates, notice that Theorem 2.3.2 is everywhere verified: the rate of $r_{ss}^{(i)}$ is equal to the square of the slowest (i.e., largest) rate among all the off-diagonal elements in row s and column s.

Recall that it is discrepancies in the sizes of off diagonal elements that bring into play nonzeroing contributions and therefore faster convergence. Pivoting seems to encourage this.)

From studying the asymptotic convergence rates, we now have two insights into how the convergence of individual elements in the pivoted QLP decomposition can be fast in the presence of repeated singular values. One, in the presence of repeated singular values, convergence tends to be faster than with distinct singular values. Significant nonzeroing contributions cause the faster convergence, and they are necessarily a part of the landscape when repeated singular values are around. They are optional (and indeed less common, at least in our observations) when singular values are distinct. Although these are only asymptotic results, the pivoting can often speed up the approach of the asymptotic range.

Two, recall that nonzeroing contributions are caused by large discrepancies in the sizes of off-diagonal elements and that pivoting tends to encourage this. The discrepancies are thus starting to form in the first step, and where there are discrepancies, there are some small off-diagonal elements already.

These observations apply not only when there are repeated singular values but also when the ratios between neighboring singular values are close to one. Indeed, prior to reaching the asymptotic range, the QLP iteration cannot distinguish identical singular values from those that are merely close. The slow convergence in the latter case is an asymptotic phenomenon. Early on the convergence is fast.

It is also interesting to note that when faster convergence occurs, the convergence of the matrices of singular vectors of the $R^{(i)}$ is also faster. For example, Chadrasekaran and Ipsen [CI95] show that the convergence to zero of the angle between the first k columns of $U^{(i)}$ (and also of $V^{(i)}$) and the first k columns of the identity matrix is bounded by σ_{k+1}/σ_k. We would like to point out that

faster rates are attainable when nonzeroing contributions are at work. For example, in the 6-by-6 matrix in Figure 2.10, we noticed that the element in the (4,5) position is converging fast, at a rate of 1/50 instead of at the "expected" rate of $\sigma_5/\sigma_4 = 5/10 = 1/2$. Correspondingly, the first four columns of $U^{(i)}$ and of $V^{(i)}$ are converging not at a rate of 1/2 but at a rate of 1/10.

Finally, we note that there are patterns in the asymptotic convergence of individual off-diagonal elements of the $U^{(i)}$ and $V^{(i)}$ similar to the patterns seen in the $R^{(i)}$.

CHAPTER 3

Truncating and Interleaving

3.1 QLP Junior

Many applications involve a large m-by-n matrix A with numerical low rank, that is, with a few significant singular values and the rest close to zero. Once again, the Singular Value Decomposition of A is given by $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times m}$ is the matrix of left singular vectors, $V \in \mathbb{R}^{n\times n}$ is the matrix of right singular vectors, and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix containing the singular values. We would like to have a rank-k approximation of A which ignores the contributions of the small singular values. It is well known that the best such approximation to A is obtained by simply truncating the SVD: $A \approx A_k = U_k\Sigma_k V_k^T$, where U_k is m-by-k, Σ_k is k-by-k, and V_k is k-by-n (see equation (1.1) in Chapter 1). The columns of U_k span what is called the *left superior singular subspace* and the columns of V_k the *right superior singular subspace.*

Just as an approximation to the full SVD of a matrix $A = U\Sigma V^T$ will provide approximations to the singular values and to the left and right singular subspaces (spanned by the columns of U and V, respectively), an approximation to the truncated SVD, $A_k = U_k\Sigma_k V_k^T$, must provide approximations to the first k singular values and to the left and right superior singular subspaces.

We saw in Chapter 2 that the QLP decomposition is an approximation to the full SVD, with $Q \approx U$, $L \approx \Sigma$, and $P \approx V$. Partition the matrices of the

decomposition as follows:

$$QLP^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} P_1^T \\ P_2^T \end{pmatrix}, \tag{3.1}$$

where Q_1 and P_1 both have k columns and L_{11} is k-by-k. Then $Q_1 L_{11} P_1^T$ is an approximate truncated SVD. One question is whether these three matrices are obtainable without having to compute the entire QLP decomposition. Stewart [Ste99] pointed out that because pivoting is not required in the second step (taking the QR factorization of R to obtain L), it is possible to truncate the QLP decomposition.

To see how truncation is possible, assume the matrix A has numerical rank r, and that we would like to obtain the r largest singular values and their associated superior singular subspaces. We begin as we would in computing the full QLP decomposition by computing the pivoted QR factorization of A. Say we have computed the first r columns of Q and thus the first r rows of R. But these rows of R are exactly what is needed to compute the first r columns of L, assuming we use no pivoting in obtaining L. See Figure 3.1.

Note that pivoting is not inhibited in this second step, that is, in triangularizing this set of r columns of R^T to obtain L^T. It is just that pivoting is confined to these r columns, for the remaining columns of R^T were not computed. If pivoting were a crucial part of this second step, truncation would be impossible, for any of the n columns of R^T might need to be pivoted to the first column or the second column or so on. Because pivoting on the second step produces only minor, if any, changes from not pivoting, we can truncate with impunity after computing only the first r rows of R. Note that in thus computing the first r rows of R, we

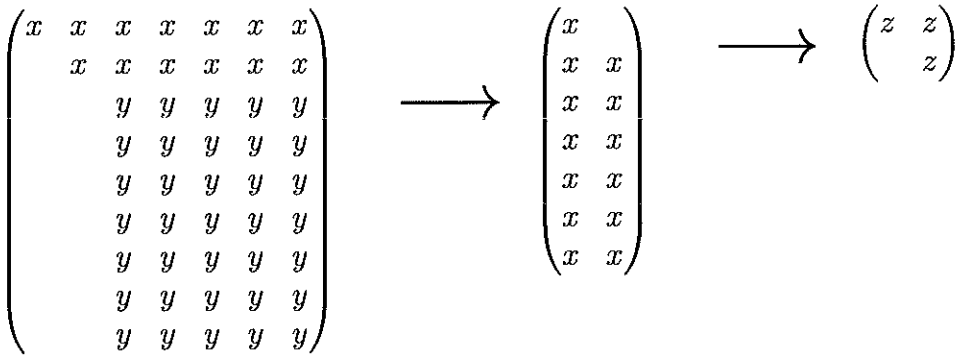

Figure 3.1: An example of truncating with $k = 2$. On the left, the 'x's represent $[R_{11}^{(0)} R_{12}^{(0)}]$. We take the transpose of this matrix (see center), and perform the QR factorization, obtaining the 2-by-2 matrix $R^{(1)}$ on the right. Note that none of the elements marked with a 'y' are needed to perform the computation.

have obtained a basis for the left superior singular subspace, and in computing the first r columns of L, we have obtained a basis for the right superior singular subspace.

Along the same lines, Stewart pointed out that the algorithm can be interleaved. This means computing the first k_1 rows of R and from them computing the first k_1 columns of L^T. Then computing the next k_2 rows of R and from them computing the next k_2 columns of L^T. And so on. This can be continued until the entire full QLP decomposition has been computed. It works for the same reason that truncating works: pivoting on the second step is not crucial. Note that pivoting is possible within each set of k_i vectors.

Interleaving is a useful tool because of the nature of the R-values and L-values. If r is the numerical rank, we know there is a large gap in the singular values, $\sigma_{r+1}/\sigma_r \ll 1$. Consider the pivoted QR factorization of A, $A\Pi = QR$. The elements on the diagonal of R, the R-values, often indicate the presence of a gap in the singular values. As a check, we can take the (unpivoted) QR factorization of R^T, $R^T = PL^T$. We know that the L-values provide excellent approximations

to the singular values and that, in particular, gaps in the singular values are clearly revealed. So the R-values reveal a potential gap, and the L-values provide confirmation.

So for a low-rank problem, interleaving can work in tandem with truncating in the following way. Say we compute k_1 rows of R until we see a gap in the R-values. (So if the potential gap in the singular values is between σ_r and σ_{r+1}, then k_1 could equal $r+1$.) These k_1 rows of R are all that are needed to compute the first k_1 columns of L (and hence the first k_1 L-values). Computing these L-values, we either confirm the gap and truncate the QLP decomposition here, or we fail to confirm the gap and hence repeat the process, calculating the next k_2 rows of R until another potential gap is revealed, which can then be checked by computing the corresponding k_2 L-values. Once a gap has been confirmed by the L-values, we are finished—the truncated QLP decomposition for this low-rank problem has been computed.

3.2 Convergence

We now assess the ability of the truncated pivoted QLP decomposition to capture the singular values. If the full pivoted QLP decomposition is $A = QLP^T$, let R and L be partitioned as usual:

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \qquad L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix},$$

where R_{11} and L_{11} are both k-by-k. As before, we make the assumption that the initial pivoting reveals the rank in the sense that

$$\|R_{22}\| \leq \sqrt{(k+1)(n-k)}\,\sigma_{k+1}, \tag{3.2}$$

$$\inf(R_{11}) \geq \frac{\sigma_k}{\sqrt{k(n-k+1)}}. \tag{3.3}$$

The pivoted QR factorization usually does this. The bounds (3.2) and (3.3) are for an RRQR algorithm by Chandrasekaran and Ipsen [CI94]. We also make the assumption that $\rho = \frac{\|R_{22}\|}{\inf(R_{11})} < 1$, which is almost always true if the gap between σ_k and σ_{k+1} is substantial, which is the situation when dealing with low-rank problems, for example.

We have the following theorem.

Theorem 3.2.1 *Let A be an m-by-n matrix and let $\sigma_k(A) > \sigma_{k+1}(A)$, $k < n$. Let $(R_{11}R_{12})$ be the first k rows of the R-factor in the pivoted QR factorization of A. Let L_{11}^T be the R-factor in the unpivoted QR factorization of $(R_{11}^T R_{12}^T)^T$. Assume that the bounds (3.2) and (3.3) hold and that $\rho = \|R_{22}\|/inf(R_{11}) < 1$.*

Then for $j = 1, \ldots, k$,

$$\frac{\sigma_j(L_{11})^{-1} - \sigma_j^{-1}}{\sigma_j^{-1}} \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^2 \mathcal{O}\left(\frac{n^{\frac{5}{2}}\|R_{12}\|^2}{(1-\rho^2)[\inf(R_{11})]^2}\right). \tag{3.4}$$

Proof:

Since there is no pivoting in the second step, the L_{11} produced by truncating the QLP decomposition is the same as the L_{11} produced by computing the entire QLP decomposition. Hence, the result follows from Theorem 2.2.3 of Chapter 2. □

Theorem 3.2.1 shows that when we have a low-rank problem and perform a truncated QLP decomposition, the relative error $\frac{\sigma_j(L_{11})^{-1}-\sigma_j^{-1}}{\sigma_j^{-1}}$ is quadratic in the gap ratio σ_{k+1}/σ_k for each $j = 1, \ldots, k$. All of the singular values of interest are well approximated.

3.3 Operation Count

In addition to being accurate, the truncated pivoted QLP algorithm is fast when k is small. To QR factorize k columns of an m-by-n matrix using Householder transformations requires about $2mnk - k^2(m+n) + 2/3k^3$ flam. Note that pivoting adds only an $\mathcal{O}(mn)$ term, so we need not distinguish when we are pivoting. (A flam is a floating point addition combined with a floating point multiplication, a very common operation in linear algebra. Stewart [Ste98, p. 96] prefers using this term to report many operation counts, because the term "flop" has changed meaning multiple times and is therefore now ambiguous.)

Computing the complete QLP factorization requires computing all n columns of the QR factorization of the m-by-n matrix A and computing all n columns of the QR factorizatoin of the n-by-n matrix R^T. This gives a total flam count of

$$2mn^2 - n^2(m+n) + \frac{2}{3}n^3 + 2n^3 - n^2(n+n) + \frac{2}{3}n^3 = mn^2 + \frac{1}{3}n^3.$$

Computing the truncated QLP factorization, in which we compute only the first r rows of R, requires computing the first r columns of the QR factorization of the m-by-n matrix A and computing all r columns of the QR factorization of the n-by-r matrix $(R_{11}^T R_{12}^T)^T$. This gives a total flam count of

$$2mnr - r^2(m+n) + \frac{2}{3}r^3 + 2nr^2 - r^2(n+r) + \frac{2}{3}r^3 = 2mnr - mr^2 + \frac{1}{3}r^3.$$

If r is small ($r \ll n$), then the $\mathcal{O}(mnr)$ time it takes to compute the truncated QLP is small compared to the $\mathcal{O}(mn^2)$ time for the full QLP decomposition. This can be a huge savings. Providing SVD-quality information, the truncated QLP decomposition is a real bargain.

CHAPTER 4

Estimating Condition Number

4.1 Background

Say we have some linear algebra algorithm. We would like to know whether the algorithm is such that given two inputs that are close, the algorithm produces two outputs that are close. An algorithm is labeled "stable" or "unstable" based on whether it has this property.

Unstable algorithms are usually worthless, and it is easy to see why. When we give the algorithm data so that it can give us an output from the data, that data will almost never be the exact data for the problem we are trying to solve. For example, suppose the problem is one in which pressure depends on temperature. We feed the algorithm some temperatures that we have measured, and it spits out a pressure. Our temperature measurements will certainly be off by a little bit—aside from human error, measurements can only have a certain instrument dependent precision. So we are not giving the algorithm the actual temperatures but only numbers that are close. Uh, oh. The unstable algorithm can therefore spit out a pressure that is far from what it would have spit out had we fed it the actual temperatures; that is, it can output a pressure which is not close to, but far from the actual pressure. Even assuming that our measurements were exact, the number line of the real world is continuous, whereas the number line of a computer is necessarily discrete, a computer being able to handle only a finite number of

possibilities on the number line. So even perfect data is misrepresented before the algorithm even starts. The data is off by a little, and an unstable algorithm can therefore make the answer it gives off by a lot.

We employ an algorithm to solve a certain problem we have, and we want the algorithm to be stable for reasons that we have seen. Before we even get to the algorithm, however, there is the question of the problem itself. Perhaps the problem is such that if the data is off by a little, the answer can be off by a lot, even if a stable algorithm is used to solve it. We say that a problem is "well-conditioned" or "ill-conditioned" depending on whether is has this property. An ill-conditioned problem is extrememly hard to solve, because even a stable algorithm can give answers that are way off. The difficulty is not in the algorithm but in the problem itself.

In this chapter we are concerned with the condition of a problem. In particular, we assume that the problem is represented by a matrix, and the question becomes whether the matrix is well-conditioned. There are different types of problems involving matrices, such as solving linear systems and computing eigenvalues. In all such problems we are concerned with how perturbations in the data will potentially affect the computed solution. In solving linear systems, a couple of bounds are [CR83]:

1. if $Ax = b$ and $A(x + \Delta x) = b + \Delta b$ then

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\|\frac{\|\Delta b\|}{\|b\|},$$

2. if $Ax = b$ and $(A + \Delta A)(x + \Delta x) = b$, then

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A\|\|A^{-1}\|\|\Delta A\|/\|A\|}{1 - \|A\|\|A^{-1}\|\|\Delta A\|/\|A\|},$$

provided $\|\Delta A\| < 1/\|A^{-1}\|$. The vector norm in the inequalties may be any that is consistent with the matrix norm.

Note the quantity $\|A\|\|A^{-1}\|$. This quantity plays a determinative role in the above bounds and in many other bounds like them. This number thus indicates the condition of a matrix, and we call the quantity

$$\kappa(A) = \|A\|\|A^{-1}\|$$

the *condition number* of A (for some matrix norm). Note that since

$$\|A\|_2\|A^{-1}\|_2 = \sigma_1\frac{1}{\sigma_n},$$

the Singular Value Decomposition can be used to determine the condition number. Of course, the SVD is very expensive to compute. As many algorithms which operate on a full matrix are $\mathcal{O}(n^3)$, the goal is to have a condition estimator which is $\mathcal{O}(n^2)$, negligible compared to whatever else is being done to the matrix to solve the problem at hand.

The usual assumption when trying to determine the condition number of a matrix A is that a factorization of A is available, and one of the factors is triangular, for example, $A = LU$ or $A = QR$. In fact, in many applications where

a condition number estimate is required, the matrix is already thus factored or is triangular to begin with [Hig87]. So the "general problem" is that of estimating the condition number of a triangular matrix.

When we say "estimating" the condition number, we mean just that. As the condition number is used to determine whether the given problem is well-conditioned, that is, whether it is sensitive to perturbations, the exact condition number is not needed, but only an order-of-magnitude estimate. (Note that this is not true for all applications, however. [HT00]) Note that $\kappa(A) \geq 1$. A condition number on the order of 10^2, say, indicates a well-conditioned problem, whereas a condition number on the order of 10^{14} indicates ill-conditioning and that special care will be needed to solve the problem accurately.

Most of the condition estimators that have been in software libraries and packages, such LINPACK, LAPACK, and Matlab, have used the 1-norm, though there are 2-norm condition estimators as well. For a short history of condition estimation, see the first section of [HT00], in which the authors also introduce an improvement on the current LAPACK estimator. For a nice comparison of condition estimators, including 2-norm estimators, see the survey [Hig87]. In the latter paper, Higham, who has worked much in this area over the past two decades, gives the rating of "reliable" to almost all of the estimators surveyed. This even includes simply taking $\kappa(A) \approx |r_{11}|/|r_{nn}|$, where r_{11} and r_{nn} are the upper-left and lower-right diagonal elements in the R factor of the pivoted QR factorization.

Higham and Tisseur point out that estimating condition number "appears inevitably to admit the possibility of arbitrarily poor estimates (although proving so is an open problem [Dem92]." So an estimator is considered useful if it performs well in practice, and especially if there are some kind of bounds that go with it.

Since the QLP decomposition computes approximate singular values, it makes sense to consider how it can be used to estimate the 2-norm condition number. The QLP decomposition does not fit exactly into the mold of a typical condition estimator. After the first (pivoted) QR factorization is performed, we are at the stage where we would want to apply an $\mathcal{O}(n^2)$ method to estimate the condition of R (and therefore of A, since Q is orthogonal). But the second step of the QLP decomposition, namely computing the QR facrtorization of R^T, requires $\mathcal{O}(n^3)$ operations.

We therefore approach QLP's ability to estimate condition number from two perspectives. First we recognize the fact that sometimes we will want to be computing the QLP decomposition in order to obtain an approximate SVD for some purpose. To receive a good condition number estimate for free (especially one with bounds) would be a nice boon. Also, consider the case when $n \ll m$ (as is often the case in least-squares problems), so that doing the second step on the n-by-n matrix R^T is relatively cheap. Then it would make sense to do the entire QLP to get a condition number estimate. (Note that if n is extremely small, however, we can just compute the SVD of R. So we are thinking of the case when n is small, but not too small.) So we derive bounds which apply when the entire QLP is performed.

Our other approach is to look at various ways of estimating condition which lie somewhere between merely doing the first QR factorization and going all the way through with the second one. Some of these techniques, observations, and bounds could be useful in some situations.

4.2 Doing the Full QLP

To estimate the condition number after performing the full pivoted QLP decomposition, we take $\kappa(A) = \sigma_1/\sigma_n \approx |r_{11}^{(1)}|/|r_{nn}^{(1)}|$. We would like to derive a bound on this estimate. Actually we will derive a bound on the underapproximation ratio, the ratio of the QLP condition number estimate to the actual condition number. Again from Mathias and Stewart [MS93] we know that if

$$\rho_{11}^{(1)} = \frac{\|R^{(1)}(2:n, 2:n)\|}{|r_{11}^{(1)}|} < 1, \tag{4.1}$$

then

$$\frac{|r_{11}^{(1)}|}{\sigma_1} \geq \left(1 - \frac{\|R^{(1)}(1, 2:n)\|^2}{[1 - (\rho_{11}^{(1)})^2](r_{11}^{(1)})^2}\right)^{1/2}. \tag{4.2}$$

Here the matrix $R^{(1)}$ is partioned as usual with $k = 1$ to isolate the $r_{11}^{(1)}$ element used to approximate the 2-norm. If we instead partition with $k = n - 1$, we know that if

$$\rho_{nn}^{(1)} = \frac{|r_{nn}^{(1)}|}{\inf(R^{(1)}(1:n-1, 1:n-1))} < 1, \tag{4.3}$$

then

$$\frac{\sigma_n}{|r_{nn}^{(1)}|} \geq \left(1 - \frac{\|R^{(1)}(2:n,n)\|^2}{[1-(\rho_{nn}^{(1)})^2]\inf(R^{(1)}(1:n-1,1:n-1))^2}\right)^{1/2}. \quad (4.4)$$

Both assumptions are very reasonable after the two steps of the pivoted QLP decomposition. A resulting bound on the underapproximation ratio is thus the product of the above bounds:

$$\frac{|r_{11}^{(1)}|/|r_{nn}^{(1)}|}{\sigma_1/\sigma_n} \geq \left(1 - \frac{\|R^{(1)}(1,2:n)\|^2}{[1-(\rho_{11}^{(1)})^2](r_{11}^{(1)})^2}\right)^{1/2} \times \left(1 - \frac{\|R^{(1)}(2:n,n)\|^2}{[1-(\rho_{nn}^{(1)})^2]\inf(R^{(1)}(1:n-1,1:n-1))^2}\right)^{1/2} \quad (4.5)$$

Note that all of the quantities in the bound (4.5) are readily available, if we make the approximations $\|R^{(1)}(2:n,2:n)\| \approx |r_{22}^{(1)}|$, and $\inf(R^{(1)}(1:n-1,1:n-1) \approx |r_{n-1,n-1}^{(1)}|$. The bound is clearly best when there are sizeable gaps between σ_1 and σ_2 and between σ_{n-1} and σ_n. Note that (4.2) can also be used as a bound on the error in using Stewart's 2-norm estimator [Ste99].

4.3 Tricks of the Trade

In this section, we do not carry the pivoted QLP decomposition all the way out. We instead start with $R^{(0)}$, the upper-triangular factor after the first pivoted QR, just like other condition estimators. Using what we know about the QLP decomposition, we want to see how much we can improve the basic pivoted QR condition estimate $|r_{11}|/|r_{nn}|$ by performing $\mathcal{O}(n^2)$ or even $\mathcal{O}(n)$ operations. Our first two items deal with improving the estimate of σ_1 and our last two items deal

with improving the estimate of σ_n, the much harder of the two to get a handle on.

4.3.1 QLP Quality for σ_1 without QLP

Our first item is simply to notice that the upper-left diagonal element $r_{11}^{(1)}$ obtained after performing the full two-step pivoted QLP decomposition is simply the norm of the first row of $R^{(0)}$, assuming no pivoting is done on the second step. In fact, noticing that the norm of this row is a better approximation to σ_1 than $r_{11}^{(0)}$ was what led Stewart to the QLP decomposition in the first place. So one possibility, given $R^{(0)}$, is to take $\sigma_n \approx |r_{nn}^{(0)}|$ and $\sigma_1 \approx |r_{11}^{(1)}| = \|R^{(0)}(1, 1:n)\|$. We thus get the bound (4.2) for only $\mathcal{O}(n)$ operations.

Of course, the bound (4.4) no longer holds. Instead we have only the RRQR bound $\sigma_n/|r_{nn}^{(0)}| \geq 1/\sqrt{n}$. This bound is usually true for QR with column pivoting, but to be sure an RRQR method guaranteeing this bound should be used (see Section 4.3.2). The bound can be useful when n is not too big and rough order-of-magnitude accuracy is all that is required.

Our second item is to point out Stewart's two norm estimator [Ste99], based on truncating the pivoted QLP decomposition. Instead of computing only $r_{11}^{(1)}$ and assuming that no pivoting is done on the second step, Stewart pivots on the second step, and computes the first few $r_{jj}^{(1)}$. He suggests using the first two or three rows of $R^{(0)}$ or using rows 1 through k, where going from row k to row $k+1$ is the first time a decrease in norm occurs. Either criterion is likely to capture the 2-norm. (Actually, Stewart was considering the case of a full matrix A before it has been QR factored the first time, and he has an efficient implementation for the 2-norm estimator in this case. We have simply described what it would look like in our simpler situation.)

Computing the first diagonal element of $R^{(1)}$ using pivoting clearly gives at least as good and perhaps a better estimate of σ_1 than not pivoting. Moreover, if we truncate at a significant gap, we expect the relative error to be small, as it depends on the square of the gap ratio (see Theorem 3.2.1 of Chapter 3).

4.3.2 That Elusive σ_n

The first of our two items in this section is simply to suggest that to possibly improve the quality of $|r_{nn}^{(0)}|$ and to ensure the bound $\sigma_n/|r_{nn}^{(0)}| \geq 1/\sqrt{n}$, we should use an RRQR algorithm that guarantees this bound. This can be useful no matter which of the two methods in Section 4.3.1 is used.

The entire RRQR algorithm need not be carried out, of course. We need only determine the permutation which puts the correct column of $R^{(0)}$ into the n-th column. Implemented correctly, determining the permutation and getting the correct column into the n-th position with the new $R^{(0)}$ of course retriangulated requires $\mathcal{O}(n^2)$ operations [Cha87].

The second item gives QLP-quality information about σ_n when it works. Once again recall that Mathias and Stewart [MS93] show that if

$$\rho = \frac{|r_{nn}^{(0)}|}{\inf(R^{(0)}(1:n-1,1:n-1))} < 1,$$

then

$$\|R^{(1)}(1:n-1,n)\| \leq \rho \|R^{(0)}(1:n-1,n)\|.$$

Now we know that the norm of the n-th column of $R^{(1)}$ is just $|r_{nn}^{(0)}|$, that is

$$(r_{nn}^{(0)})^2 = (r_{nn}^{(1)})^2 + \|R^{(1)}(1:n-1,n)\|^2.$$

Hence

$$\begin{aligned}(r_{nn}^{(1)})^2 &= (r_{nn}^{(0)})^2 - \|R^{(1)}(1:n-1,n)\|^2 \\ &\geq (r_{nn}^{(0)})^2 - \rho^2\|R^{(0)}(1:n-1,n)\|^2 \\ &\approx (r_{nn}^{(0)})^2 - \left(\frac{r_{nn}^{(0)}}{r_{n-1,n-1}^{(0)}}\right)^2 \|R^{(0)}(1:n-1,n)\|^2.\end{aligned}$$

These quantities are readily available. If the right hand side is a positive number, we can take its square root as a lower bound for $r_{nn}^{(1)}$ without having to do anything except make this $\mathcal{O}(n)$ calculation. Since we then know that we are not overestimating $r_{nn}^{(1)}$, we can use this quantity as an approximation of σ_n in a condition number estimate. As an approximation of σ_1 we can take $|r_{11}^{(1)}|$ by just taking the norm of the first row of $R^{(0)}$ (see the previous section). Then we have a good condition number estimate whose underapproximation ratio satisfies the bound 4.5 (though we cannot compute the bound).

So for only $\mathcal{O}(n)$ work, we get a QLP-quality condition number estimate. Note that the right hand side of the inequality (4.6) has the best chance of being positive when the gap ratio $r_{nn}^{(0)}/r_{n-1,n-1}^{(0)}$ is small. This is when the decrease from $r_{nn}^{(0)}$ to $r_{nn}^{(1)}$ tends to be the greatest and is hence when we are the happiest to be able to get a handle on $r_{nn}^{(1)}$.

4.4 Numerical Experiments

We now present some numerical experiments. We will test the performance of the pivoted QLP decomposition and of the QR/QLP hybrid which takes $|r_{nn}^{(0)}|$ as the approximation of σ_n and $|r_{11}^{(1)}|$ as the approximation of σ_1 (by simply taking the norm of the first row of $R^{(0)}$). Let us call this latter method the QRplus method, as it is just QR plus computing one row norm, inspired by QLP.

We perform three tests on these two condition estimators, essentially following Higham [Hig87] (and also Higham and Tisseur [HT00]) in the details. We run each test for square matrices of size n, and let $n = 10$, $n = 25$, and then $n = 50$. In each test and for each n, we do fifty runs, calculating the minimum underapproximation ratio and the average underapproximation ratio. That is, since we start with the R factor from a pivoted QR factorization, we report the minimum value and the average value of $\mathrm{EST}/\kappa_2(R)$, where EST is the estimate of the condition number.

In test 1, we start with a matrix A having elements from the uniform distribution on $[0, 1]$. We then take the pivoted QR factorization and run the tests on R.

In test 2, we create a diagonal matrix Σ having the singular values

$$\sigma_i = \alpha^i, \qquad 1 \leq i \leq n,$$

letting the condition number vary: first 10, then 10^3, then 10^6, and finally 10^9. We then produce random orthogonal U and V and set A to be $U\Sigma V^T$, take the pivoted QR factorization and begin the tests.

n	10	25	50
QLP	.80/.91	.76/.89	.77/.87
QRplus	.31/.55	.24/.37	.16/.29

Table 4.1: Test 1 with minimum/average underapproximation ratio.

κ_2	$n = 10$	25	50
10	.73/.89	.87/.96	.94/.98
10^3	.81/.97	.75/.97	.87/.99
10^6	.94/.99	.78/.98	.75/.99
10^9	.99/1.0	.78/.99	.70/.99

Table 4.2: Test 2 for QLP.

Test 3 is just like test 2, except that the singular values are

$$1 = \sigma_1 = \sigma_2 = \cdots = \sigma_{n-1} > \sigma_n = \|A^{-1}\|_2^{-1}.$$

The results are presented in Tables 4.1 through 4.5. The underapproximation ratios are given in the form minimum/average in each table.

Note that in all cases order-of-magnitude accuracy is attained. The worst results are with the random matrices in test 1. Looking at tests 2 and 3 we see

κ_2	$n = 10$	25	50
10	.53/.74	.73/.88	.85/.94
10^3	.41/.71	.64/.85	.79/.93
10^6	.37/.70	.49/.86	.71/.92
10^9	.37/.70	.35/.87	.41/.90

Table 4.3: Test 2 for QRplus.

κ_2	$n = 10$	25	50
10	.98/.99	.99/1.0	1.0/1.0
10^3	1.0/1.0	1.0/1.0	1.0/1.0
10^6	1.0/1.0	1.0/1.0	1.0/1.0
10^9	1.0/1.0	1.0/1.0	1.0/1.0

Table 4.4: Test 3 for QLP.

κ_2	$n = 10$	25	50
10	.47/.75	.61/.87	.74/.93
10^3	.49/.72	.53/.88	.74/.93
10^6	.45/.73	.54/.87	.76/.93
10^9	.51/.74	.51/.88	.81/.93

Table 4.5: Test 3 for QRplus.

that for either method and a fixed n, the values are roughly the same in the entire column. Moreover this holds true over both tests. Note the spectacular success of QLP in test 3, especially as compared with the performance of QRplus. (As only 2 decimals are recorded, a value of 1.0 indicates at least two significant digits in the condition number estimate.) Test 3 has all singular values equal to 1 except for a small σ_n. This is a fine illustration of the converging power of the two-step QLP decomposition. Note that since there is a large gap between σ_{n-1} and σ_n, this is a case where the trick at the end of Section 4.3.2 has a good chance of working. It sure would be nice to get the results of Table 4.4 essentially for free.

The results for both tests are comparable to those obtained by Higham [Hig87] for the various estimators he tested. We conclude that if we have computed a pivoted QLP decomposition, the approximations of σ_1 and σ_n are good enough to give order-of-magnitude estimates of the condition number, and sometimes correct digits in the estimate. It has been common practice just to use the given

pivoted QR factorization by itself to obtain a rough condition number estimate. Taking the norm of the first row of R improves the estimate of σ_1 and appears to produce acceptable condition number estimates.

CHAPTER 5

Latent Semantic Indexing

5.1 Querying a Matrix

Our problem in this chapter is that of representing a database of text documents and then querying the database. Say our database consists of a set of d documents, each containing some text. To each document we assign a vector, each component of which reflects an idea or word associated with the document. So the vector would have length t, the number of terms indicating ideas or words in the database. Each entry in the vector would be one or zero depending on whether the document contains or is associated with that term. More often, instead of ones, weights would be assigned indicating the importance of the term in that document. Having determined the vector for each document in the database, we can represent the database as a t-by-d *term-by-document matrix*. (Much of the discussion of this chapter comes from [BDJ99] and [BB99].)

Consider the following example from [BDJ99], with the given $t = 6$ terms and $d = 5$ books:

T1: bak(e,ing)

T2: recipes

T3: bread

T4: cake

T5: pastr(y,ies)

T6: pie

D1: How to Bake Bread Without Recipes

D2: The Classic Art of Viennese Pastry

D3: Numerical Recipes: The Art of Scientific Computing

D4: Breads, Pastries, Pies and Cakes: Quantity Baking Recipes

D5: Pastry: A Book of Best French Recipies

In forming the term-by-document matrix $\hat{A}$ representing this database, say we let $\hat{a}_{ij}$ be the number of times term i appears in the title of document j. Then we have:

$$\hat{A} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Usually the term-by-document matrix is put into normalized form:

$$A = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1.0000 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix} \tag{5.1}$$

Now let us say that we want to query the database hoping to find books about "baking bread". The vector for this query would be

$$\hat{q}^{(1)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}^T,$$

the one in the first position representing the first term, "baking", and the one in the third position representing the third term, "bread". Normalized, the query vector is

$$q^{(1)} = \begin{pmatrix} 0.7071 & 0 & 0.7071 & 0 & 0 & 0 \end{pmatrix}^T,$$

To query the database, we can take the dot product of the query vector $q^{(1)}$ with each of the columns of A. This will give us five numbers, the cosines of the angles θ_k between $q^{(1)}$ and each column a_k of A. A large cosine means that the two vectors are close to each other, whereas a small cosine means that the two vectors are far apart. Let us set a cutoff value of 0.5. Then the columns a_k of

A which return a cosine of greater than 0.5 are considered to be retrieved in the query.

For the query vector $q^{(1)}$, the only nonzero cosines are $\cos\theta_1 = 0.8165$ and $\cos\theta_4 = 0.5774$. This is what we would hope to get, for documents one and four indeed concern baking bread, whereas documents two, three, and five do not.

Let us see what happens if we try a query with only the word "baking". Then the query vector is

$$q^{(2)} = \hat{q}^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T,$$

and the cosines are 0.5774, 0, 0, 0.4082, and 0. So only the first document is retrieved. We would have liked to have had the fourth document retrieved, as well, since it is a more comprehensive book about baking.

Berry, Drmač, and Jessup [BDJ99] use this last query to motivate the need for something more than just this basic vector space model of a database. They list various adaptations that can be made, but one of the most intriguing is called *latent semantic indexing.*

The idea in LSI is to replace the term-by-document matrix by one of a lower rank. We will lose some information by making this replacement, but there are definite advantages to making the switch. First of all, performing queries can be much faster with the rank-reduced representation. Second, the new matrix models the semantic content of the database better than the original matrix. This was what was motivated in the last example. It is believed that by reducing the rank of the matrix, much of the noise in the database is removed, and this is what makes for better queries. Consider searching the term "Samuel Clemens".

We would like the query to return documents pertaining to Mark Twain. We can see that querying with the original term-by-document matrix is not going to do it for us. Reducing the rank greatly increases the probability of such success, however, as the noise in the database is reduced and its semantic content focused. For more on this, see [BDJ99] and [BB99].

5.2 Using QR and SVD to Reduce the Rank

Two possible ways of reducing the rank are the QR factorization and the SVD. We can take the QR factorization of A and partition as we usually do:

$$A = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}.$$

If $\|R_{22}\|$ is small, one possible way of reducing the rank is to simply throw R_{22} away, obtaining the reduced rank form:

$$\tilde{\mathrm{A}} = Q_1 \begin{pmatrix} R_{11} & R_{12} \end{pmatrix}.$$

If the SVD is used, the typical rank reduction can be made. Namely, if $A = U\Sigma V^T$, we just let $A_k = U_k \Sigma_k V_k$, where U_k and V_k have k columns and Σ_k is k-by-k.

In trying to remove redundancy by reducing the rank of A the question arises as to how small we can allow the rank to be without compromising the information in the matrix beyond what is acceptable. The smaller the rank, the faster we can

perform queries. It is pointed out in [BDJ99] that uncertainties of even 25% can be introduced into A simply by difference of opinion between human indexers of the database. On the other hand, a 50% loss of data would be unacceptable, going beyond the uncertainty in the database. So there are rough guidelines to follow as we reduce the rank of a factorization of A in pursuit of faster queries.

What relative change is there to R (and hence A, since Q is orthogonal) when we set R_{22} to zero in the QR factorization of A? It is simply the norm of R_{22} divided by the norm of R. If we are using the Frobenius norm, $\|R_{22}\|_F/\|R\|_F$. If we instead use the SVD, the relative change is simply $\sqrt{\sum_{j=k+1}^{n}\sigma_j^2}/\|A\|_F$. So for a given rank k, we can measure the loss of information in either method, and adjust k so that the loss is acceptable.

As we know, the SVD is more expensive to compute than the QR factorization. But one big advantage of using the SVD is that it gives reduced rank bases of both the column and row space of A. They are spanned by the columns of U and of V, respectively. The QR factorization gives only a reduced rank basis for the column space, which is spanned by the columns of Q. The row space basis is useful for term-term comparisons and narrowing searches. See [BDJ99].

Obviously, the QLP decomposition is a candidate for use in latent semantic indexing. It is cheap to compute and provides an approximate SVD. Indeed, the first k columns of the matrices Q and P provide reduced rank bases for, respectively, the column and row space of A, as they are the QLP approximations of U_k and V_k. Plus, the QLP decomposition gives excellent approximations to the singular values, so we can tell by looking at the approximate singular values $|l_{k+1,k+1}|, \ldots, |l_{n,n}|$ how much information will be lost by making a rank k approximation.

	QR	SVD	QLP as QR	QLP as SVD
rank 3	0.26	0.19	0.20	0.20
rank 2	0.51	0.42	0.44	0.43

Table 5.1: The loss of error for the three methods when reducing to a rank-3 or rank-2 approximation.

5.3 QLP Applied to an Example

Let us apply QLP to the example of Section 5.1. We can apply both the QR way and the SVD way of measuring relative change to our approximation and compare the results with those of the QR and SVD, which are reported in [BDJ99]. They consider using rank-three and rank-two approximations to the example matrix A (see (5.1)). The relative changes in A when using QR, the SVD, and the QLP decomposition, viewing the QLP as a QR and as an SVD are given in Table 5.1.

To compute the QLP statitics viewing the factorization as a QR, we compute $\|L_{22}\|_F/\|L\|_F$. For viewing it as an SVD, we compute $\sqrt{\sum_{i=k+1}^{n} l_{i,i}^2}/\|L\|_F$.

From the first two columns of the table, we see that the SVD allows a given rank reduction with less data loss than the QR, which is not surprising. What is more informative is that taking the rank-three approximation (of either QR or SVD) costs an acceptable relative change in A (around 0.25 or less), whereas taking the rank-two approximation loses too much data and is therefore unacceptable. Note that the statistics for the QLP decomposition accurately reflect this. In fact, the numbers for the QLP (viewed either way) are almost the same as for the SVD, just slightly worse, which is what we would expect.

Now we show how the QLP decomposition handles the two queries $q^{(1)}$ and $q^{(2)}$ from Section 5.1 and compare the results with what the SVD gives. The cosine between the query vector and each column of the respective rank-three

Query 1

SVD	0.73	-0.05	0.03	0.72	-0.01
QLP	0.82	0.00	0.00	0.71	0.00

Table 5.2: The cosines between $q^{(1)}$ and the columns of the rank three approximation given by the SVD and QLP.

Query 2

SVD	0.52	-0.03	0.02	0.51	-0.01
QLP	0.58	0.00	0.00	0.50	0.00

Table 5.3: The cosines between $q^{(2)}$ and the columns of the rank three approximation given by the SVD and QLP.

approximation are given in Tables 5.2 and 5.3.

Note that the QLP results are quite similar to the SVD results. In particular, if we again use the cutoff value of 0.5 (i.e., a document is considered retrieved in the query if the cosine between the document vector and the query vector is at least 0.5), then both queries retrieve documents 1 and 4, whichever factorization is used.

So based on this example, the QLP decomposition looks promising for use in latent semantic indexing. It correctly indicates the appropriate rank reduction to make and it performs well on the queries.

5.4 Some Observations

We would like to make a few final comments on the use of the QLP decomposition in latent semantic indexing and indicate possible directions for future research. Note that we are usually solving a low-rank problem because of much redun-

dancy in the term-by-document matrix. Therefore the QLP decomposition can be truncated as discussed in Chapter 3.

In fact, we would almost certainly want to use interleaving as well. As we saw above, one problem is to determine what the rank k of the low-rank approximation should be. If we replace the Frobenius norm by the 2-norm, the relative change in A is $\sigma_{k+1}/\|A\|_2 \approx |l_{k+1,k+1}|/|l_{1,1}|$. So we could interleave the computation of L, even one column at a time, until we reach a ratio $|l_{k+1,k+1}|/|l_{1,1}|$ which represents an acceptable amount of information loss. At this point, the factorization is complete, for we have the first k columns of Q and of P and the matrix $L(1 : k, 1 : k)$.

Note that the term-by-document matrix is usually sparse, as in the example earlier. When the QR factorization of a sparse matrix is taken, the sparsity is fairly well preserved in the first few columns. This sparsity might be able to be used to advantage to speed up queries. Note the query results given in Tables 5.2 and 5.3. Whereas the SVD has merely small cosines for the documents we ignored, the QLP has *zero* cosines there. This is due to the sparsity being maintained in the Q matrix. So QLP has the nice property that both the reduced rank column and row spaces are approximated as with the SVD, but sparsity is fairly well preserved in the first few columns of the column space approximation (the one used for queries) as with the QR factorization.

Finally, note that if the term-by-document matrix is normalized, then all of the columns, of course, have norm equal to one. This means that in the first step of the QLP decomposition, there will actually be no pivoting in the pivoted QR factorization. So pivoting will be used on neither step. This is a very intriguing fact, for it implies for this low-rank problem that it does not matter which columns we take before we truncate. We do get the information from all the entries of

the matrix into the factorization. During the process of triangularizing the first k columns, all of the columns of the matrix are multiplied by the k Householder transformations, and so all of the elements in the last $n-k$ columns contribute to the formation of the first k rows of R, if only passively. One question is whether there is an alternative to pivoting that can speed up the factorization or perhaps allow us to take a smaller k. At any rate, the QLP decomposition is certainly worth a look for use in latent semantic indexing.

CHAPTER 6

Conclusion

We have studied the pivoted QLP decomposition, which represents the first two steps in an algorithm which approximates the SVD. The matrix $A\Pi_0$ is first factored as $A\Pi_0 = QR$, and then the matrix $R^T\Pi_1$ is factored as $R^T\Pi_1 = PL^T$, resulting in $A = Q\Pi_1 L P^T \Pi_0^T$, with Q and P orthogonal, L lower-triangular, and Π_0 and Π_1 permutation matrices. Stewart noted that the diagonal elements of L approximate the singular values of A with surprising accuracy, and we have provided mathematical justification for this phenomenon.

Specifically, we showed that if there is a gap between σ_k and σ_{k+1}, partition the matrix L into diagonal blocks L_{11} and L_{22} and off-diagonal block L_{21}, where L_{11} is k-by-k. We show that the convergence of $(\sigma_j(L_{11})^{-1} - \sigma_j^{-1})/\sigma_j^{-1}$ for $j = 1, \ldots, k$, and of $(\sigma_j(L_{22}) - \sigma_{k+j})/\sigma_{k+j}$, for $j = 1, \ldots, n-k$ are all quadratic in the gap ratio σ_{k+1}/σ_k. The worst case is therefore at the gap, where the absolute errors $\|L_{11}^{-1}\| - \sigma_k^{-1}$ and $\|L_{22}\| - \sigma_{k+1}$ are thus cubic in σ_k^{-1} and σ_{k+1}, respectively.

The derivation illuminated the fact that one order of covergence is due to the rank-revealing pivoting in the first step; then, because of the pivoting in the first step, two more orders are achieved in the second step. In particular, the one order in the first step comes from the fact that we can bound the norms of R_{11}^{-1} and of R_{22} by σ_1^{-1} times a constant and σ_n times a constant, respectively, with the constants depending only on k and n. Our analysis assumes that $\Pi_1 = I$, that is, that pivoting is done only on the first step.

The algorithm can be continued beyond the first two steps, and we made some observations concerning the asymptotic convergence of individual elements. Assuming that Givens rotations are used to triangularize the matrix at each iteration, we saw that there were basically two effects that a Givens rotation can have on an off-diagonal element. When all of the off-diagonal elements are roughly the same size, then one type of effect dominates, and convergence is relatively slow. When, on the other hand, there are significant disparities in the sizes of off-diagonal elements, the other type of effect dominates, and the convergence is accelerated. We noted some of the common patterns.

We also showed that the asymptotic convergence of the diagonal element $r_{ss}^{(i)}$ is the square of the slowest asymptotic rate among all elements in row s and column s. We were then able to produce a bound on the asymptotic convergence rate of diagonal elements. Numerical experiments illustrated results throughout Chapter 2.

We next turned to a variant of the pivoted QLP decomposition which can be used in low-rank problems in lieu of the truncated SVD. Stewart had suggested truncating and interleaving the algorithm, and we showed that the theory for the full decomposition carries over to the truncated case. In particular, we showed that if there is a gap between σ_k and σ_{k+1}, again the convergence of $(\sigma_j(L_{11})^{-1} - \sigma_j^{-1})/\sigma_j^{-1}$ for $j = 1, \ldots, k$, are all quadratic in the gap ratio σ_{k+1}/σ_k—this without having to compute the rest of the matrix L. We also performed some operation counts and compared the truncated QLP to the full QLP. Whereas the full QLP runs in $\mathcal{O}(mn^2)$ time, the truncated version runs in $\mathcal{O}(mnk)$ time, where we truncate after k rows, making it essentially a quadratic algorithm when k is small.

Since the pivoted QLP decomposition and its truncated little brother approx-

imate the SVD and the truncated SVD, respectively, we considered just a couple of applications which require SVD-quality information and looked at what the QLP could do.

We first considered the problem of estimating condition number, since the QLP provides approximations to σ_1 and σ_n, and we derived a bound on the underapproximation ratio. As in the literature, we assumed that we already had a convenient factorization of A, and in particular, we assumed that we had A in the form of a pivoted QR factorization. Given such a factorization, a condition estimator should run in $\mathcal{O}(n^2)$ time. So the full QLP, which requires another $\mathcal{O}(n^3)$ operations, would probably be used for a condition estimate only if the full QLP were desired anyway, in which case the condition estimate would be had for free, or if n were of a size that computing the full QLP made sense. So using ideas related to the QLP as inspiration, we discussed techniques that provide estimates better than that of the pivoted QR factorization but run in $\mathcal{O}(n^2)$ or even $\mathcal{O}(n)$ time. Two of these dealt with getting a better approximation to σ_1 than simply taking $|r_{11}|$ from the pivoted QR factorization. The other two tricks attempted to get a better handle on σ_n, the more elusive of the two.

We carried out some numerical experiments to assess the ability of the QLP and one of these other methods to provide good condition number estimates. We looked at the QLP and at QRplus, which means taking the norm of the first row of R as the approximation to σ_1, and applied essentially the same tests that Higham applied in his survey of several condition numbers [Hig87]. Both QLP and QRplus gave at least order-of-magnitude approximations to the condition number and can be considered reliable condition number estimators.

Finally we used the QLP to do latent semantic indexing. We saw how the SVD is used to reduce the rank of the term-by-document matrix representing a

database, and that it is observed that the rank-reduced matrix represents the semantic content of the database better than the original full-rank matrix. As this is an observation, and since the "best" representation is somewhat subjective, the SVD does not necessarily yield the optimum results.

Here is a place where the QLP decomposition can really shine, for the error in using it to approximate the SVD causes little damage in an application where precision is not of the utmost importance anyway, and results are somewhat subjective. We noted that the QLP, just like the SVD, provides approximations for both the rank-reduced column space and row space, the former needed to perform queries and the latter needed for term-term comparisons and refining searches. We noted that in the process of truncating the QLP decomposition, interleaving comes in very handy, as we can quickly determine when to truncate based on the values of the diagonal elements being produced. We want to truncate when the relative error in representing the matrix decreases beyond an upper threshhold, and the L-values tell us when this happens. We applied the QLP to a small example, and it performed quite well, both in informing us when to truncate and in doing queries. The QLP decomposition seems to hold promise for use in latent semantic indexing.

We have studied Stewart's pivoted QLP decomposition, illuminating its convergence and showing it at work in a couple of application areas. We hope that we have demonstrated that it is a worthy substitute for the SVD in many situations.

References

[BB99] Michael W. Berry and Murry Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval.* SIAM, Philadelphia, 1999.

[BDJ99] Michael W. Berry, Zlatko Drmač, and Elizabeth R. Jessup. "Matrices, Vector Spaces, and Information Retrieval." *SIAM Review*, **41**(2):335–362, 1999.

[CCL81] Alan K. Cline, Andrew R. Conn, and Charles F. Van Loan. "Generalizing the LINPACK Condition Estimator." In *Numerical Analysis, Proceedings, Cocoyoc, Mexico*, number 909 in Lecture Notes in Mathematics, pp. 73–83. Springer-Verlag, 1981.

[Cha87] T. F. Chan. "Rank Revealing QR Factorizations." *Linear Algebra and Its Applications*, **88/89**:67–82, 1987.

[CI94] S. Chandrasekaran and I.C.F. Ipsen. "On Rank-Revealing Factorisations." *SIAM Journal on Matrix Analysis and Applications*, **15**(2):592–622, 1994.

[CI95] S. Chandrasekaran and I.C.F. Ipsen. "Analysis of a QR Algorithm for Computing Singular Values." *SIAM Journal on Matrix Analysis and Applications*, **16**(2):520–535, 1995.

[CMS79] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson. "An Estimate for the Condition Number of a Matrix." *SIAM Journal on Numerical Analysis*, **16**(2):368–375, 1979.

[CR83] A. K. Cline and R. K. Rew. "A set of Counter-Examples to Three Condition Number Estimators." *SIAM Journal on Scientific and Statistical Computing*, **4**(4):602–611, 1983.

[Dem92] J. W. Demmel. "Open Problems in Numerical Linear Algebra." IMA preprint series #961, Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN, 1992.

[Dem97] James W. Demmel. *Applied Numerical Linear Algebra.* SIAM, Philadelphia, 1997.

[Fos86] L. V. Foster. "Rank and null space calculations using matrix decomposition without column interchanges." *Linear Algebra and Its Applications*, **74**:47–71, 1986.

[GL96] G. Golub and C. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, third edition, 1996.

[Gol65] G. H. Golub. "Numerical methods for solving least squares problems." *Numerische Mathematik*, **7**:206–216, 1965.

[Hig87] Nicholas J. Higham. "A Survey of Condition Number Estimation for Triangular Matrices." *SIAM Review*, **29**(4):575–596, 1987.

[Hig92] Nicholas J. Higham. "Estimating the Matrix p-norm." *Numerische Mathematik*, **62**:539–555, 1992.

[Hig96] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 1996.

[HT00] Nicholas J. Higham and Françoise Tisseur. "A Block Algorithm for Matrix 1-Norm Estimation, with an Application to 1-Norm Pseudospectra." *SIAM Journal on Matrix Analysis and Applications*, **21**(4):1185–1201, 2000.

[Kah66] W. Kahan. "Numerical Linear Algebra." *Canadian Math. Bulletin*, **9**:757–801, 1966.

[MS93] R. Mathias and G. W. Stewart. "A Block QR Algorithm and the Singular Value Decomposition." *Linear Algebra and Its Applications*, **182**:91–100, 1993.

[Ste73] G. W. Stewart. *Introduction to Matrix Computations.* Academic Press, New York, 1973.

[Ste92] G. W. Stewart. "An Updating Algorithm for Subspace Tracking." *IEEE Transactions on Signal Processing*, **40**:1535–1541, 1992.

[Ste93] G. W. Stewart. "Updating a rank-revealing ULV Decomposition." *SIAM Journal on Matrix Analysis and Applications*, **14**:494–499, 1993.

[Ste98] G. W. Stewart. *Matrix Algorithms, Volume 1: Basic Decompositions.* SIAM, Philadelphia, 1998.

[Ste99] G. W. Stewart. "The QLP Approximation to the Singular Value Decomposition." *SIAM Journal on Scientific Computing*, **20**(4):1336–1348, 1999.

[TB97] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra.* SIAM, Philadelphia, 1997.