UMIACS TR 90-87 July 1990
CS TR 2495

Incremental Condition Calculation and Column Selection

G. W. Stewart*

ABSTRACT

This paper describes a method for calculating the condition number of a matrix in the Frobenius norm that can be used to select columns in the course of computing a QR decomposition. When the number of rows of the matrix is much greater than the number of columns, the additional overhead is negligible. Limited numerical experiments suggest that the method is quite good at finding gaps in the singular values of the matrix.

*Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. This work was supported in part by the Air Force Office of Sponsored Research under Contract AFOSR-87-0188.

Incremental Condition Calculation and Column Selection

G. W. Stewart*

ABSTRACT

This paper describes a method for calculating the condition number of a matrix in the Frobenius norm that can be used to select columns in the course of computing a QR decomposition. When the number of rows of the matrix is much greater than the number of columns, the additional overhead is negligible. Limited numerical experiments suggest that the method is quite good at finding gaps in the singular values of the matrix.

1. Introduction

The problem of CONDITION ESTIMATION is to approximate the norm of an inverse or pseudo-inverse of a matrix. The matrix is typically a triangular matrix R—often the result of computing a QR or LR factorization. The first condition estimator was devised by Gragg and Stewart [8], and was later improved by Cline, Moler, Stewart, and Wilkinson [4] for incorporation into LINPACK [5]. Other condition estimators have since been proposed, many of which are treated in a survey by Higham [9].

Condition estimators typically trade precision for speed. They produce an approximation to the norm of the inverse, usually a lower bound, in $O(n^2)$ time, where n is the order of R. This attempt to save work makes sense when R is a triangular factor of a matrix of order n, since an $O(n^3)$ algorithm would take time proportional to the factorization itself. However, when R is from the QR decomposition of an $m \times n$ matrix with $m \gg n$, there is less need of economy, since the QR decomposition requires $O(mn^2)$ operations to compute. In this paper we propose an $O(n^3)$ algorithm that implicitly computes the inverse of R and its Frobenius norm.

A closely related problem is that of COLUMN SELECTION. The goal of column selection is to determine a maximal set of "independent" columns of R. Here "independence" means that the smallest singular value of the matrix formed from the

*Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. This work was supported in part by the Air Force Office of Sponsored Research under Contract AFOSR-87-0188.

columns is greater than some prescribed tolerance. Although no one has proven that the problem can be solved — at least for large n — there are a number of effective strategies. Perhaps the oldest of these is to compute the QR decomposition with column pivoting, as suggested by Golub [6]. This method is an example of an INCREMENTAL SELECTION ALGORITHM, since it can select columns as the QR decomposition is computed. Other methods work with a precomputed decomposition. For example, the method of Golub, Klema, and Stewart [7] starts with a singular value decomposition, and the rank revealing method of Chan [3] starts with a QR decomposition.

Recently Bischof [1] has proposed an incremental condition estimator and shown how it can be used to drive a selection method. Since the condition calculator proposed here is incremental, it also yields a selection method. Unfortunately, this method (as well as Bischof's method and column pivoting) fails on a matrix devised by Kahan [10]. However, the quantities computed by the condition calculator not only detect the failure but they point to a cure.

In the next section the condition calculator and its use as a column selection strategy will be described. In §3 the selection method is shown to fail on Kahan's matrix, and a remedy is described. The last section is devoted to presenting the results of numerical experiments and concluding remarks. Throughout this paper, $\|A\|$ is the spectra norm of A, $\|A\|_{\mathrm{F}}$ the Frobenius norm of A, and $\inf(A)$ the smallest singular value of A.

2. The Condition Calculator

The condition calculator is based on an incremental scheme for calculating the columns of R^{-1}. Let R be partitioned in the form

$$R = \begin{array}{c} \\ {\scriptstyle k} \\ {\scriptstyle n-k} \end{array} \!\! \begin{array}{c} \begin{array}{cc} {\scriptstyle k} & {\scriptstyle n-k} \end{array} \\ \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \end{array}, \tag{2.1}$$

and assume that the quantities[1]

$$\nu = \|R_{11}^{-1}\|_{\mathrm{F}}$$

and

$$S_{12} \equiv (s_k, \ldots, s_n) = R_{11}^{-1} R_{12}.$$

[1]We have chosen to calculate the Frobenius norm of R^{-1}. However, the algorithm of this section can easily be adapted to calculate the 1-norm.

$\nu = 0$
for $k = 0$ **to** n
  $\nu = \sqrt{\nu^2 + \alpha_{k+1}^{-2}(1 + \|s_{k+1}\|^2)}$
  $s_j = \begin{pmatrix} s_j - \alpha_{k+1}^{-1}\rho_{k+1,j}s_{k+1} \\ \alpha_{k+1}^{-1}\rho_{k+1,j} \end{pmatrix}, \quad j = k+2,\ldots,n$
end

Figure 2.1: Basic Condition Calculator

have already been computed.

It is easily verified that if $(r_{k+1}\ \alpha_{k+1})^{\mathrm{T}}$ is the $(k+1)$th column of R then

$$\hat{R}_{11}^{-1} \equiv \begin{pmatrix} R_{11} & r_{k+1} \\ 0 & \alpha_{k+1} \end{pmatrix}^{-1} = \begin{pmatrix} R_{11}^{-1} & -\alpha_{k+1}^{-1}s_{k+1} \\ 0 & \alpha_{k+1}^{-1} \end{pmatrix}^{-1}.$$

Hence the norm of $\hat{R}_{11}^{-1}$ is

$$\hat{\nu} = \sqrt{\nu^2 + \alpha_{k+1}^{-2}(1 + \|s_{k+1}\|^2)}. \tag{2.2}$$

Moreover, the columns of the corresponding matrix $\hat{S}_{12}$ are given by

$$\begin{aligned} &\begin{pmatrix} R_{11}^{-1} & -\alpha_{k+1}^{-1}s_{k+1} \\ 0 & \alpha_{k+1}^{-1} \end{pmatrix}^{-1} \begin{pmatrix} r_j \\ \rho_{k+1,j} \end{pmatrix} \\ = &\begin{pmatrix} s_j - \alpha_{k+1}^{-1}\rho_{k+1,j}s_{k+1} \\ \alpha_{k+1}^{-1}\rho_{k+1,j} \end{pmatrix}, \qquad j = k+2,\ldots,n. \end{aligned} \tag{2.3}$$

A summary of a condition calculator based on (2.2) and (2.3) is given in Figure 2.1.

Regarded simply as a condition calculator this algorithm has little to recommend it, since it requires twice the number of flops needed to compute R^{-1} directly. However, the fact that S is present allows the algorithm to be used in a column selection strategy. Specifically, let

$$\left.\begin{aligned} \sigma_j &= \|s_j\| \\ \alpha_j &= \sqrt{\textstyle\sum_{i=k+1}^{j} \rho_{ij}^2} \end{aligned}\right. \qquad j = k+1,\ldots,n.$$

Since α_j is the norm of the part of the jth column of R that lies below the kth row, if that column is swapped with the $(k+1)$th column and the resulting matrix is reduced to triangular form, the column assumes the form $(r_j^{\mathrm{T}}\ \alpha_j\ 0\ \ldots\ 0)^{\mathrm{T}}$. Hence, if the jth column were to replace the $(k+1)$th, the new norm ν_{k+1} would be given by

$$\nu_{k+1} = \sqrt{\nu_k^2 + \alpha_j^{-2}(1+\sigma_j^2)}.$$

Thus a natural selection strategy is to choose j so that $\alpha_j^{-2}(1+\sigma_j^2)$ is minimal and replace column $k+1$ by column j.

The swapping of columns can be accomplished in a standard way with plane rotations. Briefly, to move, say, the fifth column backward, interchange the fifth and fourth column, to get a matrix of the form

$$\begin{pmatrix} \mathsf{X} & \mathsf{X} & \mathsf{X} & \mathsf{X} & \mathsf{X} \\ \mathsf{O} & \mathsf{X} & \mathsf{X} & \mathsf{X} & \mathsf{X} \\ \mathsf{O} & \mathsf{O} & \mathsf{X} & \mathsf{X} & \mathsf{X} \\ \mathsf{O} & \mathsf{O} & \mathsf{O} & \mathsf{X} & \mathsf{X} \\ \mathsf{O} & \mathsf{O} & \mathsf{O} & \mathsf{X} & \mathsf{O} \end{pmatrix}$$

Then restore R to triangular form by applying a rotation P_{45} in the $(4,5)$ plane to annihilate the element below the diagonal. Thus by repeated swaps and restorations, the column is moved into position. The effect of the plane rotations may be accumulated in an $n \times n$ array.

A summary selection algorithm is given in Figure 2.2. The amount of work done by this algorithm will depend on which columns are swapped, but an upper bound is n^3 flops. Accumulating the rotations requires an additional maximum of $2n^3$ flops.

The column selection strategy can be used in conjunction with the reduction of an $m \times n$ matrix A to triangular form by Householder transformations. Specifically, just after the kth step in the reduction, Householder transformations $H_1, \ldots, H_k$ and permutations $\Pi_1, \ldots, \Pi_k$ have been determined so that

$$H_k \cdots H_1 A \Pi_1 \cdots \Pi_k = \begin{pmatrix} R_{11} & R_{12} \\ 0 & A_{22} \end{pmatrix},$$

where R_{11} is a $k \times k$ upper triangular matrix. The matrix S and the scalars σ_i are defined as usual; however, the α_i are now the norms of the columns of A_{22}. The

$\nu = 0$
$\sigma_j = 0, \quad j = 1, \ldots, n$
$\alpha_j = \sqrt{\sum_{i=k+1}^{j} \rho_{ij}^2}, \quad j = 1, \ldots, n$
for $k = 0$ **to** n
  Determine j so that $\alpha_j^{-2}(1 + \sigma_j^2)$ is minimal.
  Swap column j into column $k+1$.
  $\nu = \sqrt{\nu^2 + \alpha_{k+1}^{-2}(1 + \|s_{k+1}\|^2)}$
  $s_j = \begin{pmatrix} s_j - \alpha_{k+1}^{-1}\rho_{k+1,j}s_{k+1} \\ \alpha_{k+1}^{-1}\rho_{k+2,j} \end{pmatrix}, \quad j = k+2, \ldots, n$
  $\sigma_j = \|s_j\|, \quad j = k+2, \ldots, n$
  $\alpha_j = \sqrt{\alpha_j^2 - \rho_{k+1,j}^2}, \quad j = k+2, \ldots, n$
end

Figure 2.2: Selection Algorithm

column for which $\alpha_j^{-2}(1+\sigma_j^2)$ is minimal is swapped with the $(k+1)$th column by means of a permutation Π_{k+1}, and a Householder transformation H_{k+1} reduces it. The quantities S, σ_i, and α_i are updated as usual.[2]

3. A Recovery Procedure

One of the purposes of column selection is to reveal gaps in the singular values. Specifically, let the singular values of R be

$$\psi_1 \geq \psi_2 \geq \cdots \geq \psi.$$

and suppose that for some integer p

$$\psi_p \gg \psi_{p+1}.$$

[2] Some care must be taken in updating the α_i, since cancellation can make them meaningless. See the program SQRDC in [5] for details. In the unlikely event that cancellation occurs in updating a column of S, the column can easily be calculated *ab initio*.

A successful selection strategy should produce a matrix R of the form

$$R = \begin{array}{c c} & \begin{array}{cc} p & n-p \end{array} \\ \begin{array}{c} p \\ n-p \end{array} & \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \end{array},$$

with $\inf_2(R_{11}) \cong \rho_{pp} \cong \psi_p$ and $\|R_{22}\|_2 \cong \rho_{p+1,p+1} \cong \psi_{p+1}$. Unfortunately, the selection strategy of the last section, along with most others, fails on the following example due to Kahan [10].

Let

$$c^2 + s^2 = 1,$$

and let K_n be the triangular matrix illustrated below for $n = 5$:

$$K_5 = \mathrm{diag}(1, s, s^2, s^3, s^4) \begin{pmatrix} 1 & -c & -c & -c & -c \\ 0 & 1 & -c & -c & -c \\ 0 & 0 & 1 & -c & -c \\ 0 & 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The norm of each column of the trailing $l \times l$ principal submatrix of K_n is easily seen to be s^{n-l}. Consequently, if the selection algorithm breaks ties by doing nothing, the final result is K_n itself.

Unfortunately, the columns of K_n are nearly dependent; or what is equivalent K_n has a small singular value. To see this, set

$$t = \frac{1}{1+c} \leq 1.$$

Then it is easily verified that

$$\mathrm{diag}(1, s, s^2, s^3, s^4) \begin{pmatrix} 1 & -c & -c & -c & -c \\ 0 & 1 & -c & -c & -c \\ 0 & 0 & 1 & -c & -c \\ 0 & 0 & 0 & 1 & -c \\ -\frac{t^4}{1-t} & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \\ \frac{t^4}{1-t} \end{pmatrix} = 0.$$

Thus a perturbation of $s^{n-1}t^{n-1}/(1+t)$ will make K_n singular, and its smallest singular value can be no greater than this quantity. In particular, the ratio of the smallest singular value to the smallest element diagonal element is bounded by

$$\frac{t^{n-1}}{1-t}.$$

When $c = 0.2$ and $n = 50$, this ratio is about $8 \cdot 10^{-4}$, which shows that the column selection algorithm fails catastrophically for this example.

The failure might be understandable if the singular values of K_n had no gap. But in fact K_n has only one small singular value; and if the first column is removed, the remaining columns are strongly independent. The problem is how to decide that the first column is the one to remove. The following heuristic strategy is based on the condition computer.

Let R be partitioned as in (2.1), and let S and α_i be as usual. The strategy consists in swapping out one of the columns of R_{11} in favor of a another column not yet entered. The choice is made to minimize a bound on the the value of α for the column swapped out. The rationale is that producing small values of α outside the current factorization will tend to expose gaps in the singular values.

To derive the bound, suppose that the jth column ($j \le k$) is to by swapped out and the lth column ($k < l$) is to be swapped in. Let $\hat{\alpha}_{jl}$ be the value of α corresponding to the jth column after the swap. Note that

$$\hat{\alpha}_{jl} = \left\| P_{jl}^{\perp} \begin{pmatrix} r_j \\ 0 \end{pmatrix} \right\|, \tag{3.1}$$

where $P_{jl}^{\perp}$ is the projection onto the orthogonal complement of the space spanned by

$$\begin{pmatrix} r_l \\ \alpha_l \end{pmatrix}, \begin{pmatrix} r_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} r_{j-1} \\ 0 \end{pmatrix}, \begin{pmatrix} r_{j+1} \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} r_k \\ 0 \end{pmatrix}.$$

Now by the definition of s_l,

$$\begin{pmatrix} r_l \\ \alpha_l \end{pmatrix} = \begin{pmatrix} R_{11} \\ 0 \end{pmatrix} s_l + \begin{pmatrix} 0 \\ \alpha_l \end{pmatrix}.$$

Hence if σ_{jl} denotes the jth component of s_l, it follows

$$\begin{pmatrix} r_j \\ 0 \end{pmatrix} = \sigma_{jl}^{-1} \left[\begin{pmatrix} r_l \\ \alpha_l \end{pmatrix} - \sum_{\substack{i=1 \\ i \ne j}}^{k} \sigma_{il} \begin{pmatrix} r_1 \\ 0 \end{pmatrix} \right] - \begin{pmatrix} 0 \\ \sigma_{jl}^{-1} \alpha_l \end{pmatrix}.$$

If this equation is multiplied by P_{jl}, it follows from (3.1) that

$$\hat{\alpha}_{jl} = \left\| P_{jl}^{\perp} \begin{pmatrix} 0 \\ \sigma_{jl}^{-1}\alpha_l \end{pmatrix} \right\| \leq |\sigma_{jl}^{-1}|\alpha_l.$$

This is the required bound.

Of course there is no need to perform a swap if the diagonals are doing an adequate job of revealing gaps. The incremental condition calculator can be used to tell when things go wrong. Let $\mathrm{tol}(k)$ be a tolerance, perhaps depending on k. The heuristic strategy is the following.

> *If $\nu\rho_{kk} > \mathrm{tol}(k)$, choose j and l so that $|\sigma_{jl}^{-1}|\alpha_l$ is maximized, and swap columns j and l.*

The details of the swapping are a little involved. The first step is to move column j into the position of column k. If this is done as described in the last section, it amounts to replacing R_{11} by $QR_{11}\Pi$, where Q is orthogonal and Π is a permutation. The matrix R_{12} is replaced by QR_{12}. Hence the matrix S_{12} must be replaced by

$$(QR_{11}\Pi)^{-1}(QR_{12}) = \Pi^{\mathrm{T}}S_{12};$$

which is to say that the interchanges made in the columns of R_{11} must also be made in the rows of S_{12}.

The second step is to remove the kth column (formerly the jth); that is, to reduce k by one and recompute S_{12} and the α_j. Write

$$R_{11} = \begin{pmatrix} \hat{R}_{11} & \hat{r}_k \\ 0 & \rho_{kk} \end{pmatrix}$$

and

$$R_{12} = \begin{pmatrix} \hat{r}_{k+1} & \cdots & \hat{r}_n \\ \rho_{k+1,k+1} & \cdots & \rho_{k+1,n} \end{pmatrix}.$$

The α's corresponding to $\hat{R}_{11}$ are given by

$$\hat{\alpha}_k = \rho_{kk}$$

and

$$\hat{\alpha}_j = \sqrt{\alpha_j^2 + \rho_{kj}^2}, \qquad j = k+1\ldots,n.$$

The columns $\hat{s}_j$ of $\hat{S}_{12}$ satisfy

$$\hat{R}_{11}\hat{s}_j = \hat{r}_j, \qquad j = k, \ldots, n.$$

For $j = k$, the above system must be solved for $\hat{s}_k$, but the other columns of $\hat{S}_{12}$ may be obtained more economically.

Let $(\tilde{s}_j^{\mathrm{T}}\ \sigma_{kj})^{\mathrm{T}}$ be the jth column of S_{12}. Then

$$\begin{pmatrix} \hat{R}_{11} & \hat{r}_k \\ 0 & \rho_{kk} \end{pmatrix} \begin{pmatrix} \tilde{s}_j \\ \sigma_{kj} \end{pmatrix} = \begin{pmatrix} \hat{r}_j \\ \rho_{kj} \end{pmatrix}.$$

Hence

$$\hat{r}_j = \hat{R}_{11}\tilde{s}_j + \sigma_{kj}\hat{r}_k = \hat{r}_j.$$

It follows on multiplying this equation by $\hat{R}_{11}^{-1}$ that

$$\hat{s}_j = \tilde{s}_j + \sigma_{kj}\hat{s}_k, \qquad j = k+1, \ldots, n.$$

The third and last step is to replace column k with column l as described in the last section.

This is a complicated procedure which should only be used *in extremis*. Moreover, it complicates the reduction of a general matrix by Householder transformations, since it intersperses rotations with the Householder transformations.[3] The simplest solution in this case is to note where the condition calculator and the diagonals of the matrix disagree and continue the reduction as usual When the factorization is complete, the matrix S_{12} can be recomputed and the recovery procedure applied.

4. Numerical Experiments and Conclusions

The condition calculator was subjected to a stiff test to see if the associated selection strategy could spot a small gap in the singular values. Random triangular matrices of order twenty with singular values

$$\begin{array}{ll} 2^{1-i}, & i = 1, \ldots, 10 \\ \beta \cdot 2^{1-i}, & i = 11, \ldots, 20 \end{array}$$

[3]Actually, if we write a Householder transformation in the form $H(u) = I - uu^{\mathrm{T}}$, where $\|u\| = \sqrt{2}$, then $H(u)Q = QH(Q^{\mathrm{T}}u)$ for any orthogonal matrix Q. Thus, rotations can be pulled out of the factorization at the cost of updating the vectors that determine the Householder transformations.

Figure 4.1: Summary of Test Results

were generated by the technique described in [11]. The singular values have a gap of ratio β^{-1} between the tenth and eleventh and otherwise show a gentle decrease in size.

The selection strategy was run on two-hundred such matrices, once for $\beta = 0.1$ and again for $\beta = 0.01$. The ratios $\lambda = \rho_{11,11}/\psi_{11}$ and $\mu = \rho_{11,11}/\rho_{10,10}$ were recorded. The empirical distribution of these ratios is shown in Figure 4.1.

The ratios λ should be near one. They are reasonably well behaved: $\rho_{11,11}$ overestimates the singular by no more than a factor of 2.5 and underestimates it by no less than 0.5.

Since μ is the ratio of two diagonal elements, it may be expected to show more variability than λ. Ideally it should be equal to β. For the most part it is close, with only a few cases producing small values. In all cases, the ratio is greater than the grading ratio of two between the other pairs of consecutive singular values, although some may feel that a the smallest ratio of 3.4 for $\beta = 0.1$ is too small for comfort. For $\beta = 0.01$, there is always a reasonable gap in the ratios of the diagonal elements.

No numerical tests of the recovery procedure of the last section were made. The reason is that Kahan's matrix seems to exhaust the good examples. The procedure works perfectly on Kahan's matrix, selecting the first column as the one to be thrown out.

The condition calculator should not be regarded as a rival of other methods; rather it is another tool with which the numerical analyst can probe rank. Its strong point is that it calculates the condition exactly. However, one should not make too much of this, since empirical studies [11, 9] have shown approximate condition estimators to be quite good in practice. The condition calculator is most effective for dense matrices many more rows than columns. For sparse matrices the technique of Bischof and Hansen [2], which combines restricted forward selection with a backward rank-revealing pass [3]. Other combinations will be suggested by the application at hand.

References

[1] C. H. Bischof (1990). "Incremental Condition Estimation." *SIAM Journal on Matrix Analysis and Applications*, **11**, 312–322.

[2] C. H. Bischof and P. C. Hansen (1989). "Structure-Preserving and Rank-Revealing QR-Factorizations." Preprint MCS-P100-0989, Mathematics and Computer Science Division, Argonne National Laboratory.

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

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

[5] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart (1979). *LINPACK User's Guide*. SIAM, Philadelphia.

[6] G. H. Golub (1965). "Numerical Methods for Solving Least Squares Problems." *Numerische Mathematik*, **7**, 206–216.

[7] G. H. Golub, V. Klema, and G. W. Stewart (1976). "Rank Degeneracy and Least Squares Problems." Technical Report TR-751, Department of Computer Science, University of Maryland.

[8] W. B. Gragg and G. W. Stewart (1976). "A Stable Variant of the Secant Method for Solving Nonlinear Equations." *SIAM Journal on Numerical Analysis*, **13**, 880–903.

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

[10] W. Kahan (1966). "Numerical Linear Algebra." *Canadian Mathematical Bulletin*, **9**, 757–801.

[11] G. W. Stewart (1980). "The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators." *SIAM Journal on Numerical Analysis*, **17**, 403–404.