UMIACS-TR-91-39 March 1991
CS-TR 2627

Updating A Rank-Revealing ULV Decomposition

G. W. Stewart*

ABSTRACT

A ULV decomposition of a matrix A of order n is a decomposition of the form $A = ULV^{\mathrm{H}}$, where U and V are orthogonal matrices and L is a lower triangular matrix. When A is approximately of rank k, the decomposition is rank revealing if the last $n - k$ rows of L are small. This paper presents algorithms for updating a rank-revealing ULV decomposition. The algorithms run in $O(n^2)$ time, and can be implemented on a linear array of processors to run in $O(n)$ time.

*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 Scientific Research under Contract AFOSR-87-0188.

Updating A Rank-Revealing ULV Decomposition

G. W. Stewart

ABSTRACT

A ULV decomposition of a matrix A of order n is a decomposition of the form $A = ULV^{\mathrm{H}}$, where U and V are orthogonal matrices and L is a lower triangular matrix. When A is approximately of rank k, the decomposition is rank revealing if the last $n - k$ rows of L are small. This paper presents algorithms for updating a rank-revealing ULV decomposition. The algorithms run in $O(n^2)$ time, and can be implemented on a linear array of processors to run in $O(n)$ time.

1. Introduction

Let A be a matrix of order n. A rank-revealing URV decomposition of A is a reduction of A by unitary transformations to a triangular matrix of the form

$$U^{\mathrm{H}}AV = \begin{pmatrix} R & H \\ 0 & E \end{pmatrix}. \tag{1.1}$$

The decomposition is rank revealing in the sense that the matrices H and E are smaller than some prespecified tolerance, and the smallest singular value of R is greater than that tolerance. Such decompositions — they are not unique — are useful in solving rank deficient systems. Moreover, if $V = (V_1\ V_2)$ is partitioned conformally, then in the spectral norm

$$\|AV_2\| = \left\| \begin{pmatrix} H \\ E \end{pmatrix} \right\| \tag{1.2}$$

so that the columns of V_2 form an orthonormal basis for an approximate null space of A, something required in signal processing applications like direction of arrival estimation.

The advantage of the URV decomposition over the more familiar singular value decomposition is that it can be updated when a row is added to A. The updating procedure, which is described in [6], requires $O(n^2)$ operations and preserves the rank-revealing character of the decomposition. Moreover, it can be implemented on a linear array of n processors to run in $O(n)$ time.

Although the URV decomposition is fully satisfactory for applications like recursive least-squares, it is less satisfactory for applications in which an approximate null space is needed. The reason is the presence of H in (1.2). To see that it should not be there, let $(U_1\ U_2)$ be a partition of U conformal with (1.1). Then it is easily seen that $\|U_2^{\mathrm{H}}A\| = \|E\|$. Consequently the last $n-k$ singular values of A are less than or equal to $\|E\|$, and the corresponding left singular vectors form an approximate null space whose residual has norm less than or equal to $\|E\|$. Thus V_2 is not the best available approximate null space.

In [6] this problem was circumvented by including a refinement step that reduces the size of H. The properties of this refinement have been analyzed in [7]. In experiments with the MUSIC algorithm for direction of arrival estimation, the refinement was found to improve the results [1]. However, it adds extra work, and aesthetically it has the appearance of a stopgap. The purpose of this paper is to present an alternative.

Specifically, we will describe how to update a lower triangular decomposition of the form

$$U^{\mathrm{H}}AV = \begin{pmatrix} L & 0 \\ H & E \end{pmatrix}, \tag{1.3}$$

where U, and V are orthogonal, L is well conditioned, and H and E are small. We will call such a decomposition a rank-revealing ULV decomposition. The updating algorithm consists of two parts: an algorithm to bring a lower triangular matrix into rank-revealing URV form and the updating algorithm proper. We will present the former in the next section and the latter in §3. The paper concludes with some general observations on the algorithms.

2. Deflation

Although adding a row to a matrix cannot decrease its rank, in many applications the matrix is first multiplied by a constant less than one to damp out old information. Under such circumstances, it is possible for the matrix L in (1.3) to become effectively rank deficient. Here we present an algorithm to calculate a rank-revealing ULV decomposition of a lower triangular matrix L.

The first step is to determine a vector w of norm one such that $\omega = \|w^{\mathrm{H}}L\|$ approximates the smallest singular value of L. This can be done by using any of a number of reliable condition estimators [4]. If ω is greater than a prescribed tolerance, then there is nothing to be done. Otherwise, we must modify L by unitary transformations so that its last row becomes small. We will use plane

$$\begin{array}{cc} \rightarrow & \check{w} \\ \rightarrow & w \\ & w \\ & w \end{array} \Longrightarrow \begin{array}{cc} & 0 \\ \rightarrow & \check{w} \\ \rightarrow & w \\ & w \end{array} \Longrightarrow \begin{array}{cc} & 0 \\ & 0 \\ \rightarrow & \check{w} \\ \rightarrow & w \end{array} \Longrightarrow \begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \end{array}$$

Figure 2.1: Reduction of W

$$\begin{array}{ccccc} & & & \downarrow & \downarrow \\ \rightarrow & l & 0 & 0 & 0 \\ \rightarrow & l & l & 0 & 0 \\ & l & l & l & 0 \\ & l & l & l & l \end{array} \Longrightarrow \begin{array}{cccc} & & \downarrow & \downarrow \\ l & \check{l} & 0 & 0 \\ l & l & 0 & 0 \\ l & l & l & 0 \\ l & l & l & l \end{array} \Longrightarrow \begin{array}{cccc} & & & \\ l & 0 & 0 & 0 \\ \rightarrow & l & l & 0 \\ \rightarrow & l & l & l \\ & l & l & l \end{array} \Longrightarrow$$

$$\begin{array}{cccc} & \downarrow & \downarrow & \\ l & 0 & 0 & 0 \\ l & l & \check{l} & 0 \\ l & l & l & 0 \\ l & l & l & l \end{array} \Longrightarrow \begin{array}{cccc} & & & \\ l & 0 & 0 & 0 \\ \rightarrow & l & l & 0 \\ \rightarrow & l & l & l \\ & l & l & l \end{array} \Longrightarrow \begin{array}{cccc} & & \downarrow & \downarrow \\ l & 0 & 0 & 0 \\ l & l & 0 & 0 \\ l & l & l & \check{l} \\ l & l & l & l \end{array} \Longrightarrow \begin{array}{cccc} & & & \\ l & 0 & 0 & 0 \\ l & l & 0 & 0 \\ l & l & l & 0 \\ h & h & h & e \end{array}$$

Figure 2.2: Reduction of PL

rotations to accomplish this reduction. The reader is assumed to be familiar with plane rotations, which are discussed in most texts on numerical linear algebra (e.g., see [3, 5, 8]).

We begin by reducing w the nth unit vector $\mathbf{e}_n$. The reduction is illustrated in Figure 2.1. The two arrows represent the plane of the rotation, and it annihilates the component of w with a check over it. Denote the product of rotations by

$$P = P_1 P_2 \cdots P_{n-1}.$$

Next we apply the rotations P_i to L from the left, as is shown in Figure 2.2. The application of P_i produces a nonzero element in the $(i, i+1)$-element of L. This element is removed by postmultiplying by a plane rotation Q_i. Let $Q = Q_1 Q_2 \cdots Q_{n-1}$ be the product of these rotation.

The appearance of h's and e in the last matrix of Figure 2.2 is meant to indicate that the last row of PLQ is small. To see that this is true, write

$$\omega = \|w^{\mathrm{H}}L\| = \|w^{\mathrm{H}}P^{\mathrm{H}}PLQ\| = \|\mathbf{e}_n^{\mathrm{T}}(PLQ)\|.$$

Thus the last row of PLQ has norm ω.

If the $(n-1) \times (n-1)$ leading principle submatrix of L is sufficiently well conditioned, we have our rank revealing ULV decomposition. If not, we can repeat the deflation procedure until a sufficiently well condition matrix is found in the upper right hand corner.

3. Updating

We now turn to the updating step. Specifically, we suppose that we are given an additional row z^{H} and wish to determine a rank-revealing ULV decomposition of

$$\begin{pmatrix} A \\ z^{\mathrm{H}} \end{pmatrix}.$$

We begin by forming $z^{\mathrm{H}}V$ to bring the row into the coordinate system of the current decomposition. Thus the problem of updating reduces to the problem of updating

$$\begin{pmatrix} L & 0 \\ H & E \\ x^{\mathrm{H}} & y^{\mathrm{H}} \end{pmatrix},$$

where we have partitioned $z^{\mathrm{H}}V = (x^{\mathrm{H}}\ y^{\mathrm{H}})$.

We now reduce y^{H} to $\mathbf{e}_1^{\mathrm{T}}$, as is shown in Figure 3.1, where only the parts corresponding to y^{H} and E are shown. At the end of the reduction the matrix has the form

$$\begin{matrix} l & 0 & 0 & 0 & 0 \\ l & l & 0 & 0 & 0 \\ l & l & l & 0 & 0 \\ h & h & h & e & 0 \\ h & h & h & e & e \\ x & x & x & y & 0 \end{matrix}$$

One way to finish the update is to continue as in Figure 3.2 to incorporate x and what is left of y into the decomposition. This process moves the presumably large elements of x^{H} into the first row of H. If there has been no increase in rank, this

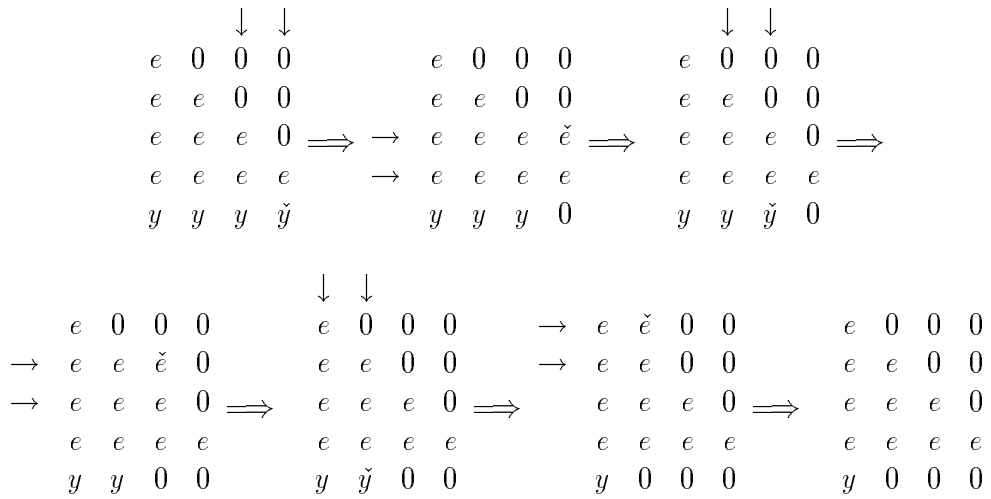

Figure 3.1: Reducing y

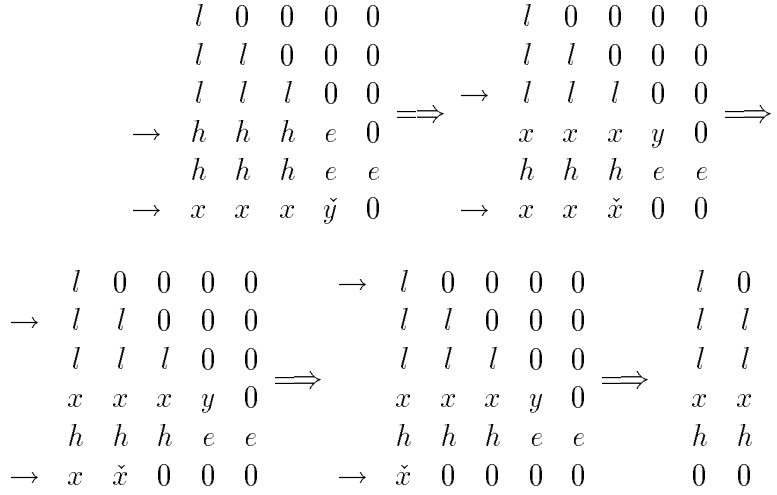

Figure 3.2: Triangularization Step

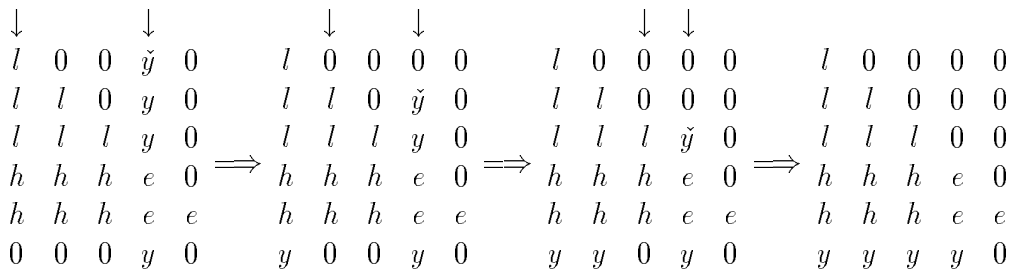

Figure 3.3: Eliminating the Tower of y's

destroys the rank-revealing character of the matrix. However, in that case the deflation procedure will restore the small elements.

If y is small enough, then the rank cannot increase. In that case there is another way of proceeding. Perform the reduction of Figure 3.2 but skip the first step. This will give a matrix having the form of the first matrix in Figure 3.3 with a tower of contributions from the scalar y at the bottom. Rotations are used as shown in the figure to reduce the tower. This fills up the bottom row again, but now the elements are of the same size as y. Since y is small, the algorithm of Figure 3.2 can be used to complete the decomposition without destroying the rank-revealing structure.

It is worth noting that if the y's in the tower are small compared with the diagonal elements of L, the y's along the last column will actually be of order $\|y\|^2$. If only an approximate decomposition is required, it may be possible to neglect them.

4. Comments

We mentioned in the introduction that a ULV decomposition can be expected to give a higher quality approximate null space than a URV decomposition. However, there are trade-offs. It costs more to deflate the URV decomposition if we insist on refinement steps. On the other hand the updating algorithm for the URV decomposition is much simpler. In fact, when there is no change of rank, it amounts to the usual LINPACK updating algorithm SCHUD [2]. Only experience with real-life problems will tell us under what circumstances one decomposition is to be preferred to the other.

Both sets of algorithms are stable and reliable. They are stable because they use orthogonal transformations straightforwardly with no additional implicit relations. They are as reliable as their underlying condition estimators.

In [6] we showed that the algorithms for the URV decomposition could, in principle, be parallelized on a linear array of processors. The same is true of the algorithms for updating a ULV decomposition. Since the techniques to show that the algorithms have parallel implementations are the same for both decompositions, we do not give the details here.

References

[1] G. Adams, M. F. Griffin, and G. W. Stewart (1991). "Direction-of-Arrival Estimation Using the Rank-Revealing URV Decomposition." In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Washington, DC. To appear.

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

[3] G. H. Golub and C. F. Van Loan (1989). *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 2nd edition.

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

[5] G. W. Stewart (1974). *Introduction to Matrix Computations*. Academic Press, New York.

[6] G. W. Stewart (1990). "An Updating Algorithm for Subspace Tracking." Technical Report CS-TR 2494, University of Maryland, Department of Computer Science.

[7] G. W. Stewart (1991). "On an Algorithm for Refining a Rank-Revealing URV Decomposition and a Perturbation Theorem for Singular Values." Technical Report CS-TR 2626, Department of Computer Science, University of Maryland.

[8] D. S. Watkins (1991). *Fundementals of Matrix Computations*. John Wiley & Sons, New York.