University of Maryland College Park

Institute for Advanced Computer Studies TR–94–30
Department of Computer Science TR–3238

On the Stability of Sequential Updates and Downdates*

G. W. Stewart†

March, 1994

ABSTRACT

The updating and downdating of QR decompositions has important applications in a number of areas. There is essentially one standard updating algorithm, based on plane rotations, which is backwards stable. Three downdating algorithms have been treated in the literature: the LINPACK algorithm, the method of hyperbolic transformations, and Chambers' algorithm. Although none of these algorithms is backwards stable, the first and third satisfy a relational stability condition. In this paper, it is shown that relational stability extends to a sequence of updates and downdates. In consequence, other things being equal, if the final decomposition in the sequence is well conditioned, it will be accurately computed, even though intermediate decompositions may be almost completely inaccurate. These results are also applied to the two-sided orthogonal decompositions, such as the URV decomposition.

*This report is available by anonymous ftp from `thales.cs.umd.edu` in the directory `pub/reports`.

†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 National Science Foundation under grant CCR 9115568.

On the Stability of Sequential Updates and Downdates*

G. W. Stewart

ABSTRACT

The updating and downdating of QR decompositions has important applications in a number of areas. There is essentially one standard updating algorithm, based on plane rotations, which is backwards stable. Three downdating algorithms have been treated in the literature: the LINPACK algorithm, the method of hyperbolic transformations, and Chambers' algorithm. Although none of these algorithms is backwards stable, the first and third satisfy a relational stability condition. In this paper, it is shown that relational stability extends to a sequence of updates and downdates. In consequence, other things being equal, if the final decomposition in the sequence is well conditioned, it will be accurately computed, even though intermediate decompositions may be almost completely inaccurate. These results are also applied to the two-sided orthogonal decompositions, such as the URV decomposition.

1. Introduction

Let A be a positive definite matrix of order p. Then A can be written in the form $A = R^{\mathrm{T}}R$, where R is an upper triangular matrix with positive diagonal elements. This factorization is called the *Cholesky decomposition* A, and the matrix R is called its *Cholesky factor.*

In some applications — recursive least squares, for example — it is required to compute the Cholesky decomposition $S^{\mathrm{T}}S$ of $B = A + xx^{\mathrm{T}}$, where x is a given p-vector. Although the *ab initio* calculation of S requires $O(p^3)$ arithmetic operations, it turns out that S can be computed from R and x in $O(p^2)$ operations, a process that is known as *updating.* The usual updating algorithm is numerically stable.

The inverse process of computing the Cholesky decomposition $R^{\mathrm{T}}R$ of $A = B - x^{\mathrm{T}}x$ from that of B is called *downdating.* Three algorithms for downdating have appeared in the literature: Chambers' algorithm [4], the LINPACK algorithm [5] (due to Michael Saunders), and the method of plane hyperbolic transformations (which in another guise is due Golub [8]).[1] Although Chambers' algorithm and

[1]A close reading of the papers involved suggests that Chambers thought he was merely

the LINPACK algorithms are not stable in the usual backward sense, it has been shown [11, 3] to have an important property, which we will call *relational stability.* Specifically, the mathematical relations that hold between the true quantities, continue to hold for the computed quantities provided they are perturbed slightly. We will show that relational stability is preserved in a sequence of updates and downdates. This, combined with the block perturbation theory of Eldén and Park [6], implies that if the final result of sequence of updates and downdates is well conditioned then it will be computed accurately.

The method of plane hyperbolic transformations is not relationally stable. Consequently, as we will show by an example, it can introduce unnecessary errors in the course of a sequence of updates and downdates.

Rank degenerate problems usually require a decomposition that reveals the rank and provides a basis for the null space of the matrix in question. Two-sided orthogonal-triangular, such as the URV and ULV decompositions [13, 15], perform these functions and in addition can be efficiently updated and downdated. We will show that the relational stability of the updating and downdating algorithms extends to these algorithms. In particular, if the nondegenerate part of the matrix is well conditioned then the basis for the null space is accurately computed.

This paper is organized as follows. In the next section we sketch the results of rounding error analyses for the various algorithms. In §3 we review the perturbation theory for the Cholesky decomposition. In §4 we establish the relational stability of a sequence of updates and downdates and derive error bounds for the results. Section 5 is devoted to an example illustrating the results of the previous section. In §6, we derive bounds for URV updating. The paper concludes with some observations on downdating and exponential windowing.

Throughout the paper, $\|A\|$ will denote the Frobenius norm of the matrix A, which is defined by

$$\|A\|^2 = \sum_{i,j} a_{ij}^2.$$

The quantity $\|x\|$ is the ordinary Euclidean norm of the vector x. For more on norms see [9].

rederiving Golub's hyperbolic algorithm, since he explicitly attributes the algorithm to Golub. However, his derivation resulted in a different formula for one of the downdated quantities — in effect a different algorithm with different properties. Mention should also be made of the method of corrected semi-normal equations in [1]. However, this method differs from the others in that it uses all the original data contained in R, and is therefore expensive when R contains many updates.

2. Rounding Error Analyses

In this section we will review the rounding error analyses of updating by plane rotations and downdating by Chambers' and the LINPACK algorithms.

The updating algorithm in general use is due to Bogert and Burris [2] and Golub [7]. The idea behind the algorithm is to compute an orthogonal matrix Q such that

$$Q^{\mathrm{T}} \begin{pmatrix} R \\ x^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} S \\ 0 \end{pmatrix},$$

where S is upper triangular. It then follows from the orthogonality of Q that

$$R^{\mathrm{T}} R + x x^{\mathrm{T}} = S^{\mathrm{T}} S,$$

so that S is the Cholesky factor of $R^{\mathrm{T}} R + x x^{\mathrm{T}} = A + x x^{\mathrm{T}}$.

The algorithm is stable in the backward sense. The very general rounding-error of plane rotations by Wilkinson [17, p. 131 ff.] applies to give the following result. If we let S denote the *computed* matrix, then there is an orthogonal matrix Q and a $(p+1) \times p$ matrix F satisfying

$$\|F\| \leq K \|S\| \epsilon_{\mathrm{M}}$$

such that

$$Q^{\mathrm{T}} \left[\begin{pmatrix} R \\ x^{\mathrm{T}} \end{pmatrix} + F \right] = \begin{pmatrix} S \\ 0 \end{pmatrix}. \tag{2.1}$$

Here ϵ_{M} is the rounding unit for the machine in question and K is a constant that depends on p and the details of the computer arithmetic. Thus the computed result, however inaccurate, comes from a slightly perturbed problem.

In exact arithmetic, both Chambers' algorithm and the LINPACK algorithm produce an orthogonal Q matrix such that

$$Q^{\mathrm{T}} \begin{pmatrix} S \\ 0 \end{pmatrix} = \begin{pmatrix} R \\ x^{\mathrm{T}} \end{pmatrix}.$$

It follows that

$$R^{\mathrm{T}} R = S^{\mathrm{T}} S - x x^{\mathrm{T}}$$

Thus R is the Cholesky factor of the matrix $S^{\mathrm{T}} S - x x^{\mathrm{T}} = B - x x^{\mathrm{T}}$.

For both downdating algorithms it has been shown [11, 3] that if R denotes the computed matrix then there is an orthogonal matrix Q and a $(p+1) \times p$ matrix E satisfying

$$\|E\| \leq K \|S\| \epsilon_{\mathrm{M}} \tag{2.2}$$

such that

$$Q^{\mathrm{T}}\begin{pmatrix} S \\ 0 \end{pmatrix} = \begin{pmatrix} R \\ x^{\mathrm{T}} \end{pmatrix} + E. \tag{2.3}$$

This result is not backward stability, since it is not possible to concentrate the entire error in the matrix S and the vector x^{T}. Instead we will call it *relational stability* because the defining mathematical relation between the true quantities continues to be satisfied up to a small error by the computed quantities. We will see later that relational stability has important consequences for the accuracy of the computed results.

Note that equation (2.1) can be brought into the form (2.3) by defining $E = -Q^{\mathrm{T}}F$. It is this common form that we will use to treat sequential updates and downdates.

The method of hyperbolic transformations is neither backward or relationally stable. The unhappy consequences of this fact will be seen in §5.

3. Perturbation Theory

The error analyses of updating and downdating say that the true result can be obtained from the computed result by perturbing its cross-product matrix slightly and computing the Cholesky factor. To find out how accurate the result actually is, we must call on perturbation theory.

The perturbation theory for Cholesky decompositions has been studied in a number of places. Since here we are concerned with small perturbations, we will give an asymptotic result that is sharp up to second order terms in the error [14].

Theorem 3.1. *Let A be positive definite, and let $\tilde{A} = A + H$, where H is symmetric. Then for all sufficiently small H, $\tilde{A}$ is positive definite. If R is the Cholesky factor of A and $\tilde{R}$ is the Cholesky factor of $\tilde{A}$, then*

$$\frac{\|\tilde{R} - R\|}{\|R\|} \lesssim \frac{\|R^{-1}\|^2}{\sqrt{2}}\|H\|. \tag{3.1}$$

Note that this result puts an inherent limit on the accuracy we can expect in a computed Cholesky factor. For example, if we merely round the elements of A, then

$$\|H\| \leq \|A\|\epsilon_{\mathrm{M}} \leq \|R\|^2\epsilon_{\mathrm{M}},$$

where ϵ_{M} is the rounding unit. It follows that

$$\frac{\|\tilde{R}-R\|}{\|R\|} \lesssim \frac{\kappa^2(R)}{\sqrt{2}}\epsilon_{\mathrm{M}}, \tag{3.2}$$

where $\kappa(R) = \|R\|\|R^{-1}\|$ is the condition number of R. It would be unfair to expect an algorithm to produce a result more accurate than the right hand side of (3.2).

If A and R are partitioned in the forms

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and

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

where A_{11} and R_{11} are of order k, then the Cholesky factor of A_{11} is R_{11}. The perturbation analysis above shows that the accuracy of $\tilde{R}_{11}$ depends not on the condition of R but on the condition of R_{11}. Thus the Cholesky factor of a well-conditioned leading principal submatrix of A will by insensitive to perturbations, even though A as a whole may be ill conditioned: the large errors end up in the terminal columns of R. We will use this fact in analyzing URV decompositions.

4. Sequential Updating

In this section we will show that a sequence of relationally stable updates and downdates is relationally stable. We will begin by considering a single downdate followed by an update.

Let R_0 be the matrix to be downdated and let x_0 be the vector to be removed. Let the computed result be R_1. Similarly, let R_1 be updated by the vector x_1 to give R_2. Then by the rounding error analyses just cited, there is an orthogonal matrix Q_0 and a small matrix E_1 such that

$$Q_0 \begin{pmatrix} R_0 \\ 0 \\ x_1^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} R_1 \\ x_0^{\mathrm{T}} \\ x_1^{\mathrm{T}} \end{pmatrix} + E_1$$

Similarly there is an orthogonal matrix Q_1 and a small matrix E_2 such that

$$Q_1^{\mathrm{T}} \begin{pmatrix} R_1 \\ x_0^{\mathrm{T}} \\ x_1^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} R_2 \\ x_0^{\mathrm{T}} \\ 0 \end{pmatrix} + E_2.$$

If we set

$$Q^{\mathrm{T}} = Q_1^{\mathrm{T}} Q_0$$

and

$$E = Q_1^{\mathrm{T}} E_1 + E_2$$

then

$$Q^{\mathrm{T}} \begin{pmatrix} R_0 \\ 0 \\ x_1^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} R_2 \\ x_0^{\mathrm{T}} \\ 0 \end{pmatrix} + E.$$

Thus a downdate followed by an update is stable and the norm of the error

$$\|E\| \le \|E_0\| + \|E_1\|$$

is bounded by the sum of the norms of the errors in the individual steps.

This analysis clearly extends to any sequence of n updates and downdates. Specifically, collect the vectors appearing in updates in the matrix $X_{\mathrm{u}}^{\mathrm{T}}$ and the vectors appearing in downdates in the $X_{\mathrm{d}}^{\mathrm{T}}$. Then there is an orthogonal transformation Q and a matrix E such that

$$Q \begin{pmatrix} R_0 \\ 0 \\ X_{\mathrm{u}}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} R_n \\ X_{\mathrm{d}}^{\mathrm{T}} \\ 0 \end{pmatrix} + E. \tag{4.1}$$

The norm of the error E is bounded by the sum of of the norms of the backward errors in the individual updates and downdates.

To derive a specific bound for the error, we note that the error bound (2.2) for updating and downdating involve computed Cholesky factors. Consequently, if we let

$$\rho = \max\{\|R_i\| : i = 1, \ldots, k, \},$$

then a common bound all the errors E_k is $K\rho\epsilon_{\mathrm{M}}$. It follows that the error in (4.1) is bounded by

$$\|E\| \le nK\rho\epsilon_{\mathrm{M}}. \tag{4.2}$$

To assess the accuracy of R_n, we do a block perturbation analysis in the spirit of Eldén and Park [6]. Specifically, from (4.1) it follows that

$$R_n^{\mathrm{T}} R_n = R_0^{\mathrm{T}} R_0 + X_{\mathrm{u}} X_{\mathrm{u}}^{\mathrm{T}} - X_{\mathrm{d}} X_{\mathrm{d}}^{\mathrm{T}} + H.$$

If we set

$$\hat{\rho} = \max\left\{\rho, \left\|\begin{pmatrix} R_0 \\ X_{\mathrm{d}}^{\mathrm{T}} \end{pmatrix}\right\|\right\}.$$

and assume that

$$nK\epsilon_{\mathrm{M}} < 1,$$

then

$$\|H\| = \|E\|(2 + \|E\|)\left\|\begin{pmatrix} R_0 \\ X_{\mathrm{d}}^{\mathrm{T}} \end{pmatrix}\right\| \leq 3nK\epsilon_{\mathrm{M}}\hat{\rho}^2, \tag{4.3}$$

It now follows from (3.1) that if S_n is the Cholesky factor of $R_0^{\mathrm{T}}R_0 + X_{\mathrm{u}}X_{\mathrm{u}}^{\mathrm{T}} - X_{\mathrm{d}}X_{\mathrm{d}}^{\mathrm{T}}$ (i.e., the true Cholesky factor) then

$$\frac{\|S_n - R_n\|}{\|R_n\|} \lesssim \sqrt{4.5}nK\hat{\rho}^2\|R_n^{-1}\|^2\epsilon_{\mathrm{M}} \tag{4.4}$$

This bound is quite crude and no doubt can be refined. However, it already tells us that if $\hat{\rho}^2\|R_n\|^2$ is not large, the computed Cholesky factor will be a good approximation to the true one, *no matter how inaccurate the intermediate quantities may be.* The factor R_n^{-1} will be large when R is ill-conditioned. The factor $\hat{\rho}$ is essentially the norm of the matrix one would get if all the updates but none of the downdates were performed. If all the rows of $X_{\mathrm{u}}^{\mathrm{T}}$ are of a size, then $\hat{\rho}$ can be expected to grow like $\sqrt{n}$. However, if even one row is very much larger than the others, the bound tells us to expect a persisting inaccuracy in the subsequent computed Cholesky factors. This phenomena has been observed in [1].

5. A Numerical Example

To illustrate the the above results we will give a numerical example in which a downdate from a well-conditioned matrix R_0 to an ill-conditioned matrix R_1 is followed by an update to a well conditioned matrix R_2. The calculations were performed in `matlab` with a rounding unit of $2 \cdot 10^{-16}$.

The following is a description of the experiment. The idea is generate an ill-conditioned matrix R_1 and create R_0 and R_1 by updating it.

1. Let R_1 be the R-factor from the QR-factorization of a matrix of independent normal random variables with mean zero and variance one. This will produce a well conditioned matrix.

2. Set the $(2,2)$-element of R_1 to 10^{-7} to produce an ill-conditioned R-factor.

3. Let x be a random normal vector and update R_1 and x to get the matrix R_0.

4. Let y be a random normal vector and update R_1 and y to get the matrix R_2.

5. Let R_{l1} be the result of using the LINPACK algorithm to downdate R_0 and x. Let R_{l2} be the result of updating R_{l1} and y.

6. Let R_{c1} be the result of using Chambers' algorithm to downdate R_0 and x. Let R_{c2} be the result of updating R_{c1} and y.

7. Let R_{h1} be the result of using plane hyperbolic transformations to downdate R_0 and x. Let R_{h2} be the result of updating R_{h1} and y.

Table 5.1 gives the result of twenty repetitions (steps 1–6 above) of this procedure for $p = 5$. The asterisks indicated cases where the hyperbolic downdating could not be carried out.

The results are entirely consistent with theory. Since R_1 is ill conditioned, any attempt to compute it by downdating a well-condition matrix must result in inaccuracies proportional to the square of the condition number. All the algorithms exhibit these inaccuracies. The difference between the algorithms becomes apparent when we examine the errors in the approximations to R_2. Here the two relationally stable algorithms restore almost full accuracy, while the hyperbolic algorithm looses several figures. However, not all of the error in R_{h1} is carried forward to R_{h2}: presumably some component of the error introduced by the hyperbolic rotations can be accounted for by relational perturbations, a point which deserves further study.

In three cases the hyperbolic downdate fails when a quantity that should be positive turns out negative. In all cases the other algorithms go through to completion. However, this comparison is a little unfair to the hyperbolic approach. The condition numbers of the matrices R_1 in Table 5.1 are on the order of 10^8, close to the point where the perturbation theory predicts no accuracy for the computed results. If we make R_2 even a little more ill conditioned, Chambers' algorithm begins to fail.[2] Decrease the condition number a little, and all algorithms go through to completion.

[2]The LINPACK algorithm continues to perform well, but this is an artifact of the simplicity of the example and special properties of the algorithm. In more realistic settings, the LINPACK algorithm would also fail.

$$\rho_{\mathrm{p}i} = \frac{\|R_{\mathrm{p}i} - R_i\|}{\|R_i\|}$$

$\kappa(R_1)$	$\kappa(R_2)$	$\rho_{\mathrm{l}1}$	$\rho_{\mathrm{l}2}$	$\rho_{\mathrm{c}1}$	$\rho_{\mathrm{c}2}$	$\rho_{\mathrm{h}1}$	$\rho_{\mathrm{h}2}$
1e+08	1e+01	4e−02	3e−15	1e−02	1e−15	2e−02	2e−10
2e+08	1e+02	8e−02	1e−13	1e−01	5e−13	*	*
2e+08	5e+00	1e−03	1e−16	3e−03	1e−16	7e−05	1e−10
4e+07	3e+00	7e−03	2e−16	8e−03	1e−16	3e−05	2e−10
1e+08	5e+00	3e−04	2e−16	2e−04	1e−16	9e−04	1e−11
7e+07	5e+00	4e−03	1e−16	7e−03	1e−16	8e−03	1e−10
6e+08	4e+00	7e−02	3e−16	1e−01	4e−16	*	*
1e+08	4e+01	3e−03	2e−16	1e−03	1e−16	5e−03	1e−10
1e+08	4e+00	6e−02	1e−16	1e−02	1e−16	1e−02	2e−10
4e+08	1e+01	4e−01	9e−16	3e−01	3e−16	*	*
1e+09	1e+01	1e−02	2e−16	1e−02	1e−16	*	*
7e+07	1e+01	1e−03	8e−16	8e−03	4e−16	2e−03	7e−10
6e+07	1e+01	5e−03	3e−16	7e−03	4e−16	8e−03	6e−10
6e+07	4e+01	1e−02	4e−15	4e−04	4e−16	1e−02	2e−10
5e+07	1e+01	5e−04	1e−16	6e−04	1e−16	5e−04	3e−11
1e+08	1e+01	1e−02	3e−16	9e−03	2e−16	6e−03	1e−09
3e+08	8e+00	3e−02	3e−16	7e−03	3e−16	2e−02	7e−11
2e+08	1e+01	8e−02	6e−16	6e−02	3e−16	8e−02	2e−10
1e+10	4e+02	2e−02	1e−14	1e−02	2e−15	3e−02	2e−10
6e+07	8e+01	9e−04	4e−16	1e−03	6e−16	1e−05	1e−11

Table 5.1: A Downdate Followed by an Update

6. URV Decompositions

In this section we will apply our results to sequential updates and downdates of URV decompositions. A URV decomposition of a matrix X is a decomposition of the form

$$U^{\mathrm{T}} X V = \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where U and V are orthogonal and R is upper triangular. Any matrix has infinitely many URV decompositions. One of them, the singular value decomposition (R diagonal), is widely used because it exhibits approximate rank degeneracies in X

and provides an orthonormal basis for an approximate null space of the matrix. However, it cannot be efficiently updated or downdated.

Rank-revealing URV decompositions overcome the computational deficiencies of the singular value decomposition. Suppose that X has been obtained from a matrix of exactly rank k by perturbing it by some noise. (We use the term "noise" rather than "error" to distinguish the perturbation from effects due to rounding error.) Then there is a URV decomposition in which R takes the form

$$R = \begin{pmatrix} T & F \\ 0 & G \end{pmatrix}$$

where T is a well-conditioned conditioned matrix of order k and F and G are the same size as the noise (F may actually be much smaller, even zero). The virtues of a rank-revealing URV decomposition are that it can be updated and downdated. Moreover, if V is partitioned in the form

$$V = (V_1 \ \ V_2),$$

then V_1 and V_2 provide orthonormal bases for approximate row and null space of R.

Although the updating and downdating algorithms are quite complicated—they involve decisions about rank and procedures for keeping the small part of the decomposition small—nontheless they fall within the purview of the analyses discussed above. Specifically, if the LINPACK or Chambers' algorithm is used to perform downdates, there are orthogonal matrices U and V such that the computed R_n satisfies

$$U^{\mathrm{T}} \begin{pmatrix} R_0 \\ 0 \\ X_{\mathrm{u}}^{\mathrm{T}} \end{pmatrix} V = \begin{pmatrix} R_n \\ X_{\mathrm{d}}^{\mathrm{T}} V \\ 0 \end{pmatrix} + E, \tag{6.1}$$

where as above $\|E\| \leq nK\rho\epsilon_{\mathrm{M}}$ [cf. (4.2)].

In interpreting this bound, there are two questions we can ask. One question is, "How accurate is V?" Actually, this question is not well posed, since there is no unique URV decomposition associated with the data. We can, however, show that the V-factor of any URV decomposition satisfying a relation like (6.1) must produce approximate null spaces that lie near that produced by V (see the appendix to this paper).

But there is a simpler alternative. For *any* V, there is a unique URV decomposition of R_0 that is obtained by computing the Cholesky decomposition of

$V^{\mathrm{T}}(R_0^{\mathrm{T}}R_0 + X_{\mathrm{u}}X_{\mathrm{u}}^{\mathrm{T}} - X_{\mathrm{d}}X_{\mathrm{d}}^{\mathrm{T}})V$. Now the URV algorithm does not compute the Cholesky decomposition of this matrix; instead it computes the Cholesky decomposition of $V^{\mathrm{T}}(R_0^{\mathrm{T}}R_0+X_{\mathrm{u}}X_{\mathrm{u}}^{\mathrm{T}}-X_{\mathrm{d}}X_{\mathrm{d}}^{\mathrm{T}}+H)V$, where H satisfies (4.3), and it is from this decomposition that we deduce that that we have revealed the rank. Thus, if this decomposition is accurate, V truly furnishes a basis for an approximate null space. Thus the second question is, "How accurate is R_n?"

Here we are on familiar territory. If the matrix T_n is well conditioned, by the comments at the end of §3 it will be accurately computed. The matrices G_n and F_n, which consist of noise, will be less accurately computed. However, R_n^{-1} will be approximated by G_n^{-1}, so that the factor $\hat{\rho}R_n^{-1}$ in (4.4) can be regarded as a signal-to-noise ratio. If this ratio is substantially above $\sqrt{\epsilon_{\mathrm{M}}}$, then F and G will be computed with reasonable accuracy. Specific bounds may be obtained as above.

It should not be thought that V is near the matrix that would have been obtained by exact computation. The algorithm for determining rank involves discrete decisions, and if rounding error causes a change in any of these decisions, the computed decomposition will diverge sharply from the exact one. Nonetheless, by the analysis sketched above, we will have computed a rank-revealing URV decomposition.

One final point. The matrix V in (6.1) is defined as the exact product of the rotations computed in the course of the sequential updates and downdates. The computed V, being contaminated with rounding error, will diverge from the original. However, this divergence will be very slow and corresponds to the factor n in (4.3).

7. Conclusions

Downdating has had bad press in some circles. Part of it is no doubt due to unfortunate experiences with bad algorithms, such as hyperbolic downdating. However, a great deal of it is the result of not understanding the limitations of both updating and downdating.

An extremely simple example will illustrate the problems. Let R be the scalar 1, and suppose that in ten-digit decimal floating-point arithmetic we wish to incorporate $x = 5 \cdot 10^{-6}$; that is we wish to update

$$\begin{pmatrix} 1 \\ 5 \cdot 10^{-6} \end{pmatrix}.$$

The exact update is $1 + 2.5 \cdot 10^{-11}$. The computed update will be 1. There is no trace of the number $5 \cdot 10^{-6}$; it has been swallowed by the update, and a subsequent downdate cannot recover it. Thus, downdating is sometimes blamed for inaccuracies that are implicit in the updating procedure.

However, downdating has limitations of its own. If for example, the computed update is perturbed (as in real life it might be by rounding error) to become 1.000000001, then the computed downdate will be about $3.2 \cdot 10^{-5}$. This is inaccurate, as we would expect; but if a relationally stable algorithm is used the unaccuracy will go away on subsequent updates. Something worse happens when the problem is perturbed to become 0.9999999999. Now the downdating process fails completely, and there is no chance to regain accuracy in a subsequent update.

The lesson is that when the condition numbers of the triangular factors approach $1/\sqrt{\epsilon_{\mathrm{M}}}$, both updating and downdating become problematical. But move a little off, and relationally stable algorithms will perform well. When inaccuracies are inherent in the problem, they will, of course, produce inaccurate answers; but well-conditioned R-factors will be computed accurately.

In some applications exponential windowing is an alternative to downdating. In this method, the matrix R is multiplied by a factor $\beta < 1$ before each update, which damps influence of older updates. Now when the sequence of vectors x_i^{T} represents a stationary process, exponential windowing is to be preferred to downdating. It is simpler and has better numerical properties [10, 12]. However, in nonstationary situations, the two techniques will produce different R-factors, so that they are not just different numerical algorithms computing the same thing. In this case, the decision between the two must depend on their behavior in the application in question. An important contribution of this paper, then, is to show when numerical considerations need not enter into this decision.

Acknowledgements

Parts of this paper were inspired by the block perturbation analysis of Park and Eldén [6], which showed how to avoid intermediate quantities in assessing the accuracy of the final result. And many thanks to Haesun Park for her useful comments at every stage of this research.

Appendix

Recall that a computed URV decomposition satisfies

$$U^{\mathrm{T}} \begin{pmatrix} R_0 \\ 0 \\ X_{\mathrm{u}}^{\mathrm{T}} \end{pmatrix} V = \begin{pmatrix} R_n \\ X_{\mathrm{d}}^{\mathrm{T}} V \\ 0 \end{pmatrix} + E, \tag{A.1}$$

where U and V are orthogonal, and E satisfies the bound $\|E\| \leq nK\rho\epsilon_{\mathrm{M}}$. The matrix R_n will be rank revealing if it has the form

$$\begin{pmatrix} T_n & F_n \\ 0 & G_n \end{pmatrix},$$

where

$$\|T_n^{-1}\|\|G_n\|, \|T_n^{-1}\|\|G_n\| \ll 1. \tag{A.2}$$

Now let

$$\bar{U}^{\mathrm{T}} \begin{pmatrix} R_0 \\ 0 \\ X_{\mathrm{u}}^{\mathrm{T}} \end{pmatrix} \bar{V} = \begin{pmatrix} R_n \\ X_{\mathrm{d}}^{\mathrm{T}} \bar{V} \\ 0 \end{pmatrix} + \bar{E}, \tag{A.3}$$

where $\bar{U}$ and $\bar{V}$ are orthogonal and $\bar{E}$ satisfies the same bound. We are going to show that if

$$V = (V_1 \ \ V_2) \quad \text{and} \quad \bar{V} = (\bar{V}_1 \ \ \bar{V}_2)$$

and we set

$$W = \bar{V}^{\mathrm{T}} V = \begin{pmatrix} \bar{V}_1^{\mathrm{T}} V_1 & \bar{V}_1^{\mathrm{T}} V_2 \\ \bar{V}_2^{\mathrm{T}} V_1 & \bar{V}_2^{\mathrm{T}} V_2 \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}$$

then W_{12} is small. Note that this implies that the space $\mathcal{R}(\bar{V}_1)$ spanned by the columns of $\bar{V}_1$ is almost orthogonal $\mathcal{R}(V_2)$. Since $\mathcal{R}(V_1)$ is exactly orthogonal $\mathcal{R}(V_2)$, it follows that $\mathcal{R}(V_1)$ and $\mathcal{R}(\bar{V}_1)$ are in some sense near each other.

More precisely, we will show that $\|W_{12}\|_2$ is small, where $\|W_{12}\|_2$ the spectral norm of W_{12}—the largest singular value of W_{12}. This number is also the sine of the largest canonical angle be between $\mathcal{R}(V_1)$ and $\mathcal{R}(V_2)$ (see [16, Ch. 1]).

We begin with a lemma.

Lemma A.1. *Let*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^{\mathrm{T}} & A_{22} \end{pmatrix}$$

be positive definite and let

$$W = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}$$

be orthogonal. Suppose that

$$\|A - W^{\mathrm{T}}AW\|_2 \le \epsilon$$

and that

$$\eta \equiv \|A_{11}^{-1}\|(\epsilon + 2\|A_{12}\| + \|A_{22}\|) < \frac{1}{2}. \tag{A.4}$$

Then

$$\|W_{12}\|_2^2 \le \frac{1 - \sqrt{1 - 4\eta^2}}{2} < 2\eta^2. \tag{A.5}$$

Proof. The $(1,2)$ block of $W^{\mathrm{T}}AW$ is $W_{11}^{\mathrm{T}}A_{11}W_{12} + W_{11}^{\mathrm{T}}A_{12}W_{22} + W_{21}^{\mathrm{T}}A_{12}^{\mathrm{T}}W_{12} + W_{21}^{\mathrm{T}}A_{22}W_{22}$. Hence

$$\epsilon \ge \|W_{11}^{\mathrm{T}}A_{11}W_{12}\|_2 - \|W_{11}^{\mathrm{T}}A_{12}W_{21}\|_2 - \|W_{21}^{\mathrm{T}}A_{12}^{\mathrm{T}}W_{12}\|_2 - \|W_{21}^{\mathrm{T}}A_{22}W_{22}\|_2.$$

Since $\|W_{ij}\|_2 \le 1$,

$$\|W_{11}^{\mathrm{T}}A_{11}W_{12}\|_2 \le \epsilon + 2\|A_{12}\|_2 + \|A_{22}\|_2,$$

and since

$$\|W_{11}^{\mathrm{T}}A_{11}W_{12}\|_2 \ge \frac{\|W_{12}\|_2}{\|A_{11}^{-1}\|_2\|W_{11}^{-1}\|_2},$$

$$\frac{\|W_{12}\|_2}{\|W_{11}^{-1}\|_2} \le \eta.$$

By the orthogonality of W, we have

$$W_{11}^{\mathrm{T}}W_{11} + W_{12}^{\mathrm{T}}W_{12} = I.$$

Thus, if $\omega = \|W_{12}\|_2$ is the largest singular value of W_{12}, then $\sqrt{1-\omega^2}$ is the smallest singular value of W_{11}. Hence $\|W_{11}^{-1}\|_2^{-1} = \sqrt{1-\omega^2}$, and

$$\omega^2(1 - \omega_2) \le \eta^2.$$

Thus ω^2 is the smallest root of the quadratic equation $\omega^4 - \omega^2 + \eta = 0$, which gives (A.5). ■

Now from (A.1) and (A.3)

$$\begin{aligned} R_0^{\mathrm{T}} R_0 + X_{\mathrm{u}}^{\mathrm{T}} X_{\mathrm{u}} - X_{\mathrm{d}}^{\mathrm{T}} &= V R_n^{\mathrm{T}} R_n V^{\mathrm{T}} + V E^{\mathrm{T}} E V^{\mathrm{T}} \\ R_0^{\mathrm{T}} R_0 + X_{\mathrm{u}}^{\mathrm{T}} X_{\mathrm{u}} - X_{\mathrm{d}}^{\mathrm{T}} &= \bar{V} R_n^{\mathrm{T}} R_n \bar{V}^{\mathrm{T}} + \bar{V} \bar{E}^{\mathrm{T}} \bar{E} \bar{V}^{\mathrm{T}} \end{aligned}$$

Set

$$A = R_n^{\mathrm{T}} R_n = \begin{pmatrix} T_n^{\mathrm{T}} T_n & T_n^{\mathrm{T}} F_n \\ 0 & F_n^{\mathrm{T}} F_n + G_n^{\mathrm{T}} G_n \end{pmatrix}$$

$H = V E^{\mathrm{T}} E V^{\mathrm{T}}$, $\bar{H} = \bar{V} \bar{E}^{\mathrm{T}} \bar{E} \bar{V}^{\mathrm{T}}$, and $W = \bar{V}^{\mathrm{T}} V$. Then it follows that

$$W^{\mathrm{T}} A W - A = W^{\mathrm{T}} \bar{H} W - H.$$

If

$$\epsilon = \|H\| + \|\bar{H}\|$$

and η, defined by (A.4), defined by (A.4) is less than $\frac{1}{2}$, then

$$\|\bar{V}_1^{\mathrm{T}} V_2\|_2 < 2\eta.$$

It is instructive to bound η. Since $\|H\|$ and $\|\bar{H}\|$ satisfy (4.3), we have

$$\eta \leq (6nK\hat{\rho}^2 \epsilon_{\mathrm{M}} + 2\|F_n\|\|\|T_n\| + \|F_n^{\mathrm{T}} F_n + G_n^{\mathrm{T}} G_n\|)\|T_n^{-1}\|^2.$$

The first term $6nK\hat{\rho}^2\|T_n^{-1}\|^2\epsilon_{\mathrm{M}}$, which represents the contribution of rounding error, is precisely the term that must be small for R_n to be computed accurately. The term $\|F_n^{\mathrm{T}} F_n + G_n^{\mathrm{T}} G_n\|\|T_n^{-1}\|^2$ is small by virtue of (A.2). If we write the middle term in the form

$$2\kappa(T_n)\|T_n^{-1}\|\|F_n\|$$

we see that, (A.2) notwithstanding, this term is potentially larger than the others. Now the algorithm for updating URV decompositions contains a refinement step that is specifically designed to make F_n small. The above analysis suggests that such a step is fully justified.

References

[1] Å. Björck, H. Park, and L. Eldén. Accurate downdating of least squares solutions. Department of Mathematics, Linköping University. To appear in *SIAM Journal on Matrix Analysis and Applications*, 1992.

[2] D. Bogert and W. R. Burris. Comparison of least squares algorithms. Report ORNL-3499, Neutron Physics Division, Oak Ridge National Laboratory, 1963. Vol. 1, Sec. 5.5.

[3] A. Bojanczyk, R. P. Brent, P. Van Dooren, and F. de Hoog. A note on downdating the Cholesky factorization. *SIAM Journal on Scientific and Statistical Computing*, 8:210–221, 1987.

[4] J. M. Chambers. Regression updating. *Journal of the American Statistical Association*, 66:744–748, 1971.

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

[6] L. Eldén and H. Park. Perturbation analysis for block downdating of a Cholesky decomposition. Technical Report LiTH-MAT-R-93-26, Department of Mathematics, Linköping Univeristy, 1993.

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

[8] G. H. Golub. Matrix decompositions and statistical computation. In R.C. Milton and J. A. Nelder, editors, *Statistical Computation*, pages 365–397, New York, 1969. Academic Press.

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

[10] M. Moonen. *Jacobi-Type Updating Algorithms for Signal Processing, Systems Identification and Control*. PhD thesis, Katholieke Universiteit Leuven, 1990.

[11] G. W. Stewart. The effects of rounding error on an algorithm for downdating a Cholesky factorization. *Journal of the Institute for Mathematics and Applications*, 23:203–213, 1979.

[12] G. W. Stewart. Error analysis of QR updating with exponential windowing. *Mathematics of Computation*, 59:135–140, 1992.

[13] G. W. Stewart. An updating algorithm for subspace tracking. *IEEE Transactions on Signal Processing*, 40:1535–1541, 1992.

[14] G. W. Stewart. On the perturbation of LU, Cholesky, and QR factorizations. *SIAM Journal on Matrix Analysis and Applications*, 14:1141–1146, 1993.

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

[16] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory.* Academic Press, Boston, 1990.

[17] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, England, 1965.