ACCURATE DOWNDATING OF LEAST SQUARES SOLUTIONS

By

Å. Björck
H. Park
and
L. Eldén

IMA Preprint Series # 947
April 1992

Accurate Downdating of Least Squares Solutions

Å. Björck*, H. Park†, and L. Eldén*

March 5, 1992

Abstract

Solutions to a sequence of modified least squares problems, where either a new observation is added (updating) or an old observation is deleted (downdating), are required in many applications. Stable algorithms for downdating can be constructed if the complete QR factorization of the data matrix is available. Algorithms which only downdate R and do not store Q require less operations. However, they do not give good accuracy and may not recover accuracy after an ill-conditioned problem has occurred. We describe a new algorithm for accurate downdating of least squares solutions, and compare it to existing algorithms. We also present numerical test results using the sliding window method, where a number of updatings and downdatings occur repeatedly.

Key words. downdating, iterative refinement, least squares, seminormal equations

1 Introduction

Many problems in signal processing can be formulated as a least squares problem

$$\min_w \|Xw - s\|_2, \quad X \in \mathbf{R}^{p\times n}, \quad p > n. \tag{1}$$

*Department of Mathematics, Linköping University, S-581 83, Linköping, Sweden, (akbjo@math.liu.se and elden@math.liu.se, respectively). This work was supported in part by the Institute of Mathematics and its Applications, Univ. of Minnesota, with funds provided by the National Science Foundation.

†Computer Science Department, University of Minnesota, Minneapolis, MN 55455, U.S.A. (hpark@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grant CCR-8813493.

If $rank(X) = n$ and the QR decomposition of the data matrix $(X\ s)$ is

$$Q^T(X\ s) = \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{p\times(n+1)}, \tag{2}$$

where $Q \in \mathbf{R}^{p\times p}$, then the least squares solution w is obtained from

$$Rw = u, \tag{3}$$

and the residual vector r and its norm satisfy

$$r = s - Xw, \qquad \|r\|_2 = |\rho|.$$

Frequently one knows the factorization in (2), and wishes to find the solution to a modified problem

$$\min_w \|\tilde{X}w - \tilde{s}\|_2,$$

where a new observation $(y^T\ \eta)$ is added (**updating**):

$$\tilde{X} = \begin{pmatrix} X \\ y^T \end{pmatrix}, \quad \tilde{s} = \begin{pmatrix} s \\ \eta \end{pmatrix},$$

or an old observation $(z^T\ \sigma)$ is removed (**downdating**):

$$X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix}, \quad s = \begin{pmatrix} \sigma \\ \tilde{s} \end{pmatrix}.$$

Often the modified problem involves both an updating and a downdating. From (3), we see that the solution to the modified problem can be obtained by modifying the R factor of the corresponding augmented matrix $(\tilde{X}\ \tilde{s})$. If R and $\tilde{R}$ are the R factors of X and $\tilde{X}$ respectively, then we have for updating

$$\tilde{R}^T\tilde{R} = R^TR + yy^T,$$

and for downdating

$$\tilde{R}^T\tilde{R} = R^TR - zz^T.$$

Throughout this paper, we will assume that the data matrices X and $\tilde{X}$ have full column rank. Hence the problem is mathematically (but not numerically) equivalent to that of updating or downdating a Cholesky factorization under a rank one perturbation. From the relation $\sigma_i^2(A) = \lambda_i(A^TA)$ and classical perturbation theory for eigenvalues [9], it follows that the singular values $\tilde{\sigma}_i = \sigma_i(\tilde{R})$ interleave with $\sigma_i = \sigma_i(R)$, where for downdating

$$\sigma_1 \geq \tilde{\sigma}_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq \tilde{\sigma}_n \geq 0.$$

In downdating, the smallest singular value may decrease and we can have $\tilde{\sigma}_n \approx 0$, even when R has full column rank. Moreover, any singular value may decrease by a considerable amount, which indicates that downdating can be a sensitive problem [16]. On the other hand, updating R will increase all its singular values.

Important applications where the recursive least squares problems arise include speech echo cancellation, speech coding, and adaptive radar signal processing. The following issues are critical for these applications [1]:

1. The modification should be performed with *as few operations (real time applications) and as little storage requirement as possible.* Recomputing the QR factorization is too costly since it requires $O(pn^2)$ operations, and thus, a modification technique must be used.

2. The solution should be accurate up to the limitations of data and conditioning of the problem, i.e. a *stable numerical method* must be used. It should be possible to use a computer with *short word-length.* This rules out the use of the method of normal equations, which requires twice the word-length as methods based on the QR decomposition.

The purpose of this paper is to discuss accurate and efficient algorithms for downdating least squares solutions. We consider the LINPACK algorithm and indicate that it does not give an accurate solution when the downdating problem is ill-conditioned. Then we discuss more accurate algorithms: the downdating algorithm based on Gram-Schmidt orthogonalization, and an algorithm based on corrected seminormal equations.

The paper is organized as follows. In Section 2, we review the algorithms for updating and downdating the QR decomposition when both Q and R factors are available. In Section 3, the downdating algorithm based on the Gram-Schmidt orthogonalization method is summarized. In Sections 4 and 5, the LINPACK algorithm for downdating the Cholesky factor and its stability properties are presented. A downdating algorithm based on the corrected seminormal equation method is described in Section 6. In Section 7 we briefly describe the application of the methods of this paper to the problem of downdating R^{-1}. Finally, we compare the algorithms discussed in this paper and present the results of numerical tests in Section 8.

2 Updating and Downdating the QR Decomposition

Assume that we have computed the QR decomposition of $(X\ s)$ as in (2). Then we have

$$\begin{pmatrix} Q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X & s \\ y^T & \eta \end{pmatrix} = \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \\ y^T & \eta \end{pmatrix}. \tag{4}$$

The row $(y^T\ \eta)$ can now be annihilated and the updated factor is computed by a sequence of plane rotations $U = G_1 \cdots G_{n+1}$, where G_k is a rotation in the plane $(k, p+1)$. We obtain

$$U^T \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \\ y^T & \eta \end{pmatrix} = \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ 0 & 0 \end{pmatrix}. \tag{5}$$

It then follows that

$$\tilde{Q} = \begin{pmatrix} Q & 0 \\ 0 & 1 \end{pmatrix} U \tag{6}$$

is the updated factor $\tilde{Q}$. *Note that Q is not needed for the construction of U and of the updated factor $\tilde{R}$.* This algorithm for updating is backward stable [10]. Indeed, if we construct the upper triangular factor by a sequence of such modifications, the resulting algorithm is equivalent to the sequential row orthogonalization method for computing the QR decomposition.

Assume that we have the QR decomposition

$$(X \quad s) = \begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = Q \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} \tag{7}$$

and want to *remove* the first row $(z^T\ \sigma)$. We now show that this is equivalent to updating the QR factorization when a special column $e_1 = (1, 0, \ldots, 0)^T$ is *added* to the left of $(X\ s)$,

$$(e_1 \quad X \quad s) = \begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix}.$$

Using (7) it follows that

$$Q^T (e_1 \quad X \quad s) = \begin{pmatrix} q_1 & R & u \\ \psi & 0 & \rho \\ q_2 & 0 & 0 \end{pmatrix},$$

where $q^T \equiv (q_1^T \; \psi \; q_2^T)$ is the first row of Q. We can now determine a sequence of plane rotations J_k, $k = p-1, p-2, \ldots, 1$, in the plane $(k, k+1)$ such that

$$U^T \begin{pmatrix} q_1 & R & u \\ \psi & 0 & \rho \\ q_2 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & v^T & \tau \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \\ 0 & 0 & 0 \end{pmatrix}, \qquad U^T = J_1 \cdots J_{p-2} J_{p-1}. \tag{8}$$

Here J_k is chosen to annihilate the $(k+1)$st component in q. Then we have

$$\hat{Q}^T \begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} 1 & v^T & \tau \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \\ 0 & 0 & 0 \end{pmatrix},$$

where $\hat{Q} = QU$. Note that by an extra reflection we could ensure that $\tilde{\rho} \geq 0$, but we do not assume this in the following. Equating the first columns on both sides we see that $\hat{Q}^T e_1 = e_1$, so the first row in $\hat{Q}$ equals e_1. Hence, $\hat{Q}$ must have the form

$$\begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q} \end{pmatrix},$$

and it follows that $(v^T \; \tau) = (z^T \; \sigma)$. Dropping the first row and column gives the downdated QR decomposition of

$$(\tilde{X} \; \tilde{s}) = \tilde{Q} \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ 0 & 0 \end{pmatrix}.$$

Note the important fact that in the downdating case, we need the first row of the square orthogonal factor Q to construct the matrix U. Paige [10] has proved that this downdating algorithm is *mixed stable*, i.e., the computed $\tilde{R}, \tilde{u}$, and $\tilde{\rho}$ are close to the corresponding quantities in the exact factor of

$$(\tilde{X} + E, \tilde{s} + f), \qquad \|E\|_2 = c_1 \mu, \quad \|f\|_2 = c_2 \mu,$$

where c_1 and c_2 are constants depending on the dimension of X, and μ is the roundoff unit.

3 Modifying the Gram-Schmidt Factorization

In many applications, especially if $p \gg n$, *it is too costly to save and modify the full QR decomposition.* When we use the Gram-Schmidt QR factorization, the storage requirement for the Q factor is reduced to pn from p^2, which is for the full QR decomposition. In [5] stable algorithms are derived for modifying the Gram-Schmidt QR factorization of a matrix A when A is changed by a matrix of rank one, or when a row or column is added or deleted. A principal tool of the algorithms is the Gram-Schmidt process *with reorthogonalization.* A slightly simplified algorithm given in Reichel and Gragg [14] relies on the fact that in the full rank case one reorthogonalization is always enough, see Parlett [13].

The algorithm given in Section 2 for adding a row applies with trivial modifications also to the Gram-Schmidt factorization. Assume now that we have the QR factorization

$$(X \quad s) = \begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} q_1^T & \psi \\ Q_1 & y \end{pmatrix} \begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}, \tag{9}$$

and want to delete the first row $(z^T \quad \sigma)$. Note that (9) can be written as

$$\begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} q_1^T & \psi & 1 \\ Q_1 & y & 0 \end{pmatrix} \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix}. \tag{10}$$

Following [5] we first apply the GS process (with reorthogonalization) so that $e_1 = (1, 0 \cdots, 0)^T$ is orthogonalized to $\hat{Q}_1 \equiv \begin{pmatrix} q_1^T & \psi \\ Q_1 & v \end{pmatrix} \in \mathbf{R}^{p \times (n+1)}$. Because of the special form of the appended column, the result has the form

$$\begin{pmatrix} q_1^T & \psi & 1 \\ Q_1 & y & 0 \end{pmatrix} = \begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} \begin{pmatrix} I & 0 & q_1 \\ 0 & 1 & \psi \\ 0 & 0 & \hat{\gamma} \end{pmatrix} \tag{11}$$

for some $h \in \mathbf{R}^{(p-1) \times 1}$, $\hat{\gamma}$, and $\bar{\gamma} \in \mathbf{R}$. Here $q_1^T q_1 + \psi^2 + \hat{\gamma}^2 = \|e_1\|_2^2 = 1$, and equating the first element in the last column in (11) $q_1^T q_1 + \psi^2 + \bar{\gamma}\hat{\gamma} = 1$. Hence we have $\hat{\gamma} = \bar{\gamma}$. If e_1 is linearly dependent on the columns of $\hat{Q}_1$ then we get $\bar{\gamma} = 0$, $h = 0$, and the orthogonalization will fail. In this case we can take a random vector in $\mathbf{R}^{(p-1) \times 1}$ and reorthogonalize to find a unit vector h that is orthogonal to $(Q_1 \; y)$, see [5].

We now write using (10) and (11)

$$\begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix},$$

and determine a sequence of plane rotations J_k, $k = n+1, n, \ldots, 1$, in the plane $(k, n+2)$ such that

$$\begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} U = \begin{pmatrix} 0 & 0 & \tau \\ \tilde{Q}_1 & \tilde{y} & \tilde{h} \end{pmatrix}, \qquad U = J_{n+1} J_n \cdots J_1,$$

where J_k is chosen to annihilate the kth component in $(q_1^T \ \psi \ \bar{\gamma})$. Since orthogonal transformations preserve length we can make $\tau = 1$. The transformed matrix has orthonormal columns and so $\tilde{h} = 0$. It follows that

$$\begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ \tilde{Q}_1 & \tilde{y} & 0 \end{pmatrix} \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ z^T & \sigma \end{pmatrix},$$

where

$$U^T \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ z^T & \sigma \end{pmatrix}$$

with $\tilde{R}$ upper triangular, and the downdated QR decomposition becomes

$$(\tilde{X} \quad \tilde{s}) = (\tilde{Q}_1 \ \tilde{y}) \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \end{pmatrix}. \tag{12}$$

Summarizing the above, we get the following algorithm.

GS Downdating Algorithm:

Given $\hat{Q}_1 = \begin{pmatrix} q_1^T & \psi \\ Q_1 & y \end{pmatrix} \in \mathbf{R}^{p\times(n+1)}$ and $\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix} \in \mathbf{R}^{(n+1)\times(n+1)}$, the following algorithm computes the downdated quantities $\tilde{Q}_1, \tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$:

1. Orthogonalize e_1 to $\hat{Q}_1$ by the GS process with one reorthogonalization step:
   (a) $s := (q_1^T \ \psi)^T$
   (b) $v := e_1 - \hat{Q}_1 s$
   (c) if $\|v\|_2 \geq 1/\sqrt{2}$, $v := v/\|v\|_2$
   (d) else
      i. $s' := \hat{Q}_1^T v$; $v' := v - \hat{Q}_1 s'$
      ii. if $\|v'\|_2 \leq \|v\|_2/\sqrt{2}$, $\bar{\gamma} = 0$;
      determine h with unit length orthogonal to $(Q_1 \ y)$
      iii. else $v := v'/\|v'\|_2$
   (e) $\bar{\gamma} := v(1)$; $h := v(2:p)$

2. Determine an orthogonal matrix U as a product of Givens rotations such that

$$\begin{pmatrix} 0 & 0 & 1 \\ \tilde{Q}_1 & \tilde{y} & \tilde{h} \end{pmatrix} := \begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} U \tag{13}$$

3. Update the R factor by U^T:

$$\begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ z^T & \sigma \end{pmatrix} := U^T \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} \tag{14}$$

4. Compute the new solution $\tilde{w}$ from $\tilde{R}\tilde{w} = \tilde{u}$ and take $\tilde{\rho}$ as the new residual norm. ■

With one reorthogonalization process, the GS downdating algorithm requires about $7pn + 2.5n^2$ flops. This can be reduced to $5pn + 1.5n^2$ flops when fast scaled rotations [2, 9] are used in (13) and (14). Note that the data matrix X is never needed: to delete the first row of X, the R factor and the corresponding row in $\hat{Q}_1$ are needed. Thus, the storage requirement is about $pn + 0.5n^2$ for $\hat{Q}_1$ and R.

4 Downdating the Cholesky Factor

There are several algorithms for *downdating the Cholesky factor* of A^TA, which is mathematically the same as downdating the R factor of the QR decomposition of A. These algorithms have the property that the Q factor is never used. One important algorithm of this type, which uses hyperbolic rotations, has been analyzed by Alexander, Pan, and Plemmons [1]. Another standard algorithm is the LINPACK algorithm, due to Saunders [15].

To derive the LINPACK algorithm for downdating R, note that downdating the ith row of the data matrix $(X \quad s)$ by the method in Section 2 requires the ith row of the orthogonal factor Q. Also note that the transformations $J_{n+2} \ldots, J_{p-1}$ in (8) do not affect $\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}$, but the vector q_2, which is replaced by $\bar{\gamma}e_1$, $\bar{\gamma} = \|q_2\|_2$. Thus, mathematically it suffices to know the first $n+1$ components $(q_1^T \ \psi)$ of the ith row of Q and $\bar{\gamma} \equiv \|q_2\|_2 = \sqrt{1 - (\|q_1\|^2 + \psi^2)}$ to delete the ith row of $(X \quad s)$. From the first row of the QR decomposition (7), we have

$$(z^T \quad \sigma) = (q_1^T \quad \psi \quad q_2^T) \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} = (q_1^T \quad \psi) \begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}.$$

It follows that q_1 and ψ can be computed by solving the triangular system

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix} \begin{pmatrix} q_1 \\ \psi \end{pmatrix} = \begin{pmatrix} z \\ \sigma \end{pmatrix}.$$

Using the relation $u^Tq_1 = u^TR^{-T}z = w^Tz$, we obtain

$$q_1 = R^{-T}z, \quad \psi = (\sigma - z^Tw)/\rho \qquad (\rho \neq 0). \tag{15}$$

Next we should determine a product of plane rotations U such that

$$U^T \begin{pmatrix} q_1 & R & u \\ \psi & 0 & \rho \\ \bar{\gamma} & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \end{pmatrix}. \tag{16}$$

The first rotation in the $(n+1, n+2)$ plane only affects the 2×2 matrix

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \psi & \rho \\ \bar{\gamma} & 0 \end{pmatrix} = \begin{pmatrix} \gamma & \hat{\rho} \\ 0 & \tilde{\rho} \end{pmatrix},$$

and a short calculation gives

$$\gamma^2 = \bar{\gamma}^2 + \psi^2 = 1 - \|q_1\|_2^2, \qquad \hat{\rho} = (\sigma - z^Tw)/\gamma, \qquad \tilde{\rho} = (\rho^2 - \hat{\rho}^2)^{1/2}. \tag{17}$$

(Note that ψ in (15) need not be computed so the assumption that $\rho \neq 0$ is not needed.)
Collecting these results we get:

LINPACK Downdating Algorithm:

Given R, u, ρ, w and $(z^T \ \ \sigma)$, the following algorithm computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$:

1. Compute q_1, γ, and $\hat{\rho}$ from

$$R^Tq_1 = z, \qquad \gamma := (1 - \|q_1\|_2^2)^{1/2}, \qquad \hat{\rho} := (\sigma - z^Tw)/\gamma. \tag{18}$$

2. Determine an orthogonal matrix $\hat{U}$ as a product of Givens rotations such that

$$\begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{R} & \tilde{u} \end{pmatrix} := \hat{U}^T \begin{pmatrix} q_1 & R & u \\ \gamma & 0 & \hat{\rho} \end{pmatrix} \tag{19}$$

3. Compute the new solution $\tilde{w}$ and the residual norm from

$$\tilde{R}\tilde{w} = \tilde{u}, \qquad \tilde{\rho} := (\rho^2 - \hat{\rho}^2)^{1/2}.$$

■

This algorithm requires about $3n^2$ flops for two triangular solves and updating in (19), which can be reduced to $2n^2$ flops when fast rotations are used in (19). Unlike the GS downdating algorithm, the LINPACK algorithm does not require the Q factor. However, the row $(z^T \; \sigma)$ to be deleted should be known to recover the necessary elements in the Q factor. Thus, for solving recursive least squares problems with the sliding window method, all the rows in the data matrix X need to be stored for future downdating. Accordingly, the storage requirement is $pn + 0.5n^2$. This means that *the storage requirement for the LINPACK downdating algorithm can be about as large as that for the GS based downdating algorithm for recursive least squares problems* contrary to a widespread misconception. In special cases such as when X is sparse or its elements can be generated by a formula, the cost of storing X can be much smaller than that of storing Q.

5 Stability Properties of the LINPACK algorithm

It is well known that downdating the Cholesky factor can be very ill-conditioned and can fail. We first note that

$$R^T R - zz^T = R^T(I - q_1 q_1^T)R = \tilde{R}^T \tilde{R}. \tag{20}$$

If we put $I - q_1 q_1^T = LL^T$, then $\tilde{R} = L^T R$, where

$$\kappa(L) = \frac{1}{\gamma}, \qquad \gamma = \sqrt{1 - \|q_1\|^2}. \tag{21}$$

Stewart [16] considered the effect of a perturbation δz in z on the downdated factor $\tilde{R}$. He showed that if

$$\|\delta z\|_2 \leq \mu\sigma_1, \qquad \sigma_1 = \|R\|_2,$$

where μ is the roundoff unit, then, neglecting higher order terms

$$|\delta\tilde{\sigma}_i| \leq 2\mu\sigma_1(\sigma_1/\tilde{\sigma}_i), \tag{22}$$

where $\tilde{\sigma}_i = \sigma_i(\tilde{R})$. This shows that the method can break down if $\tilde{\sigma}_i/\sigma_1 \approx \mu^{1/2}$, i.e. if we downdate to an ill-conditioned matrix $\tilde{R}$. The analysis in [16] also shows that the downdating problem is ill-conditioned if any singular value is reduced significantly (not necessarily becoming small). This happens e.g. if the row to be downdated contains an *outlier*, i.e. an erroneous and large element.

Pan [11] has given a detailed perturbation analysis of the downdating problem for the Cholesky factor, and proved the following result.

Theorem 5.1 *Let $\alpha > 0$ be small enough so that the factorization*

$$\tilde{R}(\epsilon)^T \tilde{R}(\epsilon) = (R + \epsilon E)^T (R + \epsilon E) - (z + \epsilon f)^T (z + \epsilon f) \tag{23}$$

exists for all $\epsilon \in (-\alpha, \alpha)$, where E is an upper triangular matrix. Then we have the bound

$$\begin{aligned} \frac{\|\tilde{R}(\epsilon) - \tilde{R}\|_2}{\|\tilde{R}\|_2} &\leq |\epsilon| \kappa^2(R) C \Big[2n(n^{1/2}C + 1) \frac{\|f\|_2}{\|z\|_2} \\ &+ (2n^{3/2}C + 2n + 1) \frac{\|E\|_2}{\|R\|_2} \Big] + O(\epsilon^2), \end{aligned} \tag{24}$$

where $\kappa(R)$ is the condition number of R, and $C = \|q_1\|^2/\gamma$.

The above perturbation analysis shows that *using R to form the downdating transformations may be a much more ill-conditioned problem than downdating the original matrix X.* This is because the original row in X is not perturbed in the same way as the vector $(q_1^T\ \gamma)$ which is computed by solving a triangular system to determine the downdating transformation in the LINPACK algorithm. Hence, any method that uses R alone to recover the necessary elements of Q cannot be *backward stable in the same sense as the downdating algorithm that uses Q directly.*

We now illustrate the perturbation result, and the possible failure of the LINPACK algorithm using a simple 2 by 1 example:

Example 1. Let $X = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix}$ where $\epsilon = \sqrt{\mu}$. Then the QR decomposition of X correctly rounded to single precision is

$$X = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} = \begin{pmatrix} 1 & -\epsilon \\ \epsilon & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The LINPACK algorithm will compute

$$q_1 = 1/1 = 1, \quad \gamma^2 = 1 - 1 = 0, \quad J_1 = I,$$

and we obtain the downdated R factor $\tilde{R} = 0$ instead of the correct value $\tilde{R} = \epsilon$. It is easily verified that if we downdate using Q we get the correct result.

The information from the second row in X is not present in R, only in Q. Therefore *no method working only from R can hope to do better.*

6 Downdating using Seminormal Equation

We now consider a downdating algorithm in which the method of the iterative refinement is incorporated. The method is based on the **seminormal equations** (SNE)

$$R^T Rw = X^T s,$$

for solving a least squares problem $\min_w \|Xw - s\|_2$. This method is in general no more accurate than the method of normal equations. We instead consider the method of **corrected seminormal equations** (CSNE)

$$\begin{aligned} R^T Rw &= X^T s, \qquad r = s - Xw, \\ R^T R\delta w &= X^T r, \qquad w_c = w + \delta w. \end{aligned} \tag{25}$$

Here a corrected solution w_c is computed by performing one step of iterative refinement on the solution computed from the SNE. Note that we assume that *all computations* are performed in *single precision.*

Assume that the computed $R = \bar{R}$ is such that there exists an exactly orthogonal matrix $\hat{Q}$ such that

$$X + E = \hat{Q}\bar{R}, \qquad \|E\|_F \le c_1\mu\|X\|_F.$$

Then in Björck [4] the following error bound is given for the solution $\bar{w}_c$ computed by CSNE:

Theorem 6.1 *Let $\bar{w}_c$ be the least squares solution computed by the CSNE method, and assume that $\rho = c_1 n^{1/2}\mu\kappa < 1$, where κ is the spectral condition number of X. Then the following error estimate holds up to terms of higher order in $\mu\kappa$:*

$$\begin{aligned} \|w - \bar{w}_c\|_2 &\le \sigma\mu\kappa\Big(c_2\|w\|_2 + n^{1/2}p\frac{\|s\|_2}{\|X\|_2}\Big) \\ &+ n^{1/2}\mu\kappa\Big(n\|w\|_2 + p\kappa\frac{\|r\|_2}{\|X\|_2}\Big) + n^{1/2}\mu\|w\|_2, \end{aligned} \tag{26}$$

where

$$\sigma = c_3\mu\kappa^2, \qquad c_2 = 2n^{1/2}(c_1 + n), \qquad c_3 \le 2n^{1/2}(c_1 + 2n + p/2) \tag{27}$$

Hence, provided that $c_3\mu\kappa^2 < 1$, the forward error will not be worse than for a backward stable method.

In [4] it was shown how the CSNE method can be used to update the R factor when a new column is added. This can be adopted to reconstruct the vector $(q_1^T\gamma)$ as follows.

Theorem 6.2 *Let x be the solution to*

$$\min_x \|e_1 - Xx\|_2,$$

and R the R factor of X. Then the R factor of $(X\ e_1)$ is

$$\begin{pmatrix} R & q_1 \\ 0 & \gamma \end{pmatrix}, \quad q_1 = Rx, \quad |\gamma| = \|e_1 - Xx\|_2. \tag{28}$$

The downdated R factor is then obtained by applying orthogonal transformations to transform the last column into the vector e_1. We now apply this result to downdate the *augmented* R factor by solving the least squares problem

$$\min_{v,\phi} \|e_1 - (X\ s)\begin{pmatrix} v \\ \phi \end{pmatrix}\|_2$$

using the CSNE.

The first step is similar to the LINPACK algorithm. From

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix}\begin{pmatrix} q_1 \\ \psi \end{pmatrix} = \begin{pmatrix} z \\ \sigma \end{pmatrix},$$

assuming $\rho \neq 0$, we get

$$q_1 = R^{-T}z, \quad \psi = (\sigma - z^T w)/\rho.$$

Next we solve

$$\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}\begin{pmatrix} v \\ \phi \end{pmatrix} = \begin{pmatrix} q_1 \\ \psi \end{pmatrix},$$

which gives

$$\phi = \psi/\rho, \quad v = R^{-1}(q_1 - u\phi) = R^{-1}q_1 - \phi w.$$

We only need ϕ and v to compute $|\gamma| = \|t\|_2$, where

$$t = e_1 - (X\ s)\begin{pmatrix} v \\ \phi \end{pmatrix} = e_1 - X(R^{-1}q_1) + \psi(Xw - s)/\rho.$$

Note that

$$\rho^2 = \|Xw - s\|_2^2 = \|\tilde{X}w - \tilde{s}\|_2^2 + (\sigma - z^T w)^2,$$

and therefore we have $|\psi| \leq 1$. It turns out to be important to refine q_1 before ψ is computed– i.e., to perform the algorithm in Gauss-Seidel rather than Jacobi fashion!

CSNE Downdating Algorithm:

Given R, u, ρ, w, the data $(X\ s)$ the following algorithm deletes the first row $(z^T\ \sigma)$ and computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$:

1. Compute q_1, v and t from

$$R^T q_1 = z, \qquad Rv = q_1, \qquad t := e_1 - Xv.$$

2. Update q_1, v and compute γ:

$$\begin{aligned} R^T \delta q_1 &= X^T t, \qquad q_1 := q_1 + \delta q_1, \\ R\delta v &= \delta q_1, \qquad t := t - X\delta v, \qquad \gamma := \|t\|_2. \end{aligned}$$

3. Set $\psi := \tilde{\rho} := \hat{\rho} := 0$
   If $\rho \neq 0$,
   (a) compute the normalized residual: $\hat{r} := (s - Xw)/\rho$
   (b) modify t: $\psi := e_1^T \hat{r}, \quad t := t - \psi\hat{r}$
   (c) update ψ and t: $\delta\psi := \hat{r}^T t, \quad \psi := \psi + \delta\psi, \quad t := t - \delta\psi\hat{r}$
   (d) compute $\hat{\rho} = \psi\rho/\gamma, \qquad \tilde{\rho} = \rho\|t\|_2/\gamma$
4. Determine an orthogonal matrix U^T as a product of Givens rotations such that

$$\begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{R} & \tilde{u} \end{pmatrix} := U^T \begin{pmatrix} q_1 & R & u \\ \gamma & 0 & \hat{\rho} \end{pmatrix}$$

5. Compute the new solution $\tilde{w}$ from

$$\tilde{R}\tilde{w} = \tilde{u}.$$

■

Example 2. Let X be as in Example 1. In the method of seminormal equations we compute

$$v = q_1 = 1, \quad \gamma = \| \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} 1 \|_2 = \epsilon.$$

There is no need for the refinement steps 2 and 3 here, and we get

$$U^T \begin{pmatrix} 1 & 1 \\ \epsilon & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & \epsilon \end{pmatrix}, \quad \tilde{R} = \epsilon,$$

which is the correct result.

Algorithm CSNE requires three more triangular solves than the LINPACK method, if the iterative refinement is carried out. Also, four extra matrix times vector multiplications with X and X^T are required. Hence, the added computational complexity is quite high, $\approx 4pn + 1.5n^2$ flops. However, sometimes X may have a special structure and the matrix-vector multiplications can be performed by fast algorithms. This is the case, for example, when X is a sparse or Toeplitz matrix. The storage requirement is the same as that for the LINPACK algorithm.

When the CSNE algorithm is too expensive to use in every step, we suggest a **hybrid algorithm** [6], where the CSNE algorithm is used if the downdating is ill-conditioned and the LINPACK algorithm is used otherwise. As a measure of conditioning of the downdating problem for R augmented by the right hand side u and residual ρ, we have used the quantity (cf. (16) and [16])

$$\bar{\gamma}^2 = 1 - \|q_1\|_2^2 - \psi^2. \tag{29}$$

If $\bar{\gamma}^2$ is less than a user-specified constant *tol* then the downdating is performed with CSNE. Our numerical experiments indicate the hybrid method with *tol* in the range $[0.25, 0.5]$ produces much more accurate results than the LINPACK algorithm.

7 Downdating the Inverse of R

The problem of updating and downdating the inverse of the matrix R in the QR decomposition has been studied in [12] for methods based on orthogonal as well as hyperbolic rotations. One motivation for working with R^{-1} instead of R itself is that such an algorithm can be parallelized more easily. Furthermore, there are applications (see [12]) where the elements of the inverse are needed.

The downdating methods described in this paper can be modified to downdate the inverse of R. We first describe how the inverse and the solution vector w can be downdated recursively.

Consider the transformation (19)

$$\hat{U}^T \begin{pmatrix} q_1 & R \\ \gamma & 0 \end{pmatrix} = \begin{pmatrix} 1 & z^T \\ 0 & \tilde{R} \end{pmatrix}, \qquad \hat{U}^T \begin{pmatrix} u \\ \hat{\rho} \end{pmatrix} = \begin{pmatrix} \sigma \\ \tilde{u} \end{pmatrix}, \tag{30}$$

where $\hat{\rho}$ is computed from (17). By simply inverting the matrices in the first equation in (30) we get the following formula for downdating the inverse:

$$\begin{pmatrix} 0 & \frac{1}{\gamma} \\ R^{-1} & -\frac{1}{\gamma}R^{-1}q_1 \end{pmatrix} \hat{U} = \begin{pmatrix} 1 & -z^T\tilde{R}^{-1} \\ 0 & \tilde{R}^{-1} \end{pmatrix}. \tag{31}$$

Combining (31) and the second equation in (30), we see that

$$\begin{pmatrix} \sigma - z^T\tilde{R}^{-1}\tilde{u} \\ \tilde{R}^{-1}\tilde{u} \end{pmatrix} = \begin{pmatrix} 1 & -z^T\tilde{R}^{-1} \\ 0 & \tilde{R}^{-1} \end{pmatrix}\begin{pmatrix} \sigma \\ \tilde{u} \end{pmatrix} = \begin{pmatrix} 1 & -z^T\tilde{R}^{-1} \\ 0 & \tilde{R}^{-1} \end{pmatrix}\hat{U}^T\begin{pmatrix} u \\ \hat{\rho} \end{pmatrix}$$
$$= \begin{pmatrix} 0 & \frac{1}{\gamma} \\ R^{-1} & -\frac{1}{\gamma}R^{-1}q_1 \end{pmatrix}\begin{pmatrix} u \\ \hat{\rho} \end{pmatrix} = \begin{pmatrix} \frac{\hat{\rho}}{\gamma} \\ R^{-1}u - \frac{\hat{\rho}}{\gamma}R^{-1}q_1 \end{pmatrix},$$

and we get the following downdating formula for the solution $w = R^{-1}u$:

$$\tilde{w} = \tilde{R}^{-1}\tilde{u} = w - \frac{\hat{\rho}}{\gamma}R^{-1}q_1. \tag{32}$$

Since we assume that $rank(\tilde{X}) = n$, we have $\gamma \neq 0$. The LINPACK, the CSNE, and the hybrid algorithms only differ in the way the vector $(q_1^T\ \gamma)$ is computed. For determining this vector, triangular systems with the matrices R and R^T need to be solved. This can be replaced by the corresponding multiplication by the inverse. Therefore these methods can also be used in connection with inverse downdating.

8 Numerical Experiments

In a sliding window method, a least squares solution is computed based on the p latest rows of an observation matrix A, where p is the number of rows in the window matrix [1]. In step k, the new row of observation, $A(k,:)$, is updated into the QR decomposition, and the existing row $A(k-p,:)$ of the data matrix is downdated from the decomposition. If an outlier occurs at step j, then in exact arithmetic its influence will not be seen after step $j+p$. However, the downdating problem is very ill-conditioned in the step when the outlier is to be removed, and any algorithm that does not explicitly use Q or the original data X, e.g., the LINPACK algorithm or a hyperbolic rotation-based algorithm [1], is likely to introduce a large error into the decomposition.

In the sliding window context, the storage requirements of the four algorithms presented in the previous sections are the same in the general case, $pn + 0.5n^2$. The GS algorithm requires the orthogonal factor but not the data matrix in storage, while the other algorithms require the data matrix but not the orthogonal factor. However, when X has a special structure, the storage requirement for X can be much smaller than that for Q.

The computational complexities of the four algorithms for each downdating are compared in Table 1. We give the operation counts (1 flop = 1 addition and 1 multiplication) for standard and fast [2, 9] Givens rotations. For the hybrid algorithm, the computational complexity is either the same as that for the LINPACK algorithm or the CSNE algorithm. The LINPACK algorithm has a clear advantage in computational complexity. In a sliding window context, the GS algorithm is more expensive also in the updating stage, since the Q factor needs to be modified. For the CSNE algorithm the updating stage is the same as for the LINPACK method. Hence for a complete up/downdating step using standard Givens rotations the GS algorithm requires $11pn + 4.5n^2$ but the CSNE algorithm only $4pn + 6.5n^2$ flops. Finally, note that if, e.g., X is Toeplitz the term $4pn$ in the operation count for the CSNE algorithm can be reduced by using a fast algorithm for the matrix-vector products with X and X^T.

Table 1: **Computational Complexity of Downdating algorithms (flops)**

Algorithm	Standard Givens Rotations	Fast Givens Rotations
GS	$7pn + 2.5n^2$	$5pn + 1.5n^2$
LINPACK	$3n^2$	$2n^2$
CSNE	$4pn + 4.5n^2$	$4pn + 3.5n^2$
Hybrid	$3n^2$ or $4pn + 4.5n^2$	$2n^2$ or $4pn + 3.5n^2$

One application area where downdating is used in connection with the sliding window method is adaptive filtering. In [3] it is noted that in certain situations this method does not perform well as rounding errors accumulate and eventually destroy the solution. Our numerical tests indicate that these difficulties are not to be ascribed to the sliding window method itself but rather to the downdating algorithm used.

Numerical tests using the sliding window method have been performed in Pro-Matlab with IEEE double precision floating point arithmetic to compare the accuracy of the four downdating algorithms. The solution obtained from the QR decomposition of the window matrix was used as a reference and a window of size 8 was used throughout. In each figure, we present the relative error in Euclidean norm in the downdated solution vector by the LINPACK, CSNE, hybrid, and GS algorithms. The spectral condition number κ_X of the window matrix to be downdated and $1/\bar{\gamma}^2$, which is a measure of the conditioning of the downdating problem (29), are also shown. A "+" sign in the plot shows where iterative refinement is made in the hybrid method. We have used the following criterion: if $\bar{\gamma}^2 < 0.25$, then downdating is performed with the CSNE method.

The following two test problems are similar to those in [6]. They were also used in the context of adaptive condition number estimation in [7].

Test I. A random matrix $A \in \mathbf{R}^{50\times 5}$ was constructed with elements taken from a uniform distribution in $(0,1)$. An outlier equal to $r \cdot 10^3$, where r is random number from the same distribution, was added in position (18,3). The right hand side vector b was taken to be $b = Ax_0 + b_r$, where b_r has random elements uniformly distributed in $(0, 10^{-6})$, and x_0 is 5×1 vector with ones as its components.

The results are shown in Figure 1. It is seen that the relative error in the solution using the LINPACK algorithm is considerably magnified in the ill-conditioned downdating step and that it remains on that high level even if the subsequent downdating steps are well-conditioned. The other algorithms are much less affected by the ill-conditioned downdating and the errors remain on a low level throughout.

Test II. A 50×5 matrix was constructed by taking a 25×5 Hilbert matrix as the first 25 rows, and the same rows in reversed order as the 25 last rows. Then a perturbation from a uniform distribution in $(0, \delta)$ was added to each matrix element. Two different cases were studied, with $\delta = 10^{-5}$ and 10^{-9}, respectively. The right hand side was constructed as in Test I, but here with a random perturbation in $(0,1)$.

In Figure 2 we show the results obtained with $\delta = 10^{-5}$. Throughout this test the downdating problem is rather ill-conditioned, so iterative refinement is performed in most steps in the hybrid algorithm. It is remarkable that the LINPACK algorithm performs so much worse than the others. This is probably due to the fact that the window matrix is very ill-conditioned, which leads to large errors in the computed approximations of q_1. In the CSNE method this vector is refined and much better accuracy is attained.

In Figure 3 we show the results obtained with $\delta = 10^{-9}$. Here the window matrix is even more ill-conditioned, and after some steps the LINPACK algorithm breaks down because a computed q_1 has norm larger than 1.

9 Concluding Remarks

We have studied two standard methods for downdating least squares solutions, the LINPACK and the Gram Schmidt (GS) algorithms, and two new methods, the CSNE algorithm and a hybrid algorithm CSNE/LINPACK. In terms of storage requirements

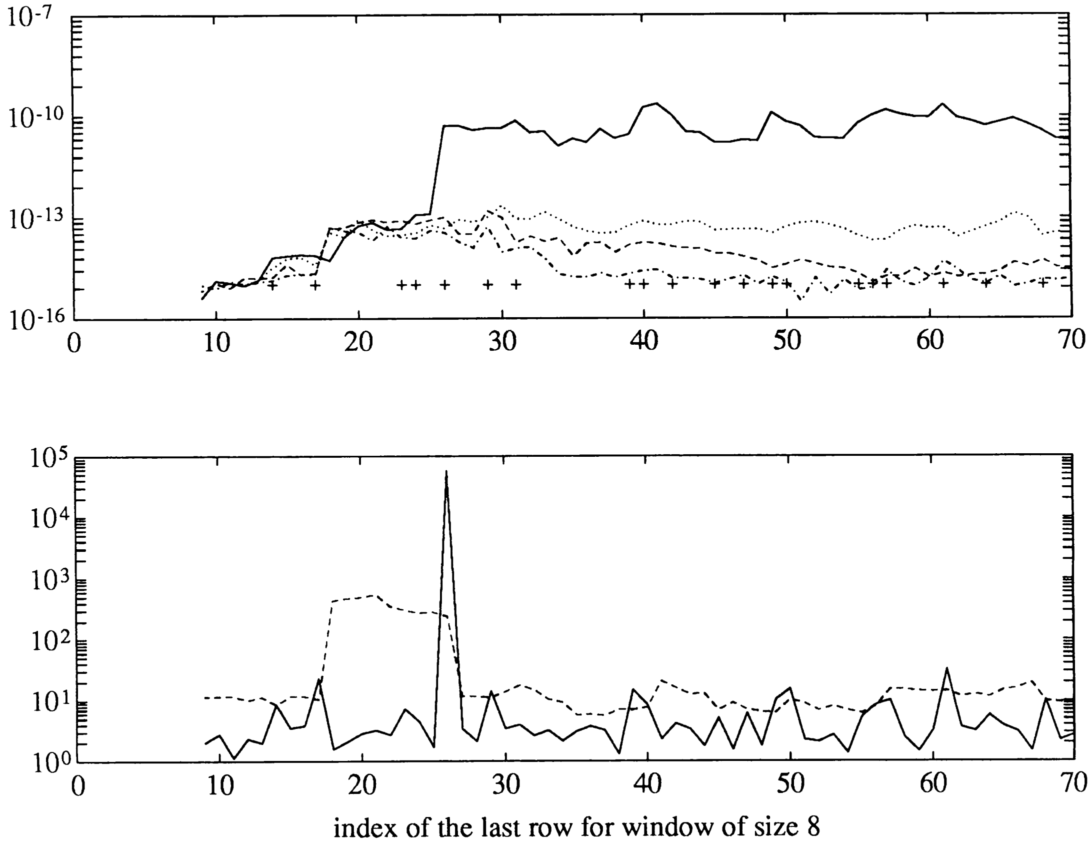

Figure 1: Test I. The upper graph shows the relative error in Euclidean norm in the downdated solution vector by the LINPACK (solid line), CSNE (dashed), hybrid (dotted), and GS (dot-dashed) algorithms. A "+" sign in the plot shows where iterative refinement is made in the hybrid method. The lower graph shows the condition number κ_X of the window matrix to be downdated (dotted) and $1/\bar{\gamma}^2$ (solid line).

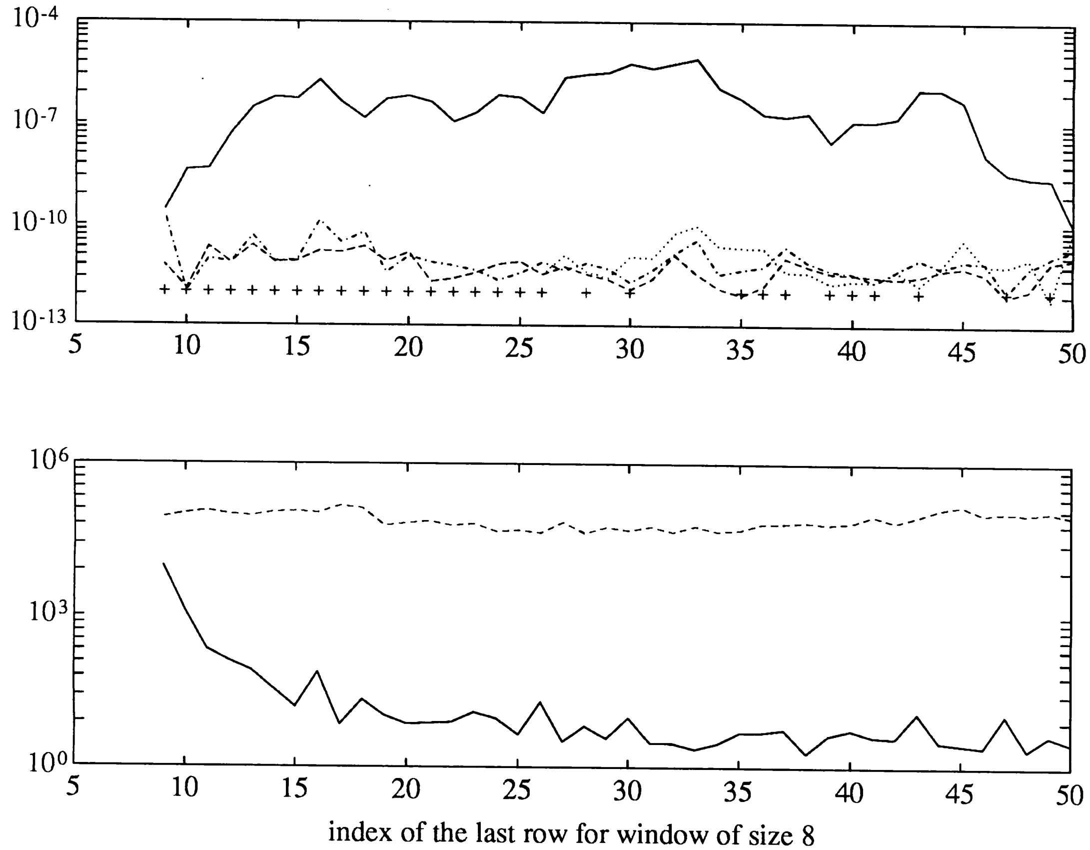

Figure 2: Test IIa. Modified Hilbert matrix with perturbations from a uniform distribution in $(0, 10^{-5})$.

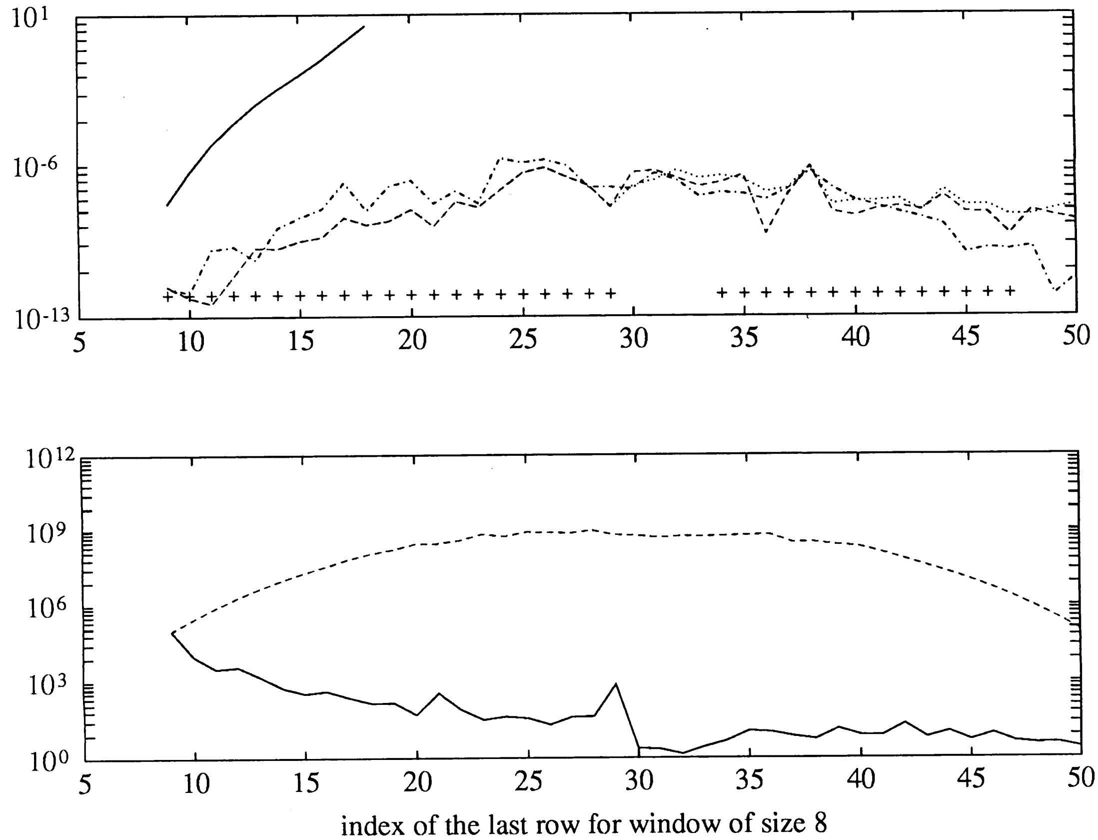

Figure 3: Test IIb. Modified Hilbert matrix with perturbations from a uniform distribution in $(0, 10^{-9})$.

the four algorithms are all the same in the general case. However, when X has a special structure, the storage requirement for the LINPACK and CSNE algorithms can be much smaller than that for the GS algorithm.

The algorithms differ considerably in efficiency and accuracy. The LINPACK algorithm is the fastest but the analysis and the tests show that it can be much less accurate or even fail. It is clear that the CSNE algorithm is more accurate but considerably slower than the LINPACK algorithm. However, if X has some special structure the difference in efficiency may be less pronounced. The hybrid algorithm has almost as good accuracy as the CSNE algorithm and it can be much more efficient. The GS algorithm is comparable in accuracy to the CSNE algorithm, but it is also the slowest of the four algorithms.

The reason why the LINPACK algorithm is inferior in terms of accuracy to the three others is that it uses less information, i.e. only the R factor. The others use both R and either the Q factor (GS) or the original data matrix X (CSNE and Hybrid). Note that other methods which only use R (e.g., methods based on hyperbolic transformations) will show a similar loss of accuracy as the LINPACK algorithm.

Our results indicate that in cases where, e.g., outliers occur, the CSNE and the GS algorithms are the safest choices. If accuracy and efficiency are both important then the hybrid method may be a better alternative than the LINPACK algorithm. Further study is needed in deciding how to choose the tolerance used in the hybrid algorithm to switch between the CSNE and LINPACK algorithms.

10 Acknowledgements

A large part of this work was performed when the first and the third author visited the Institute of Mathematics and its Applications at University of Minnesota, Minneapolis, during the "Applied Linear Algebra" year. We are grateful for the pleasant working conditions and for the stimulating environment.

We are also grateful to R.J. Plemmons for suggesting the application of the algorithms to the problem of downdating the inverse of R.

References

[1] S. T. Alexander, C.-T. Pan, and R. J. Plemmons. Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing. *Lin. Alg. Appl.*,

98:3–40, 1988.

[2] A.A. Anda and H. Park. Fast plane rotation algorithms with dynamic scaling. To appear in *SIAM J. Matrix Anal. Appl.* Also *tech. report*, 90-41, AHPCRC, Univ. of Minnesota, Dec., 1990.

[3] M. G. Bellanger. The Family of Fast Least Squares Algorithms for Adaptive Filtering. In Mathematics in Signal Processing, ed. J. G. McWhirter, Clarendon Press. Oxford, 1990, 415–434.

[4] Å. Björck. Stability analysis of the method of semi-normal equations for least squares problems. *Lin. Alg. Appl.*, 88/89:31–48, 1987.

[5] J. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Math. Comp.*, 30:772–95, 1976.

[6] L. Eldén. Downdating QR decompositions, *Proceedings of the Conference on Mathematics in Signal Processing*, 1988. The Institute of Mathematics and Its Application, Essex, 1990.

[7] W.R. Ferng, G.H. Golub, and R.J. Plemmons. Adaptive Lanczos methods for recursive condition estimation. *J. Numer. Alg.* 1:1-20, 1991.

[8] G. H. Golub and G. P. Styan. Numerical computations for univariate linear models. *J. Stat. Comp. Simul.*, 2:253-274, 1973.

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

[10] C. C. Paige. Error analysis of some techniques for updating orthogonal decompositions. *Math. Comp.*, 34:465–471, 1980.

[11] C. T. Pan. A perturbation analysis of the problem of downdating a Cholesky factorization. Tech. Report, Dept. of Mathematics, Northern Illinois University, October 1990 (revised).

[12] C. T. Pan and R.J. Plemmons. Least Squares Modifications with Inverse Factorizations: Parallel Implications. *Comp. Appl. Math.*, 27:109-127, 1989.

[13] B.N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall Englewood Cliffs, N.J., 1980.

[14] L. Reichel and W. B. Gragg. FORTRAN subroutines for updating the QR decomposition. *ACM Trans. Math. Software*, 16:369–377, 1990.

[15] M. A. Saunders. *Large-Scale Linear Programming Using The Cholesky Factorization.* Technical Report CS252, Computer Science Department, Stanford University, 1972.

[16] 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.

Recent IMA Preprints

#	Author/s	Title

866 **Gui–Qiang Chen and Tai–Ping Liu**, Zero relaxation and dissipation limits for hyperbolic conservation laws
867 **Gui–Qiang Chen and Jian–Guo Liu**, Convergence of second-order schemes for isentropic gas dynamics
868 **Aleksander M. Simon and Zbigniew J. Grzywna**, On the Larché–Cahn theory for stress-induced diffusion
869 **Jerzy Łuczka, Adam Gadomski and Zbigniew J. Grzywna**, Growth driven by diffusion
870 **Mitchell Luskin and Tsorng-Whay Pan**, Nonplanar shear flows for nonaligning nematic liquid crystals
871 **Mahmoud Affouf**, Unique global solutions of initial-boundary value problems for thermodynamic phase transitions
872 **Richard A. Brualdi and Keith L. Chavey**, Rectangular L-matrices
873 **Xinfu Chen, Avner Friedman and Bei Hu**, The thermistor problem with zero–one conductivity II
874 **Raoul LePage**, Controlling a diffusion toward a large goal and the Kelly principle
875 **Raoul LePage**, Controlling for optimum growth with time dependent returns
876 **Marc Hallin and Madan L. Puri**, Rank tests for time series analysis a survey
877 **V.A. Solonnikov**, Solvability of an evolution problem of thermocapillary convection in an infinite time interval
878 **Horia I. Ene and Bogdan Vernescu**, Viscosity dependent behaviour of viscoelastic porous media
879 **Kaushik Bhattacharya**, Self-accommodation in martensite
880 **D. Lewis, T. Ratiu, J.C. Simo and J.E. Marsden**, The heavy top: a geometric treatment
881 **Leonid V. Kalachev**, Some applications of asymptotic methods in semiconductor device modeling
882 **David C. Dobson**, Phase reconstruction via nonlinear least-squares
883 **Patricio Aviles and Yoshikazu Giga**, Minimal currents, geodesics and relaxation of variational integrals on mappings of bounded variation
884 **Patricio Aviles and Yoshikazu Giga**, Partial regularity of least gradient mappings
885 **Charles R. Johnson and Michael Lundquist**, Operator matrices with chordal inverse patterns
886 **B.J. Bayly**, Infinitely conducting dynamos and other horrible eigenproblems
887 **Charles M. Elliott and Stefan Luckhaus**, 'A generalised diffusion equation for phase separation of a multi-component mixture with interfacial free energy'
888 **Christian Schmeiser and Andreas Unterreiter**, The derivation of analytic device models by asymptotic methods
889 **LeRoy B. Beasley and Norman J. Pullman**, Linear operators that strongly preserve the index of imprimitivity
890 **Jerry Donato**, The Boltzmann equation with lie and cartan
891 **Thomas R. Hoffend Jr., Peter Smereka and Roger J. Anderson**, A method for resolving the laser induced local heating of moving magneto-optical recording media
892 **E.G. Kalnins, Willard Miller, Jr. and Sanchita Mukherjee**, Models of q-algebra representations: the group of plane motions
893 **T.R. Hoffend Jr. and R.K. Kaul**, Relativistic theory of superpotentials for a nonhomogeneous, spatially isotropic medium
894 **Reinhold von Schwerin**, Two metal deposition on a microdisk electrode
895 **Vladimir I. Oliker and Nina N. Uraltseva**, Evolution of nonparametric surfaces with speed depending on curvature, III. Some remarks on mean curvature and anisotropic flows
896 **Wayne Barrett, Charles R. Johnson, Raphael Loewy and Tamir Shalom**, Rank incrementation via diagonal perturbations
898 **Mingxiang Chen, Xu-Yan Chen and Jack K. Hale**, Structural stability for time-periodic one-dimensional parabolic equations
899 **Hong-Ming Yin**, Global solutions of Maxwell's equations in an electromagnetic field with the temperature-dependent electrical conductivity
900 **Robert Grone, Russell Merris and William Watkins**, Laplacian unimodular equivalence of graphs
901 **Miroslav Fiedler**, Structure-ranks of matrices
902 **Miroslav Fiedler**, An estimate for the nonstochastic eigenvalues of doubly stochastic matrices
903 **Miroslav Fiedler**, Remarks on eigenvalues of Hankel matrices
904 **Charles R. Johnson, D.D. Olesky, Michael Tsatsomeros and P. van den Driessche**, Spectra with positive elementary symmetric functions
905 **Pierre-Alain Gremaud**, Thermal contraction as a free boundary problem
906 **K.L. Cooke, Janos Turi and Gregg Turner**, Stabilization of hybrid systems in the presence of feedback delays
907 **Robert P. Gilbert and Yongzhi Xu**, A numerical transmutation approach for underwater sound propagation
908 **LeRoy B. Beasley, Richard A. Brualdi and Bryan L. Shader**, Combinatorial orthogonality
909 **Richard A. Brualdi and Bryan L. Shader**, Strong hall matrices
910 **Håkan Wennerström and David M. Anderson**, Difference versus Gaussian curvature energies; monolayer versus bilayer curvature energies applications to vesicle stability
911 **Shmuel Friedland**, Eigenvalues of almost skew symmetric matrices and tournament matrices
912 **Avner Friedman, Bei Hu and J.L. Velazquez**, A Free Boundary Problem Modeling Loop Dislocations in Crystals
913 **Ezio Venturino**, The Influence of Diseases on Lotka-Volterra Systems
914 **Steve Kirkland and Bryan L. Shader**, On Multipartite Tournament Matrices with Constant Team Size

915 **Richard A. Brualdi and Jennifer J.Q. Massey**, More on Structure-Ranks of Matrices
916 **Douglas B. Meade**, Qualitative Analysis of an Epidemic Model with Directed Dispersion
917 **Kazuo Murota**, Mixed Matrices Irreducibility and Decomposition
918 **Richard A. Brualdi and Jennifer J.Q. Massey**, Some Applications of Elementary Linear Algebra in Combinations
919 **Carl D. Meyer**, Sensitivity of Markov Chains
920 **Hong-Ming Yin**, Weak and Classical Solutions of Some Nonlinear Volterra Integrodifferential Equations
921 **B. Leinkuhler and A. Ruehli**, Exploiting Symmetry and Regularity in Waveform Relaxation Convergence Estimation
922 **Xinfu Chen and Charles M. Elliott**, Asymptotics for a Parabolic Double Obstacle Problem
923 **Yongzhi Xu and Yi Yan**, An Approximate Boundary Integral Method for Acoustic Scattering in Shallow Oceans
924 **Yongzhi Xu and Yi Yan**, Source Localization Processing in Perturbed Waveguides
925 **Kenneth L. Cooke and Janos Turi**, Stability, Instability in Delay Equations Modeling Human Respiration
926 **F. Bethuel, H. Brezis, B.D. Coleman and F. Hélein**, Bifurcation Analysis of Minimizing Harmonic Maps Describing the Equilibrium of Nematic Phases Between Cylinders
927 **Frank W. Elliott, Jr.**, Signed Random Measures: Stochastic Order and Kolmogorov Consistency Conditions
928 **D.A. Gregory, S.J. Kirkland and B.L. Shader**, Pick's Inequality and Tournaments
929 **J.W. Demmel, N.J. Higham and R.S. Schreiber**, Block *LU* Factorization
930 **Victor A. Galaktionov and Juan L. Vazquez**, Regional Blow-Up in a Semilinear Heat Equation with Convergence to a Hamilton-Jacobi Equation
931 **Bryan L. Shader**, Convertible, Nearly Decomposable and Nearly Reducible Matrices
932 **Dianne P. O'Leary**, Iterative Methods for Finding the Stationary Vector for Markov Chains
933 **Nicholas J. Higham**, Perturbation theory and backward error for $AX - XB = C$
934 **Z. Strakos and A. Greenbaum**, Open questions in the convergence analysis of the lanczos process for the real symmetric eigenvalue problem
935 **Zhaojun Bai**, Error analysis of the lanczos algorithm for the nonsymmetric eigenvalue problem
936 **Pierre-Alain Gremaud**, On an elliptic-parabolic problem related to phase transitions in shape memory alloys
937 **Bojan Mohar and Neil Robertson**, Disjoint essential circuits in toroidal maps
938 **Bojan Mohar**, Convex representations of maps on the torous and other flat surfaces
939 **Bojan Mohar and Svatopluk Poljak** Eigenvalues in combinatorial optimization
940 **Richard A. Brualdi, Keith L. Chavey and Bryan L. Shader**, Conditional sign-solvability
941 **Roger Fosdick and Ying Zhang**, The torsion problem for a nonconvex stored energy function
942 **René Ferland and Gaston Giroux**, An unbounded mean-field intensity model: Propagation of the convergence of the empirical laws and compactness of the fluctuations
943 **Wei-Ming Ni and Izumi Takagi**, Spike-layers in semilinear elliptic singular Perturbation Problems
944 **Henk A. Van der Vorst and Gerard G.L. Sleijpen**, The effect of incomplete decomposition preconditioning on the convergence of conjugate gradients
945 **S.P. Hastings and L.A. Peletier**, On the decay of turbulent bursts
946 **Apostolos Hadjidimos and Robert J. Plemmons**, Analysis of p-cyclic iterations for Markov chains
947 **ÅBjörck, H. Park and L. Eldén**, Accurate downdating of least squares solutions
948 **E.G. Kalnins, William Miller, Jr. and G.C. Williams**, Recent advances in the use of separation of variables methods in general relativity
949 **G.W. Stewart**, On the perturbation of LU, Cholesky and QR factorizations
950 **G.W. Stewart**, Gaussian elimination, perturbation theory and Markov chains
951 **G.W. Stewart**, On a new way of solving the linear equations that arise in the method of least squares
952 **G.W. Stewart**, On the early history of the singular value decomposition
953 **G.W. STewart**, On the perturbation of Markov chains with nearly transient states
954 **Umberto Mosco**, Composite media and asymptotic dirichlet forms
955 **Walter F. Mascarenhas**, The structure of the eigenvectors of sparse matrices
956 **Walter F. Mascarenhas**, A note on Jacobi being more accurate than QR
957 **Raymond H. Chan, James G. Nagy and Robert J. Plemmons**, FFT-based preconditioners for Toeplitz-Block least squares problems
958 **Zhaojun Bai**, The CSD, GSVD, their applications and computations
959 **D.A. Gregory, S.J. Kirkland and N.J. Pullman**, A bound on the exponent of a primitive matrix using Boolean rank
960 **Richard A. Brualdi, Shmuel Friedland and Alex Pothen**, Sparse bases, elementary vectors and nonzero minors of compound matrices
961 **J.W. Demmel**, Open problems in numerical linear algebra
962 **James W. Demmel and William Gragg**, On computing accurate singular values and eigenvalues of acyclic matrices
963 **James W. Demmel**, The inherent inaccuracy of implicit tridiagonal QR
964 **J.J.L. Velázquez**, Estimates on the $(N-1)$-dimensional Hausdorff measure of the blow-up set for a semilinear heat equation