ACCURATE DOWNDATING OF A QR DECOMPOSITION *

K. YOO[1] and H. PARK[2] †

[1] *Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, U.S.A. email: yoo@cs.umn.edu*

[2] *Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, U.S.A. email: hpark@cs.umn.edu*

Abstract.

A new algorithm for downdating a QR decomposition is presented. We show that, when the columns in the Q factor from the Modified Gram-Schmidt QR decomposition of a matrix X are exactly orthonormal, the Gram-Schmidt downdating algorithm for the QR decomposition of X is equivalent to downdating the full Householder QR decomposition of the matrix X augmented by an $n \times n$ zero matrix on top. Using this relation, we derive an algorithm that improves the Gram-Schmidt downdating algorithm when the columns in the Q factor are not orthonormal. Numerical test results show that the new algorithm produces far more accurate results than the Gram-Schmidt downdating algorithm for certain ill-conditioned problems.

AMS subject classification: 65C20, 65F05, 65M12, 15A04

Key words: QR decomposition downdating, Gram-Schmidt downdating algorithm, Householder transformation, least squares problems, modified Gram-Schmidt algorithm, reorthogonalization.

1 Introduction

Let $X \in \mathbf{R}^{m\times n}$, $m > n$, have a QR decomposition

$$X = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{1.1}$$

where $Q \in \mathbf{R}^{m\times m}$ is orthogonal and $R \in \mathbf{R}^{n\times n}$ is upper triangular. Frequently, one knows the factorization (1.1) and wishes to find the new QR decomposition

*Received November 1994. Revised October 1995.

†This work was partially supported in part by the National Science Foundation grants CCR-9209726 and CCR-9509085.

after a new observation y^T is added (updating):

$$\check{X} = \begin{pmatrix} X \\ y^T \end{pmatrix} = \check{Q} \begin{pmatrix} \check{R} \\ 0 \end{pmatrix}$$

or after an old observation z^T is removed (downdating):

$$X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix}, \qquad \tilde{X} = \tilde{Q} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}.$$

In numerous applications, updating or downdating of a QR decomposition is needed [11]. There is essentially one algorithm for updating, which is like an intermediate step of QR decomposition with Givens rotations, and it is backward stable [8, 10]. The transformation for updating is determined based on the upper triangular factor R and the new vector y. The main difficulty with downdating lies in the fact that the transformation needed for downdating the row z^T is determined based on the first row of the full $(m \times m)$ orthogonal factor Q. In general, when the QR decomposition is downdated by deleting the kth row of X, we need to know the kth row of $Q \in \mathbf{R}^{m \times m}$. With the kth row of Q known, downdating is not more difficult than updating and there is a $O(m^2)$ mixed stable algorithm in this case, which is summarized in the next section [8, 10]. However, storing and modifying the $m \times m$ orthogonal factor Q can be costly. In many downdating algorithms presented in literature, such as LINPACK algorithm and the algorithms based on the stabilized hyperbolic (or hyperbolic) transformation or corrected-seminormal equations (CSNE) [1, 6, 7, 9, 14], it is assumed that the orthogonal factor $Q \in \mathbf{R}^{m \times m}$ is not available. Accordingly, these algorithms *recover* the necessary information on Q to compute the downdating transformation. In fact, these algorithms compute the rank one downdating of the Cholesky factor of $X^T X$ since

$$\tilde{R}^T \tilde{R} = R^T R - z z^T,$$

which is mathematically the same as downdating the R factor in the QR decomposition of X *when* Q *is unknown.* For the comparison of these algorithms, see [6].

Any downdating algorithm that uses only the upper triangular factor R and does not use the original data X or the orthogonal factor Q relies on less information and can suffer from loss of accuracy. The Gram-Schmidt downdating algorithm and the algorithm based on the CSNE method are the two most reliable $O(mn)$ algorithms [6]. For notational purpose, we will call the decomposition

$$X = Q_1 R \tag{1.2}$$

where $Q_1 \in \mathbf{R}^{m \times n}$ consists of the first n columns of Q in (1.1) and $R \in \mathbf{R}^{n \times n}$ is the same as R in (1.1), as a *skinny* QR decomposition in contrast to the *full* QR decomposition.

In this paper, we present a new $O(mn)$ algorithm for downdating the *skinny* QR decomposition, which further improves the accuracy of the Gram-Schmidt downdating algorithm. The key idea in our new algorithm is based on the result of Björk and Paige [5] that shows the relation between the modified Gram-Schmidt QR decomposition of X and the full Householder QR decomposition of the same matrix augmented by an $n \times n$ zero matrix on top. We show that, when the columns in the factor Q_1 from the Modified Gram-Schmidt algorithm for X are exactly orthonormal, the Gram-Schmidt downdating algorithm for the skinny QR decomposition of X is equivalent to downdating the full Householder QR decomposition of the matrix X augmented by an $n \times n$ zero matrix on top. Using this relation, we derive an algorithm that improves the Gram-Schmidt downdating algorithm when the columns in the factor Q_1 are not orthonormal. Numerical test results show that the new algorithm produces far more accurate results than the Gram-Schmidt downdating algorithm for certain ill-conditioned problems.

2 Downdating a Full QR Decomposition

Assume that we have the QR decomposition (1.1) for $X \in \mathbf{R}^{m\times n}$ and want to obtain the QR decomposition for $\tilde{X}$ after deleting the first row z^T. Let a plane rotation $J_{k,k+1}$ in the plane $(k, k+1)$ be a rank two modification to the identity matrix such that

$$J_{k,k+1}(k,k) = J_{k,k+1}(k+1,k+1) = c, \quad J_{k,k+1}(k,k+1) = -J_{k,k+1}(k+1,k) = s$$

with $c = \cos(\theta)$ and $s = \sin(\theta)$ for some θ. Suppose

$$Q = \begin{pmatrix} p^T \\ \hat{Q} \end{pmatrix},$$

where $p^T \in \mathbf{R}^{1\times m}$ and $\hat{Q} \in \mathbf{R}^{(m-1)\times m}$ and we determine plane rotations $J_{k,k+1}$, $k = m-1, m-2, \ldots, 1$, so that

$$Up \equiv J_{1,2} \cdots J_{m-1,m} p = e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbf{R}^{m\times 1}. \tag{2.1}$$

Then

$$QU^T = \begin{pmatrix} p^T \\ \hat{Q} \end{pmatrix} J^T_{m-1,m} \cdots J^T_{1,2} = \begin{pmatrix} 1 & 0 \\ w & \tilde{Q} \end{pmatrix},$$

where $w \in \mathbf{R}^{(m-1)\times 1}$ and $\tilde{Q} \in \mathbf{R}^{(m-1)\times(m-1)}$. Since QU^T is an orthogonal matrix, $w = 0$ and $\tilde{Q}$ is orthogonal. From

$$X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = \begin{pmatrix} p^T \\ \hat{Q} \end{pmatrix} U^T U \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q} \end{pmatrix} \begin{pmatrix} v^T \\ \tilde{R} \\ 0 \end{pmatrix},$$

where $\tilde{R} \in \mathbf{R}^{n\times n}$ is upper triangular, we have $v = z$ and the new QR decomposition

$$\tilde{X} = \tilde{Q}\begin{pmatrix}\tilde{R}\\ 0\end{pmatrix}.$$

For downdating the kth row of X, we can apply the same procedure (2.1) so that the kth row of Q is transformed to e_k^T. It is important in this algorithm that *the rows and columns of Q have unit 2-norm*: the kth column automatically becomes e_k when the kth row is transformed to e_k^T. This makes it possible that after deleting the kth row and kth column from the transformed orthogonal matrix, the remaining matrix stays orthogonal. This algorithm was introduced in [8] and shown to be mixed stable [12]. Its computational complexity is approximately $4m^2 + 2n^2$ flops (floating point operations) where 1 flop $\approx$ 1 addition and 1 multiplication.

3 Gram-Schmidt Downdating Algorithm for a QR Decomposition

In many applications, we do not need the full $m \times m$ orthogonal factor. Especially if $m \gg n$, it is costly to store and modify the full QR decomposition. When we use the *skinny* QR decomposition, the storage requirement for the Q factor is reduced to mn from m^2, compared to the full QR decomposition. In [7], algorithms are derived for modifying the skinny QR decomposition of a matrix when the matrix 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 (or modified Gram-Schmidt) process with reorthogonalization [7, 13, 14].

Assume that we have the skinny QR decomposition of X,

$$X = \begin{pmatrix}z^T\\ \tilde{X}\end{pmatrix} = Q_1 R = \begin{pmatrix}f^T\\ \hat{Q}_1\end{pmatrix} R \tag{3.1}$$

and want to obtain the new QR decomposition after deleting the first row z^T. Following [7], we first apply the Gram-Schmidt process (with reorthogonalization if necessary) so that $e_1 = (1\,0\,\cdots\,0)^T \in \mathbf{R}^{m\times 1}$ is orthogonalized to Q_1. Because of the special form of e_1, the result has the form

$$(\,Q_1 \quad e_1\,) = \begin{pmatrix}f^T & 1\\ \hat{Q}_1 & 0\end{pmatrix} = \begin{pmatrix}f^T & \alpha\\ \hat{Q}_1 & h\end{pmatrix}\begin{pmatrix}I_{n\times n} & f\\ 0 & \alpha\end{pmatrix} \tag{3.2}$$

for some $h \in \mathbf{R}^{(m-1)\times 1}$ and $\alpha \in \mathbf{R}$. If e_1 is linearly dependent on the columns of Q_1, then the orthogonalization will fail. For a solution, see [7].

The importance of this procedure is that after orthogonalizing, we have $f^T f + \alpha^2 = 1$ in (3.2). Therefore, *when the first row of* $\begin{pmatrix}f^T & \alpha\\ \hat{Q}_1 & h\end{pmatrix}$ *is transformed*

Algorithm GSD (Gram-Schmidt Downdating)

Given $Q_1 = \begin{pmatrix} f^T \\ \hat{Q}_1 \end{pmatrix} \in \mathbf{R}^{m\times n}$ and $R \in \mathbf{R}^{n\times n}$ for $X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = Q_1 R$, this algorithm computes the downdated $\tilde{Q}_1 \in \mathbf{R}^{(m-1)\times n}$ and $\tilde{R} \in \mathbf{R}^{n\times n}$ for $\tilde{X}$:

1. Orthogonalize e_1 to Q_1 by the Gram-Schmidt process with one reorthogonalization step, if necessary :
   (a) $v := e_1 - Q_1 f$
   (b) if $\|v\|_2 \geq 1/\sqrt{2}$, $v := v/\|v\|_2$
   (c) else
      i. $s := Q_1^T v$; $v' := v - Q_1 s$
      ii. if $\|v'\|_2 \leq \|v\|_2/\sqrt{2}$, $\alpha = 0$; determine h orthogonal to $\hat{Q}_1$, with $\|h\|_2 = 1$
      iii. else $v := v'/\|v'\|_2$
   (d) $\alpha := v(1)$; $h := v(2:m)$
2. Determine an orthogonal matrix U as in (3.3) :
$$\begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q}_1 \end{pmatrix} := \begin{pmatrix} f^T & \alpha \\ \hat{Q}_1 & h \end{pmatrix} U^T$$
3. Modify the upper triangular factor: $\begin{pmatrix} z^T \\ \tilde{R} \end{pmatrix} := U \begin{pmatrix} R \\ 0 \end{pmatrix}$

to $e_1^T \in \mathbf{R}^{1\times(n+1)}$, the first column will also become $e_1 \in \mathbf{R}^{m\times 1}$. From (3.1) and (3.2), we have

$$\begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = \begin{pmatrix} f^T & \alpha \\ \hat{Q}_1 & h \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix},$$

and now we can determine a sequence of plane rotations $J_{k,k+1}$, $k = n, \cdots, 1$, in the plane $(k, k+1)$ such that

$$U \begin{pmatrix} f \\ \alpha \end{pmatrix} \equiv J_{1,2} \cdots J_{n,n+1} \begin{pmatrix} f \\ \alpha \end{pmatrix} = e_1,$$

which gives

$$\begin{pmatrix} f^T & \alpha \\ \hat{Q}_1 & h \end{pmatrix} U^T = \begin{pmatrix} 1 & 0 \\ w & \tilde{Q}_1 \end{pmatrix}. \tag{3.3}$$

The transformed matrix has orthonormal columns, so $w = 0$. It follows that

$$\begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = \begin{pmatrix} f^T & \alpha \\ \hat{Q}_1 & h \end{pmatrix} U^T U \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q}_1 \end{pmatrix} \begin{pmatrix} v^T \\ \tilde{R} \end{pmatrix}$$

where $v = z$ and $\tilde{R}$ is upper triangular. Then the downdated QR decomposition is

$$\tilde{X} = \tilde{Q}_1 \tilde{R}. \tag{3.4}$$

The above is summarized in Algorithm GSD (Gram-Schmidt Downdating). As the downdating algorithm for the full QR decomposition, this algorithm relies on the fact that the first row and the first column of the matrix $\begin{pmatrix} f^T & \alpha \\ \hat{Q}_1 & h \end{pmatrix}$ have unit 2-norm. With one reorthogonalization process, the Algorithm GSD requires about $7mn + 2n^2$ flops. This can be reduced to $5mn + n^2$ flops when fast scaled rotations [2, 10] are used in Steps 2 and 3. Note that the data matrix X is never needed. Thus, the storage requirement is about $mn + 0.5n^2$ for Q_1 and R.

4 Recovering Orthogonality

In Sections 2 and 3, we have shown that the orthonormality of the Q factor plays an important role in downdating full or skinny QR decompositions. It is well known that the orthogonal factor computed as a product of Householder or Givens transformations is accurately orthogonal whereas that obtained from Gram-Schmidt may lose orthogonality when the condition number of X is large. In [5], it is shown that the full QR decomposition of a matrix $A \equiv \begin{pmatrix} 0_{n\times n} \\ X \end{pmatrix} \in \mathbf{R}^{(m+n)\times n}$ by Householder transformations can be completely determined by the skinny QR decomposition of X obtained by the Gram-Schmidt process. We will first present the relevant results from [5] and show how they can be used to obtain a more accurate downdating algorithm.

In the row oriented modified Gram-Schmidt(MGS) algorithm for computing the skinny QR decomposition of $X \in \mathbf{R}^{m\times n}, m \geq n = \text{rank}(X)$, a sequence of matrices,

$$X = X^{(1)}, X^{(2)}, \ldots, X^{(n)} = Q_1 \in \mathbf{R}^{m\times n}$$

is computed where $X^{(k)}$ has the columns

$$X^{(k)} = (q_1, \ldots, q_{k-1}, x_k^{(k)}, \ldots, x_n^{(k)})$$

and $x_k^{(k)}, \ldots, x_n^{(k)}$ have been made orthogonal to $q_1, \ldots, q_{k-1}$. In the kth step, we compute

$$r_{kk} = ({x_k^{(k)}}^T x_k^{(k)})^{\frac{1}{2}}, \qquad q_k = \frac{x_k^{(k)}}{r_{kk}}. \tag{4.1}$$

Then $x_{k+1}^{(k)}, \cdots, x_n^{(k)}$ are orthogonalized against q_k as

$$x_j^{(k+1)} = x_j^{(k)} - r_{kj} q_k, \qquad r_{kj} = q_k^T x_j^{(k)}, \qquad j = k+1, \cdots, n. \tag{4.2}$$

After n steps of orthogonalization, we have the decomposition $X = Q_1 R$ where the columns of Q_1 are orthonormal by construction, in exact arithmetic. The orthonormality of the computed factor Q_1 from the MGS algorithm depends on the condition number of X. For a detailed analysis, see [3].

Now we present the relation between the Householder QR decomposition applied to a matrix X augmented with a square matrix of zero elements on top and the modified Gram-Schmidt QR decomposition applied to X [5], which we use in deriving our new downdating algorithm. Suppose the Householder QR decomposition gives the full QR decomposition of $A \in \mathbf{R}^{(m+n)\times n}$ as

$$A \equiv \begin{pmatrix} 0_{n\times n} \\ X \end{pmatrix} = P \begin{pmatrix} R \\ 0 \end{pmatrix} \tag{4.3}$$

where $P \in \mathbf{R}^{(m+n)\times(m+n)}$ is orthogonal and $R \in \mathbf{R}^{n\times n}$ is upper triangular. Let

$$A^{(1)} \equiv \begin{pmatrix} 0_{n\times n} \\ X^{(1)} \end{pmatrix} = \begin{pmatrix} 0_{n\times n} \\ X \end{pmatrix}.$$

Then the first Householder transformation P_1 applied to $A^{(1)}$ is

$$P_1 = I - u_1 u_1^T, \qquad u_1 = \begin{pmatrix} -e_1 \\ q_1 \end{pmatrix}.$$

Also

$$P_1 a_j^{(1)} = a_j^{(1)} - u_1 u_1^T a_j^{(1)} = \begin{pmatrix} 0 \\ x_j^{(1)} \end{pmatrix} - \begin{pmatrix} -e_1 \\ q_1 \end{pmatrix} q_1^T x_j^{(1)} = \begin{pmatrix} e_1 \rho_{1j} \\ x_j^{(2)} \end{pmatrix}, \quad j = 2, \cdots, n$$

Note that ρ_{1j} is the same as r_{1j} in (4.2), $j = 1, \cdots, n$ and $x_j^{(2)}, j = 2, \cdots, n$ are those obtained by applying the first step of the MGS algorithm to X. It can be easily verified that the next step of the Householder transformation P_2 produces the second row of R and $x_3^{(3)}, \cdots, x_n^{(3)}$ just as in the MGS algorithm. After performing n steps of Householder transformations, we have (4.3) with $P \equiv (P_n \cdots P_1)^T = P_1 \cdots P_n$.

Ideally, if $q_i^T q_j = 0$ when $i \neq j$, the orthogonal matrix P in (4.3) has the form

$$P \equiv P_1 \cdots P_n = I - \sum_{k=1}^{n} u_k u_k^T = \begin{pmatrix} 0_{n\times n} & Q_1^T \\ Q_1 & I - Q_1 Q_1^T \end{pmatrix} \tag{4.4}$$

where $u_k^T = (\, -e_k^T \quad q_k^T \,)$, $P_k = I - u_k u_k^T$. Note that P is fully defined by its (2,1) block, Q_1. Let $\bar{Q}_1 = (\bar{q}_1, \cdots, \bar{q}_n)$ be the matrix of vectors *computed* by the MGS algorithm in finite precision arithmetic. It is well known that the orthogonality of $\bar{Q}_1$ depends on the condition number of X and the departure of $\bar{Q}_1^T\bar{Q}_1$ from the unit matrix can be significant. Therefore, the computed $\bar{Q}_1$ does not necessarily satisfy $\bar{q}_i^T\bar{q}_j = 0$ when $i \neq j$, and under this condition (4.4) becomes

$$\text{(4.5)} \qquad \begin{aligned} \bar{P} &\equiv \bar{P}_1 \cdots \bar{P}_n = I - \sum_{k=1}^{n} \bar{u}_k\bar{u}_k^T = \begin{pmatrix} \bar{P}_{11} & \bar{P}_{12} \\ \bar{P}_{21} & \bar{P}_{22} \end{pmatrix} \\ &= \begin{pmatrix} \bar{P}_{11} & (I-\bar{P}_{11})\bar{Q}_1^T \\ \bar{Q}_1(I-\bar{P}_{11}) & I-\bar{Q}_1(I-\bar{P}_{11})\bar{Q}_1^T \end{pmatrix} \end{aligned}$$

where $\bar{u}_k^T = (-e_k^T\ \bar{q}_k^T)$ and $\bar{P}_k = I - \bar{u}_k\bar{u}_k^T$. In order to observe the improvement in orthogonality in $\bar{P}$ over $\bar{Q}_1$, define the orthonormal equivalent P'_k such as

$$q'_k = \bar{q}_k/\|\bar{q}_k\|_2, \quad u'_k = \begin{pmatrix} -e_k \\ q'_k \end{pmatrix}, \quad P'_k = I - u'_k u'^T_k$$

and define $P' \equiv P'_1 \cdots P'_n$. According to an error analysis on the Householder QR decomposition algorithm [16], $\bar{P}$ obtained from Householder transformations applied on A and $\bar{R}$ computed by the MGS algorithm on X satisfy

$$\text{(4.6)} \qquad \begin{pmatrix} E_1 \\ X+E_2 \end{pmatrix} = P' \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix}, \qquad \bar{P} = P' + E'$$

$$\|E_i\|_2 \le c_i\epsilon\|X\|_2, \quad i = 1,2, \quad \|E'\|_2 \le c_3\epsilon$$

where c_i, $1 \le i \le 2$, are constants depending on m, n and the details of arithmetic and ϵ is the unit round off. These error bounds show that the computed $\bar{P}$ is very close to the exactly orthogonal P' while the orthogonality of $\bar{Q}_1$ depends on the condition number of X [5]. Therefore, *it is better to use $\bar{P}_{21} = \bar{Q}_1(I-\bar{P}_{11})$ instead of $\bar{Q}_1$ when the orthonormality of a computed factor is crucial for a given problem.*

5 New Algorithm for Downdating a Skinny QR Decomposition

In this section, we present a new algorithm for downdating a skinny QR decomposition. It is based on a relation between the Gram-Schmidt downdating algorithm and downdating of a row in the decomposition (4.3). Consider the two decompositions

$$\text{(5.1)} \qquad X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = Q_1 R \quad \text{and} \quad A = \begin{pmatrix} 0_{n\times n} \\ X \end{pmatrix} = P \begin{pmatrix} R \\ 0 \end{pmatrix}$$

where $Q_1 \in \mathbf{R}^{m\times n}$ and $P \in \mathbf{R}^{(m+n)\times(m+n)}$ have orthonormal columns. Since deleting the first row of X is equivalent to deleting the $(n+1)$th row of A, the downdated new upper triangular factor $\tilde{R}$ for $\tilde{X}$ can be obtained by downdating the full QR decomposition of A. This can be done if we have the $(n+1)$th row of the orthogonal factor P.

We first show that if the columns of Q_1 are exactly orthonormal, then the Gram-Schmidt downdating algorithm for deleting the first row of X is equivalent to downdating the $(n+1)$th row of A from the full QR decomposition of A shown in (5.1). From (4.4), we have

$$A = \begin{pmatrix} 0_{n\times n} \\ X \end{pmatrix} = \begin{pmatrix} 0_{n\times n} & Q_1^T \\ Q_1 & I - Q_1Q_1^T \end{pmatrix} \begin{pmatrix} R \\ 0_{m\times n} \end{pmatrix} \tag{5.2}$$

and if we have the first row $(f^T \; g^T)$ of $(\, Q_1 \quad I - Q_1Q_1^T \,)$, then we can determine the downdating transformations for deleting the first row of X. We first choose a Householder matrix H that gives

$$g^T H = (\pm\|g\|_2 \; 0 \; \cdots \; 0) \in \mathbf{R}^{1\times m}. \tag{5.3}$$

Then, from (5.2) and (5.3),

$$\begin{aligned} X &= (\, Q_1 \quad (I - Q_1Q_1^T)\,) \begin{pmatrix} R \\ 0_{m\times n} \end{pmatrix} \\ &= (\, Q_1 \quad (I - Q_1Q_1^T)H\,) \begin{pmatrix} R \\ 0_{m\times n} \end{pmatrix} \\ &= (\, Q_1 \quad ((I - Q_1Q_1^T)H)e_1\,) \begin{pmatrix} R \\ 0_{1\times n} \end{pmatrix} \end{aligned} \tag{5.4}$$

since a transformation that operates on $I - Q_1Q_1^T$ does not affect the upper triangular factor R. In (5.4), the matrix

$$(\, Q_1 \quad (I - Q_1Q_1^T)He_1\,) \in \mathbf{R}^{m\times(n+1)}$$

has orthonormal columns and its first row has unit 2-norm since

$$g = e_1 - Q_1 f, \quad He_1 = \pm\frac{g}{\|g\|_2}, \quad (I - Q_1Q_1^T)He_1 = \pm\frac{g}{\|g\|_2},$$

where e_1's denote the unit vector with 1 in the first component, but the dimension may be different in each occurrence. We can now determine the downdating transformation U as in Step 2 of Algorithm GSD. The vector $(I - Q_1Q_1^T)He_1$ obtained in the above process is the same as the vector obtained from orthogonalizing e_1 to Q_1 in Algorithm GSD. Therefore, *the Gram-Schmidt downdating algorithm for the skinny QR decomposition of X is mathematically equivalent to downdating the full Householder QR decomposition of A.*

The key idea in our new downdating algorithm is that the downdating transformation is computed based on the $(n+1)$th row of $\bar{P}$, where its structure [5] is

$$\bar{P} = \begin{pmatrix} \bar{P}_{11} & \bar{P}_{12} \\ \bar{P}_{21} & \bar{P}_{22} \end{pmatrix}$$

$$(5.5)\quad \left(\begin{array}{cccccc|c} 0 & \bar{q}_1^T\bar{q}_2 & \bar{q}_1^T\bar{M}_2\bar{q}_3 & \cdots & \bar{q}_1^T\bar{M}_2\bar{M}_3\cdots\bar{M}_{n-1}\bar{q}_n & & \bar{q}_1^T\bar{M}_2\bar{M}_3\cdots\bar{M}_n \\ 0 & 0 & \bar{q}_2^T\bar{q}_3 & \cdots & \bar{q}_2^T\bar{M}_3\bar{M}_4\cdots\bar{M}_{n-1}\bar{q}_n & & \bar{q}_2^T\bar{M}_3\bar{M}_4\cdots\bar{M}_n \\ \vdots & \vdots & \vdots & \cdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \bar{q}_{n-1}^T\bar{q}_n & & \bar{q}_{n-1}^T\bar{M}_n \\ 0 & 0 & 0 & \cdots & 0 & & \bar{q}_n^T \\ \hline \bar{q}_1 & \bar{M}_1\bar{q}_2 & \bar{M}_1\bar{M}_2\bar{q}_3 & \cdots & \bar{M}_1\bar{M}_2\cdots\bar{M}_{n-1}\bar{q}_n & & \bar{M}_1\bar{M}_2\cdots\bar{M}_n \end{array}\right)$$

where

$$\bar{P}_{11} \in \mathbf{R}^{n\times n},\ \bar{P}_{12} \in \mathbf{R}^{n\times m},\ \bar{P}_{21} \in \mathbf{R}^{m\times n},\ \bar{P}_{22} \in \mathbf{R}^{m\times m},$$
$$\bar{M}_i = I - \bar{q}_i\bar{q}_i^T, i = 1,\cdots,n.$$

Thus the effect is as if the downdating transformation is computed based on the full QR decomposition with the orthogonal factor computed from the Householder transformations.

In the new algorithm, we begin with recovering the first row, $\bar{g}^T$ of $\bar{P}_{22} = \bar{M}_1\cdots\bar{M}_n$, recursively from

$$\bar{g}^T = (((e_1^T\bar{M}_1)\bar{M}_2)\cdots\bar{M}_n). \tag{5.6}$$

Then we determine the first column of a Householder matrix $\bar{H}$ that makes $\bar{g}^T\bar{H} = (\|\bar{g}\|_2\ 0\cdots 0)$, and compute the first column, $\bar{v}$ of $\bar{P}_{22}\bar{H}$, from

$$\bar{v} = \bar{P}_{22}\bar{H}e_1 = (\bar{M}_1\cdots(\bar{M}_n(\bar{H}e_1))). \tag{5.7}$$

These steps replace the steps for orthogonalizing e_1 to Q_1 in Algorithm GSD. Also, instead of using the first row of $\bar{Q}_1$ as in Algorithm GSD, we recover the first row $\bar{f}^T$ of $\bar{P}_{21}$ from

$$e_{n+1}^T\bar{P} = e_{n+1}^T\bar{P}_1\cdots\bar{P}_n, \quad e_{n+1} \in \mathbf{R}^{(m+n)\times 1}, \tag{5.8}$$

where e_{n+1} is a unit vector with 1 on its $(n+1)$th position. Although it is possible to obtain $\bar{f}^T$ by extracting the first row of (2,1) block, $\bar{P}_{21}$ in (5.5), this method is much more costly than extracting the $(n+1)$th row of $\bar{P}$ as we did in (5.8). From the structure of $\bar{P}$ shown in (5.5), we can see that after the downdating transformation is computed and applied to $\bar{P}$ based on the first row of $(\bar{P}_{21}\ \bar{v})$, the first column of $\bar{P}$ becomes $e_{n+1} \in \mathbf{R}^{(m+n)\times 1}$. Therefore, the first column of $\bar{P}_{21}$ becomes $e_1 \in \mathbf{R}^{m\times 1}$ as far as the first column of $\bar{P}_{21}$ had a unit norm. (Note that the first column of $\bar{P}_{11}$ is 0.) We summarize the new algorithm in Algorithm HGSD (Householder-Gram-Schmidt Downdating).

Algorithm HGSD (Householder-Gram-Schmidt Downdating)

Given $Q_1 = \begin{pmatrix} f^T \\ \hat{Q}_1 \end{pmatrix} \in \mathbf{R}^{m\times n}$ and $R \in \mathbf{R}^{n\times n}$ for $X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = Q_1 R$, this algorithm computes the downdated $\tilde{Q}_1 \in \mathbf{R}^{(m-1)\times n}$ and $\tilde{R} \in \mathbf{R}^{n\times n}$ for $\tilde{X}$:

1. Compute $\begin{pmatrix} \alpha \\ h \end{pmatrix}$ which is orthogonal to Q_1:
   - (a) Compute the first row, g^T of P_{22} as in (5.6):
     $g := e_1 \in \mathbf{R}^{m\times 1}$
     for $i = 1$ to n, $g^T := g^T - (g^T Q_1(:,i))Q_1(:,i)^T$
   - (b) Determine the Householder vector x for H: $x := g + \mathrm{sign}(g_1)\|g\|_2 e_1$.
   - (c) Compute the first column of $P_{22}H$ as in (5.7):
     $v := He_1 = e_1 - 2x(x^T e_1)/x^T x$.
     for $i = n$ down to 1, $v := v - Q_1(:,i)(Q_1(:,i)^T v)$
   - (d) $\alpha := v(1)$; $h := v(2:m)$.
2. Refine the first row of Q_1 by recovering the first row, f of P_{21} as in (5.8):
   - (a) $f := e_{n+1} \in \mathbf{R}^{(m+n)\times 1}$
     for $i = 1$ to n, $u := [-I_{n\times n}(:,i); Q_1(:,i)]$, $f^T := f^T - (f^T u)u^T$
   - (b) $f := f(1:n)$.
3. Determine an orthogonal matrix U as in (3.3) :
$$\begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q}_1 \end{pmatrix} := \begin{pmatrix} f^T & \alpha \\ \hat{Q}_1 & h \end{pmatrix} U^T$$
4. Modify the upper triangular factor: $\begin{pmatrix} z^T \\ \tilde{R} \end{pmatrix} := U \begin{pmatrix} R \\ 0 \end{pmatrix}$

In order to simplify the presentation,we drop all the $\bar{\cdot}$ (bar) notations for the computed quantities such as $\bar{f}, \bar{g}$ and $\bar{Q}_1$. Step 1(a) as well as Step 1(c) in Algorithm HGSD requires $2mn$ flops because there are n iterations, with each iteration costing $2m$ flops due to one inner product and one vector updating. Also Step 2 requires about $2mn$ flops. Thus the overall computational complexity is approximately $10mn + 2n^2$.

6 Numerical Experiments

In this section, we discuss the numerical results obtained by solving least squares problems to compare the performance of Algorithm GSD and Algorithm HGSD. The QR decomposition is most widely used to solve linear least squares problems under the assumption that the involved matrix is of full rank. Depending on whether full QR decomposition or skinny QR decomposition is used, the technique differs. For a given matrix X and a right hand side vector s, the skinny QR decomposition on an augmented matrix $(X\ s)$ such that

$$(\,X \quad s\,) = (\,Q_1 \quad q\,)\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}$$

yields the solution w to $\min \|Xw - s\|$ from $Rw = u$. When a row of X and the corresponding element of s are deleted, the solution to the new least squares problem can be obtained using QR decomposition downdating rather than the new QR decomposition computed from scratch.

The tests we have performed solve a series of least squares problems, defined by the downdated data with the top row of the data matrix and the top element of the right hand side vector removed each time. Initially, the skinny QR decomposition is computed on the matrix formed by attaching an $m \times 1$ right hand side vector s as the last column to an $m \times n$ data matrix X and the solution vector w is obtained. After each step, the first row of the data matrix and the first element of the right hand side vector are deleted, and the QR decomposition of the data matrix attached with the right hand side vector is downdated to obtain the new least squares solution vector. Therefore, at the kth step, we solve a least squares problem with an $(m-k+1) \times n$ data matrix X and an $(m-k+1) \times 1$ right hand side vector s.

Also, at each step, we consider the forward error and the backward error. In order to estimate the forward error, w_H is computed as a reference using Householder QR decomposition on the windowed data matrix attached with the corresponding subsection of the right hand side vector and then w_H is compared to the solution w obtained by two downdating algorithms as :

$$\text{relative forward error in solution} = \frac{\|w - w_H\|_2}{\|w_H\|_2}.$$

Also the Q_1 and R obtained from downdating are compared to Q_{1H} and R_H from Householder QR decomposition on the windowed matrix as :

$$\text{relative forward error in } \mathrm{Q}_1 \text{ factor} = \frac{\|Q_1 - Q_{1H}\|_F}{\|Q_{1H}\|_F},$$

$$\text{relative forward error in R factor} = \frac{\|R - R_H\|_F}{\|R_H\|_F}.$$

We also use the following error estimate as suggested in [4] :

$$\text{backward error for R factor} = \min_V \|V \begin{pmatrix} R \\ 0 \end{pmatrix} - X\|_2, \quad V^T V = I.$$

This can be computed by solving the orthogonal Procrustes problem [10]. Finally the backward error for the least squares problem $\min \|Xw - s\|_2$ is estimated as shown in [15].

$$\begin{aligned} \text{backward error for LS solution} &= \min\{\frac{\|r\|_2}{\|w\|_2}, \sigma_{min}([X\ T])\} \\ &= \min\{\|E\|_F : \|(X+E)w - s\|_2 \\ &= \min_{w'} \|(X+E)w' - s\|_2\} \end{aligned}$$

where

$$r = s - Xw, \qquad T = \frac{\|r\|_2}{\|w\|_2}(I - rr^T)$$

and σ_{min} is the smallest singular value of the matrix $(X\ T)$.

The following three tests have been implemented in Matlab version 4.2 with IEEE double precision floating point arithmetic (machine $\epsilon \approx 2.2204 \times 10^{-16}$). Similar test data have been used in [6] to compare other downdating algorithms to Algorithm GSD. The results from Test I, II and III are shown in Figure 1, 2 and 3, respectively. For each figure, the upper left graph shows the forward error in Euclidean norm in the downdated solution vector w, the upper right graph shows the condition number of the data matrix X before removing the top row, the middle left graph shows the forward error in Frobenius norm in the downdated R factor and the middle right graph shows the forward error in Frobenius norm in the downdated Q factor. The lower left graph shows the backward error for the solution vector w which is obtained from the downdated R and the lower right graph shows the backward error for the downdated R.

Test I The data matrix X is a 50×5 Hilbert matrix. The right hand side vector s is constructed by multiplying X to $(1\ 1\ 1\ 1\ 1)^T$ and then each element is perturbed by a random number uniformly distributed in $(0,\ 10^{-6})$. The solution to $\min \|Xw - s\|_2$ is computed using the skinny QR decomposition on $(X\ s)$ and then the downdating steps with Algorithm GSD and Algorithm HGSD are applied 40 times. The forward error in the solution as well as the forward errors in R and Q_1 increase as the condition number of X increases. The forward solution error after the first step in Algorithm GSD is of order 10^{-9} and increases to almost 1 after 40 steps while the forward solution error from Algorithm HGSD starts from around 10^{-12} and reaches only near 10^{-5}. Both backward errors for the LS solution and the R factor are smaller with Algorithm HGSD. In particular, the backward error for the LS solution with Algorithm HGSD stays close to machine ϵ. Throughout the test, Algorithm HGSD performs consistently better than Algorithm GSD.

Test II The data matrix X is a 60×3 random matrix with elements generated from (0, 1) following uniform distribution, where the second column is made close to the first column by adding a very small number ($\sqrt{\epsilon}\times$ a random number from (0, 1)) to each element of the first column, which creates near linear dependency between the first and the second columns. Also a random number of order 10^5 is added to the $(18, 3)$ component of X as an outlier. The right hand side vector s is a 60×1 random vector generated from (0, 100). As in Test I, the skinny QR decomposition on $(X\ s)$ is initially computed followed by downdating with Algorithm GSD and Algorithm HGSD. The downdating step is repeated 50 times. The forward solution errors from both algorithms are of order 10^{-9} initially. Then, as is shown in Figure 2, we observe large increases in all types of the errors from Algorithm GSD near the 18th step when the outlier is removed since the downdating problem is very ill-conditioned at this step. This phenomenon was explained in [6]. However, the forward solution error from Algorithm HGSD remains around 10^{-9}. Also the backward error for the solution from Algorithm HGSD stays near machine ϵ while that from Algorithm GSD shows a large increase at the 18th step. The backward error for R from Algorithm HGSD stays close to 10^{-8} throughout the test, contrary to that from Algorithm GSD.

Test III The data matrix X and the right hand side vector s are similar to those in Test II. The matrix X is a random 60×3 matrix with an outlier of order 10^5 in the (18,3) position. The norm of the first row of Q_1 from the skinny QR decomposition of X is made close to 1 in order to observe the behavior of both algorithms when e_1 is almost linearly dependent on the columns of Q_1. In this case, orthogonalizing e_1 to Q_1 can be difficult. Initially when the data matrix is relatively well-conditioned, the errors for both algorithms, except the backward error for R, are near 10^{-16} or less. After the outlier is removed at the 18th step, both algorithms show large increases in the errors. The difference between two algorithms is not as pronounced as in Test II although Algorithm HGSD is consistently better.

Algorithm HGSD outperforms Algorithm GSD in many other test cases, implicitly utilizing the full QR decomposition when downdating. Finally, we would like to mention the observations we made while performing the tests, which deserve further investigations. The initial Q factor seems crucial to Algorithm GSD. When the accuracy of Q_1 was improved by using $\bar{P}_{21}$ block in (5.5) at the beginning of Algorithm GSD, a large improvement was observed in the solution. In particular, the result for Test II with this scheme was close to that with Algorithm HGSD. In many cases, the accuracy of the Q factor rather than that of the R factor had more effect on the accuracy of the solution. It is yet to be studied what effect Q has on the solution other than providing an accurate R factor obtained from accurate downdating.

REFERENCES

1. S. T. Alexander, C. T. Pan, and R. J. Plemmons. *Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing.* Linear Algebra and Its Applications, 98(1988), pp. 3–40.
2. A.A. Anda and H. Park. *Fast plane rotations with dynamic scaling.* SIAM J. Matrix Anal. Appl., 15(1994), pp. 162–174.
3. A. Björck. *Solving linear least squares problems by Gram-Schmidt orthogonalization.* BIT, 7(1967), pp. 1-21.
4. A. Björck. *Error analysis of least squares algorithms.* In Numerical Linear Algebra, Digital signal Processing and Parallel Algorithms, G.H. Golub and P. Van Dooren, eds., NATO ASI Series, Vol. F70, Springer-Verlag Berlin Heidelberg, 1991.
5. A. Björck and C. C. Paige. *Loss and Recapture of orthogonality in the modified Gram-Schmidt algorithm.* SIAM J. Matrix Anal. Appl., 13(1992), pp. 176–190.
6. A. Björck, H. Park and L. Eldén. *Accurate downdating of least squares solutions.* SIAM J. Matrix Anal. Appl., 15(1994), pp. 549–568.
7. J. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization.* Mathematics of Computation, 30(1976), pp. 772–95.
8. P. E. Gill, G. H. Golub, W. Murray and M. A. Saunders. *Methods for modifying matrix factorizations.* Mathematics of Computation, 28(1974), pp. 505–535.
9. G. H. Golub. *Matrix decompositions and statistical computations.* In Statistical Computation, R. C. Milton and J. A. Nelder, editors, Academic Press, New York, 1969, pp. 365–397.
10. G. H. Golub and C. F. Van Loan. *Matrix Computations. 2nd ed.* Johns Hopkins Press, Baltimore, MD., 1989.
11. S. Haykin. *Adaptive Filter Theory. 2nd ed.* Prentice Hall, Englewood Cliffs, NJ, 1991.
12. C. C. Paige. *Error analysis of some techniques for updating orthogonal decompositions.* Mathematics of Computation, 34(1980), pp. 465–471.
13. B.N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall Englewood Cliffs, NJ., 1980.
14. L. Reichel and W. B. Gragg. *FORTRAN subroutines for updating the QR decomposition.* ACM Trans. Math. Software, 16(1990), pp. 369–377.
15. B. Waldén, R. Karlson and J. Sun. *Optimal backward perturbation bounds for the linear least squares problem.* Numerical Linear Algebra and Applications, 2(1995), pp 271–286.
16. J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, England, 1965.

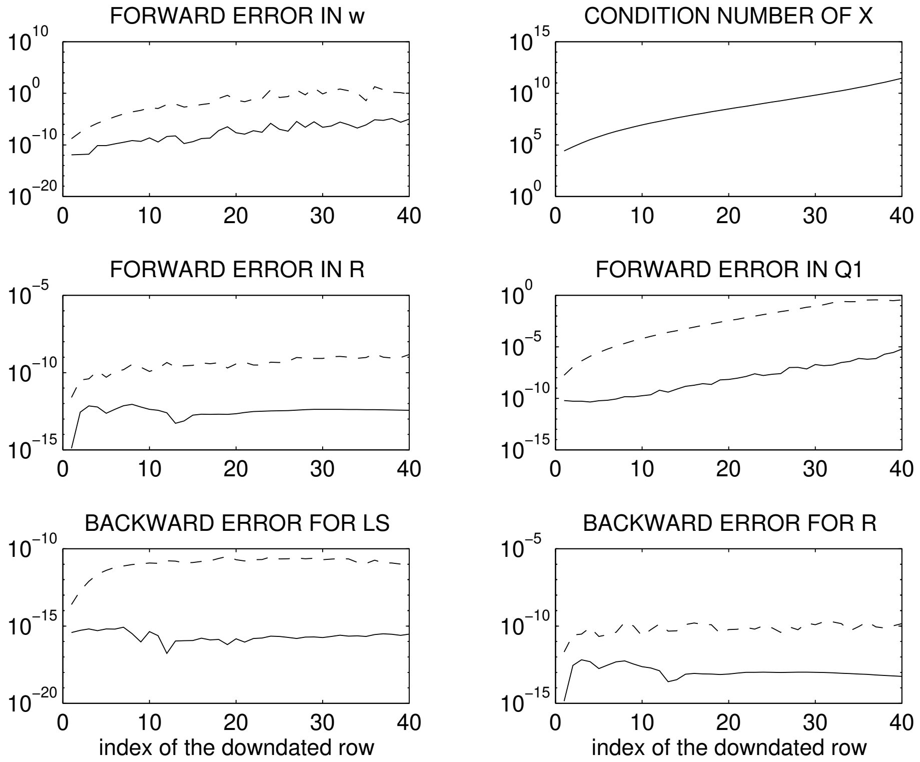

Figure 6.1: Comparison of Algorithm GSD(dashed) and Algorithm HGSD(solid) on Data I

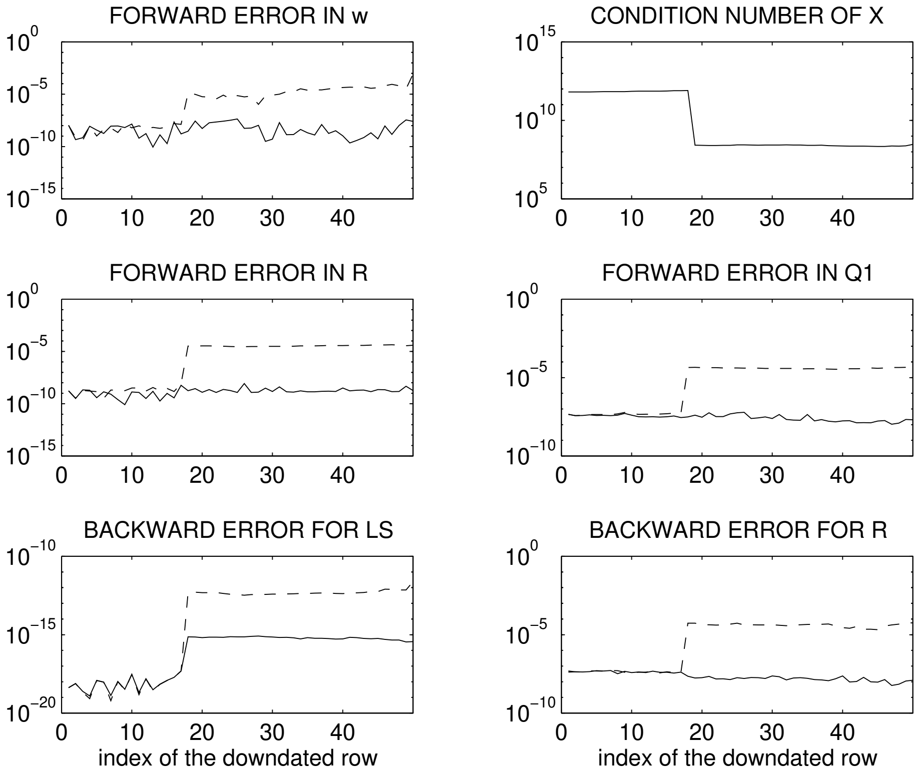

Figure 6.2: Comparison of Algorithm GSD(dashed) and Algorithm HGSD(solid) on Data II

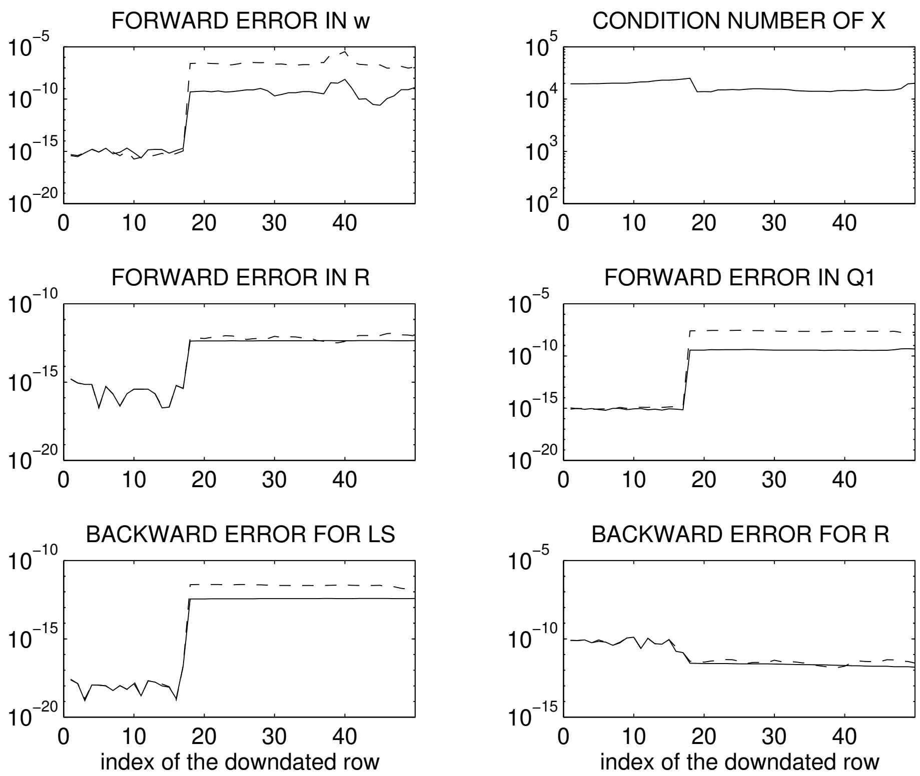

Figure 6.3: Comparison of Algorithm GSD(dashed) and Algorithm HGSD(solid) on Data III