Journal of Computational and Applied Mathematics 27 (1989) 109–127
North-Holland

Least squares modifications with inverse factorizations: parallel implications

C.-T. PAN * and R.J. PLEMMONS **
Departments of Computer Science and Mathematics, North Carolina State University, Raleigh, NC 27695-8205, U.S.A.

Received 8 February 1988
Revised 15 July 1988

Abstract: The process of modifying least squares computations by updating the covariance matrix has been used in control and signal processing for some time in the context of linear sequential filtering. Here we give an alternative derivation of the process and provide extensions to downdating. Our purpose is to develop algorithms that are amenable to implementation on modern multiprocessor architectures. In particular, the inverse Cholesky factor R^{-1} is considered and it is shown that R^{-1} can be updated (downdated) by applying the same sequence of orthogonal (hyperbolic) plane rotations that are used to update (downdate) R. We have attempted to provide some new insights into least squares modification processes and to suggest parallel algorithms for implementing Kalman type sequential filters in the analysis and solution of estimation problems in control and signal processing.

1. Introduction

The linear least squares problem is one of the oldest topics in applied mathematics; yet new applications and computational techniques are receiving ever increasing attention. Significantly these computations have numerous important applications throughout modern engineering and science [6].

Let X be a real $m \times n$ matrix with full column rank n and let s be a real m-vector. Then the *linear least squares problem* is to solve the possibly overdetermined system of equations

$$Xw = s \tag{1.1}$$

by computing a particular vector w such that

$$\| s - Xw \| \tag{1.2}$$

is minimized. (Here and throughout this paper $\| \cdot \|$ denotes the Euclidean norm.)

In the past decade several algorithms have been proposed and analyzed [2,3,5,7,17,20–22, 27,33,35,41,42,49] for solving a succession of problems after the addition or deletion of an equation (least squares observation) in (1.1). Such procedures are called least squares *updating* and *downdating*, respectively. These algorithms find applicability in a variety of applications (see

* Current address: Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL 60115, U.S.A.
Research supported in part by Air Force Grant no. AFOSR-83-0255-D to NC State University.
** Research supported in part by Air Force Grant no. AFOSR-83-0255-D and National Science Foundation Grant no. DMS-85-21154.

[6]), especially in the area of linear estimation and prediction in control and signal processing [1,5,9,10,30,33,35,39]. The applications are very broad here and encompass adaptive antennas and beamforming [42,48], data communications [10], space navigation [5] and system identification [1], to name just a few. Our purpose in this paper is to consider fast algorithms for updating and downdating least squares problems using modern parallel architectures.

The solution to the linear least squares problem (1.2) can be obtained by forming and solving the *normal equations*

$$X^{\mathrm{T}}Xw = X^{\mathrm{T}}s. \tag{1.3}$$

If R denotes the upper triangular *Cholesky factor* of $X^{\mathrm{T}}X$, i.e., $R^{\mathrm{T}}R = X^{\mathrm{T}}X$, then w is obtained by solving the triangular systems

$$R^{\mathrm{T}}v = X^{\mathrm{T}}s, \qquad Rw = v$$

where v is an intermediate vector. However, in most applications where accuracy and stability are important (see e.g., [5,25]), R is computed directly from X by a sequence of orthogonal transformations; that is,

$$QX = \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad Q^{\mathrm{T}}Q = I. \tag{1.4a}$$

Then setting

$$Qs = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{1.4b}$$

where s_1 is an n-vector, w satisfies the upper triangular system

$$Rw = s_1.$$

Such schemes are called "square root" methods in the signal processing literature [1,5,30].

In many applications, for example in signal processing, very often it is required to recalculate w when observations (i.e., equations) are successively added to and/or deleted from the linear system (1.1). Here one wishes to add a row vector y^{T} to X and/or delete a row vector z^{T} from X. In these situations we seek to solve the modified least squares problem of minimizing

$$\| s' - X'w \| \tag{1.5}$$

for the new least squares vector w', where X' denotes the result of adding y^{T} to or deleting z^{T} from X, and s' denotes the corresponding modification to s. Clearly one should be able to compute w' by modifying the factors R and s_1 in (1.4), without performing a complete refactorization of X'. In particular, algorithms involving $\mathrm{O}(n^2)$ rather than $\mathrm{O}(mn^2)$ multiplications are sought. Also, in many applications, one considers the updating and downdating problems as separate processes, both of which occur at each recursive time step (see e.g., [2,3,11,42]). Rank two modifications arising in optimization applications are discussed in [14].

The updating problem can be described more fully as follows. Without loss of generality assume that the additional data vector y and scalar σ for the equation

$$y^{\mathrm{T}}w = \sigma$$

are appended to the end of X and s, forming

$$X' = \begin{bmatrix} X \\ y^{\mathrm{T}} \end{bmatrix}, \qquad s' = \begin{bmatrix} s \\ \sigma \end{bmatrix}.$$

It follows that

$$X'^{\mathrm{T}}X' = X^{\mathrm{T}}X + yy^{\mathrm{T}} = R^{\mathrm{T}}R + yy^{\mathrm{T}}$$

and thus the Cholesky factor of the updated matrix X, which we denote by U, can be calculated by applying orthogonal rotations to $[{R \atop y^{\mathrm{T}}}]$ in order to reduce the row vector y^{T} to zero, as described in [21]. Of course, the same orthogonal rotations are also applied to the vector $[{s_1 \atop \sigma}]$ where s_1 is given by (1.4b). More precisely, if

$$Q'\begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U \\ 0^{\mathrm{T}} \end{bmatrix}, \qquad Q'\begin{bmatrix} s_1 \\ \sigma \end{bmatrix} = \begin{bmatrix} s_1' \\ \sigma' \end{bmatrix} \tag{1.6}$$

where $Q'^{\mathrm{T}}Q' = I$, then

$$U^{\mathrm{T}}U = R^{\mathrm{T}}R + yy^{\mathrm{T}} = X'^{\mathrm{T}}X'.$$

Consequently the updated least squares vector w' is the solution to the triangular system

$$Uw' = s_1'.$$

The least squares updating problem is known to be well-conditioned and the method described above for computing w' is known to be stable [20,21,25]. Generally the scheme for updating R to U using orthogonal plane rotations [21] requires $2n^2 + \mathrm{O}(n)$ multiplications. Alternate schemes for computing U requiring fewer multiplications have been devised [17,20,21]. These schemes can be derived in terms of a Cholesky factorization of the matrix

$$I + aa^{\mathrm{T}}, \quad a = R^{-\mathrm{T}}y.$$

Here

$$U^{\mathrm{T}}U = R^{\mathrm{T}}R + yy^{\mathrm{T}} = R^{\mathrm{T}}(I + aa^{\mathrm{T}})R$$

and if A is the Cholesky factor of $I + aa^{\mathrm{T}}$, i.e., A is upper triangular and

$$A^{\mathrm{T}}A = I + aa^{\mathrm{T}},$$

then $R' = AR$. A scheme based upon this approach can be implemented with $\frac{3}{2}n^2 + \mathrm{O}(n)$ multiplications. Moreover, if an $L\Lambda L^{\mathrm{T}}$ factorization of $X^{\mathrm{T}}X$ is used, L unit lower triangular and Λ diagonal, then an updating algorithm can be implemented in $n^2 + \mathrm{O}(n)$ multiplications. But such schemes based upon the factorization of $I + aa^{\mathrm{T}}$ generally achieve less accuracy than the orthogonal methods [17,38] and, moreover, are less amenable to parallel implementation [2,3,32,46]. The $L\Lambda L^{\mathrm{T}}$ factorization causes additional numerical difficulties [17,38] and algorithms involving this factorization will not be considered here.

The downdating problem can be described in a similar manner. Here the purpose is to remove the effects of an observation on w; that is, to remove a row z^{T} from X and a scalar η from s, corresponding to the equation

$$z^{\mathrm{T}}w = \eta.$$

Without loss of generality, suppose that z^{T} is the last row of X, so that if X' and s' are the downdated factors in (1.1), then X' and s' are related to X and s by

$$X = \begin{bmatrix} X' \\ z^{\mathrm{T}} \end{bmatrix}, \qquad s = \begin{bmatrix} s' \\ \eta \end{bmatrix}.$$

For the orthogonal factorization of X given by (1.4a), it follows that

$$X'^{\mathrm{T}}X' = R^{\mathrm{T}}R - zz^{\mathrm{T}}.$$

Assuming that X' retains full column rank, the downdating problem is now to use the Cholesky factor R of $X^{\mathrm{T}}X$ to compute the *downdated Cholesky factor* of X', which we denote by D. But now the orthogonal rotation scheme given for the updating problem does not apply directly due to the negative sign of zz^{T}. An approach to downdating by the use of orthogonal transformations has been suggested by Saunders [43] (see also [21]), analyzed by Stewart [49] and implemented as Algorithm CHDD in LINPACK [15]. A more computationally efficient scheme for downdating R to D based upon hyperbolic rather than trigonometric rotations can be given. The scheme was originally suggested by Golub [22] for least squares downdating and its numerical properties have been investigated in [2] and [7].

For the purpose of reviewing the use of hyperbolic rotations for least squares downdating, we say that a matrix H is *pseudo-orthogonal* if

$$H^{\mathrm{T}}SH = S$$

for some signature matrix $S = \mathrm{diag}(\pm 1)$. A special symmetric 2×2 pseudo-orthogonal matrix H can be written as

$$H = \begin{bmatrix} \cosh\theta & -\sinh\theta \\ -\sinh\theta & \cosh\theta \end{bmatrix}$$

for any real θ, and corresponds to a 2×2 *hyperbolic plane rotation*. Here $H^{\mathrm{T}}SH = S$ with $S = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$. It is well known (see [2,3,7,9,13,42]) that hyperbolic rotations can be used to zero selected components of a vector. A key observation is that a sequence of hyperbolic rotations can be found, resulting in a pseudo-orthogonal matrix H with respect to $S = \mathrm{diag}(I, -1)$ so that

$$H\begin{bmatrix} R \\ z^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D \\ 0^{\mathrm{T}} \end{bmatrix}, \qquad H\begin{bmatrix} s_1 \\ \eta \end{bmatrix} = \begin{bmatrix} s_1' \\ \eta' \end{bmatrix}$$

where $H^{\mathrm{T}}SH = S$, and the downdated Cholesky factor D satisfies

$$D^{\mathrm{T}}D = R^{\mathrm{T}}R - zz^{\mathrm{T}}.$$

As with the updating calculations using orthogonal rotations, the downdated vector w' is now the solution to a triangular system, in this case

$$Dw' = s_1'.$$

The hyberbolic downdating scheme just described has the same computational efficiency as the orthogonal updating scheme. The hyperbolic scheme requires $\frac{1}{2}n^2 + \mathrm{O}(n)$ fewer multiplications than does the Algorithm CHDD in LINPACK. Moreover, recent analyses [2,7] have shown that downdating algorithms based upon hyperbolic rotations, when implemented appropriately, provide accuracy comparable to that of CHDD. Consequently, we will consider only the use of hyperbolic transformations for downdating least squares problems in this paper.

It should also be noted that the number of multiplications required to compute D from R can be reduced to $\frac{3}{2}n^2 + \mathrm{O}(n)$ multiplications (or even $n^2 + \mathrm{O}(n)$ multiplications if the $L\Lambda L^{\mathrm{T}}$ factorization is used). Such schemes can be derived in terms of the formation and Cholesky factorization of

$$I - bb^{\mathrm{T}}, \quad b = R^{-\mathrm{T}}z,$$

and are based upon the observation that

$$D^{\mathrm{T}}D = R^{\mathrm{T}}R = zz^{\mathrm{T}} = R^{\mathrm{T}}(I - bb^{\mathrm{T}})R.$$

But as mentioned earlier for updating, such alternate approaches generally can be expected to achieve less accuracy and less amenable to parallel implementation [2,13,41]. In particular, schemes based upon the $L\Lambda L^{\mathrm{T}}$ again cause additional numerical difficulties and will not be considered in this paper.

Each of the approaches described thus far to computing modified least squares solutions w' to w involve updating or downdating the Cholesky factor R of $X^{\mathrm{T}}X$. One purpose of this paper is to describe precisely how corresponding orthogonal or hyperbolic transformations can be used to update or downdate the inverse, R^{-1}, of R instead of R itself (the orthogonal case is described by Morf and Kailath [39]). In many signal processing applications, an inverse orthogonal factorization of the Toeplitz autocorrelation matrix T results in the computation of R^{-1} rather than R (see e.g., [12,9]). The advantages of working with the inverse Cholesky factor are clear. Instead of backsolving a triangular system, a triangular matrix is multiplied by a vector, an extremely important consideration on modern multiprocessor architectures [12,32,46]. Moreover, one can more readily recover the covariances associated with the inverse of the normal equations matrix $(X^{\mathrm{T}}X)^{-1} = (R^{\mathrm{T}}R)^{-1} = R^{-1}R^{-\mathrm{T}}$. These covariances are important in signal processing (see e.g., [5,40]) as well as in many other least squares applications, such as geodetic computations [23,24].

Sections 2 and 3 of this paper concern updating and downdating the inverse Cholesky factor R^{-1}. In particular, it is described how the same sequence of orthogonal or hyperbolic rotations used to compute the updated factor U and the downdated factor D from R can also be used to compute U^{-1} and D^{-1} from R^{-1}. In this regard, the following lemma connecting orthogonal and pseudo-orthogonal matrices will be quite useful. The lemma is stated in some generality and may be of independent interest. It is well known for the special case of $n = 2$ (cf., [13]). Similar results find useful applications in network and systems theory [50].

Lemma 1. *Let Q denote an $n \times n$ matrix $(n > 1)$ partitioned as*

$$Q = \begin{bmatrix} E & F \\ G^{\mathrm{T}} & K \end{bmatrix} \tag{1.7}$$

where E is $p \times p$ and both E and K are nonsingular, and set

$$H = \begin{bmatrix} E^{-\mathrm{T}} & FK^{-1} \\ -K^{-1}G^{\mathrm{T}} & K^{-1} \end{bmatrix}. \tag{1.8}$$

Then Q is orthogonal $(Q^{\mathrm{T}}Q = I)$ if and only if H is pseudo-orthogonal $(H^{\mathrm{T}}SH = S)$ with respect to

$$S = \begin{bmatrix} I_p & 0 \\ 0 & -I_{n-p} \end{bmatrix}.$$

Moreover, letting X and Y denote matrices of dimensions $p \times k$ and $(n-p) \times k$, respectively,

$$Q\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X' \\ Y' \end{bmatrix} \tag{1.9}$$

if and only if

$$H\begin{bmatrix} X \\ Y' \end{bmatrix} = \begin{bmatrix} X' \\ Y \end{bmatrix}. \tag{1.10}$$

Proof. The forward direction of the proof can essentially be found in [41], as Theorem 1.3-3, and consequently only an outline will be given here. Assuming that Q is orthogonal, then it can be shown that $H^TSH = S$ from the equations $Q^TQ = QQ^T = I$. For example, $E^{-1}FK^{-1} + GK^{-T}K^{-1} = 0$ and $K^{-T}F^TE^{-T} + K^{-T}K^{-1}G^T = 0$ can be established using $EG + FK^T = 0$. By using the latter equation as well as $EE^T + FF^T = I_p$, one derives $E^{-1}E^{-T}K^{-1}G = I_p$, and $K^{-T}F^TFK^{-1} - K^{-T}K^{-1} = -I_{n-p}$ follows from the equation $F^TF + K^TK = I_{n-p}$.

Next it is shown that if Q is orthogonal and (1.9) holds, then (1.10) follows for the pseudo-orthogonal matrix H. From (1.9) one has $EX + FY = X'$ and $G^TX + KY = Y'$. From the second equation we have $Y = -K^{-1}G^TX + K^{-1}Y'$ which verifies the second equation of (1.10). Substituting into the first equation of (1.9) and simplifying yields $(E - FK^{-1}G^T)X + FK^{-1}Y' = Y$. It thus remains to show that $E - FK^{-1}G^T = E^{-T}$. But this equation follows using the equations from $Q^TQ = I$ and so the forward direction of the lemma holds.

The backward direction of the proof can be established in a similar manner. □

The previous lemma will be used in Sections 2 and 3 to discuss schemes for updating and downdating the inverse Cholesky factor, R^{-1}. Section 4 contains a review of schemes for least squares updating and downdating without triangular solves, along with some comments on parallel implementations.

2. Updating R^{-1}

In this section we review the work of Morf and Kailath [39] on how the same sequence of orthogonal transformations used to update the Cholesky factor R to U can be used to update R^{-1} to U^{-1}. It follows that if Q is an orthogonal matrix such that

$$Q\begin{bmatrix} R \\ y^T \end{bmatrix} = \begin{bmatrix} U \\ 0^T \end{bmatrix}, \tag{2.1}$$

then in fact

$$Q\begin{bmatrix} R^{-T} \\ 0^T \end{bmatrix} = \begin{bmatrix} U^{-T} \\ u^T \end{bmatrix}, \tag{2.2}$$

where y corresponds to the observation being added. It turns out that the vector u in (2.2) is related to the Kalman gain vector [1,5,33] arising in signal processing applications where such schemes are called square root methods.

As before, R is the Cholesky factor of X^TX, y^T is the row vector being appended to X, forming $X' = [{X \atop y^T}]$, and U is the Cholesky factor of X'^TX', i.e.,

$$U^TU = X'^TX' = X^TX + yy^T = R^TR + yy^T.$$

The following general lemma gives a useful expression for the Cholesky factor A of $I + aa^T$, where $a = R^{-T}y$ in the case of interest, along with an expression for A^{-1}. It can also be found (in a slightly different formulation) in [21].

Lemma 2. *Let a denote an n-vector and set*

$$\alpha_0 = 1, \qquad \alpha_k = \sqrt{1 + a_1^2 + \cdots + a_k^2}, \quad k = 1, \ldots, n. \tag{2.3}$$

Then the Cholesky factor A of $I + aa^{\mathrm{T}}$ is given by $A = (a_{ij})$ with

$$a_{ij} = \begin{cases} \dfrac{a_i a_j}{\alpha_{i-1}\alpha_i}, & i < j, \\ \dfrac{\alpha_i}{\alpha_{i-1}}, & i = j. \end{cases} \tag{2.4}$$

Moreover the inverse, A^{-1}, is given by $A^{-1} = (a_{ij}^{(-1)})$ with

$$a_{ij}^{(-1)} = \begin{cases} -\dfrac{a_i a_j}{\alpha_{j-1}\alpha_j}, & i < j, \\ \dfrac{\alpha_{i-1}}{\alpha_i}, & i = j. \end{cases} \tag{2.5}$$

Proof. The proof proceeds by direct verification that $A^{\mathrm{T}}A = I + aa^{\mathrm{T}}$ and that $A^{-1}A = I$, with A and A^{-1} given by (2.4) and (2.5), respectively (see [41]). □

From Lemma 2 it follows immediately that for $a = R^{-\mathrm{T}}y$, the updated Cholesky factor U for R can be expressed as

$$U = AR, \tag{2.6}$$

where A is given explicitly by (2.4). To see this, observe that

$$U^{\mathrm{T}}U = R^{\mathrm{T}}R + yy^{\mathrm{T}} = R^{\mathrm{T}}(I + aa^{\mathrm{T}})R = (AR)^{\mathrm{T}}(AR)$$

and that AR is upper triangular.

There is an inherent connection between orthogonal and pseudo-orthogonal transformations in solving the Cholesky updating problem. We next derive explicit expressions for an orthogonal matrix Q and a pseudo-orthogonal matrix H, with respect to $S = \mathrm{diag}(I_n, -1)$, such that

$$Q\begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U \\ 0^{\mathrm{T}} \end{bmatrix} \tag{2.7}$$

and

$$H\begin{bmatrix} R \\ 0^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U \\ y^{\mathrm{T}} \end{bmatrix}. \tag{2.8}$$

The expressions we derive are based upon Lemma 1 and will lead to formal schemes for updating R^{-1} to U^{-1}.

Theorem 3. *The matrices Q and H in (2.7) and (2.8) are uniquely determined by R and y. Moreover*

$$Q = \begin{bmatrix} A^{-\mathrm{T}} & Aa/\delta^2 \\ -a^{\mathrm{T}}/\delta & 1/\delta \end{bmatrix} \tag{2.9}$$

and

$$H = \begin{bmatrix} A & Aa/\delta \\ a^{\mathrm{T}} & \delta \end{bmatrix} \tag{2.10}$$

where A is given by (2.4) and the vector a is the solution to $R^{\mathrm{T}}a = y$ with $\delta = \sqrt{1 + \| a \|^2}$.

Proof. First, suppose Q in (2.7) is partitioned as

$$Q=\begin{bmatrix} E & f \\ g^{\mathrm{T}} & k \end{bmatrix}$$

where E is $n \times n$, the same dimension as R. Then from

$$\begin{bmatrix} E & f \\ g^{\mathrm{T}} & k \end{bmatrix}\begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix}=\begin{bmatrix} U \\ 0^{\mathrm{T}} \end{bmatrix}, \tag{2.11}$$

$y^{\mathrm{T}}=a^{\mathrm{T}}R$, and $U=AR$ we have

$$AR=U=ER+fy^{\mathrm{T}}=ER+fa^{\mathrm{T}}R,$$

and thus

$$A=E+fa^{\mathrm{T}}. \tag{2.12a}$$

On the other hand, from (2.11) we have $g^{\mathrm{T}}R=-ky^{\mathrm{T}}$, therefore

$$g^{\mathrm{T}}=-ky^{\mathrm{T}}R^{-1}=-ka^{\mathrm{T}}. \tag{2.12b}$$

Next, from $Q^{\mathrm{T}}Q=I$ and (2.12b) it follows that $f^{\mathrm{T}}E=-kg^{\mathrm{T}}=k^2a^{\mathrm{T}}$. Multiplying (2.12a) on the left by f^{T} yields

$$f^{\mathrm{T}}A=f^{\mathrm{T}}E+f^{\mathrm{T}}fa^{\mathrm{T}}=k^2a^{\mathrm{T}}+f^{\mathrm{T}}fa^{\mathrm{T}}=\left(k^2+f^{\mathrm{T}}f\right)a^{\mathrm{T}}=a^{\mathrm{T}}.$$

But

$$a^{\mathrm{T}}=\frac{a^{\mathrm{T}}(I+aa^{\mathrm{T}})}{1+a^{\mathrm{T}}a}=\frac{a^{\mathrm{T}}A^{\mathrm{T}}A}{1+\|a\|^2},$$

and consequently

$$f^{\mathrm{T}}=a^{\mathrm{T}}A^{-1}=\frac{a^{\mathrm{T}}A^{\mathrm{T}}}{1+\|a\|^2}$$

so that

$$f=Aa/\delta^2. \tag{2.13}$$

Next, observe from $k^2+f^{\mathrm{T}}f=1$ that

$$k^2=1-\frac{a^{\mathrm{T}}a+\left(a^{\mathrm{T}}a\right)^2}{\delta^4}=\frac{1}{\delta^2}$$

so that

$$k=1/\delta. \tag{2.14}$$

Consequently from (2.12b)

$$g^{\mathrm{T}}=-ky^{\mathrm{T}}R^{-1}=-ka^{\mathrm{T}}=-a^{\mathrm{T}}/\delta. \tag{2.15}$$

Then from (2.12),

$$E=A-fa^{\mathrm{T}}=A\left(I-aa^{\mathrm{T}}/\delta^2\right)=A\left(A^{\mathrm{T}}A\right)^{-1}$$

so that

$$E=A^{-\mathrm{T}}. \tag{2.16}$$

Equations (2.13)–(2.16) now establish that Q is uniquely (if we choose $k>0$) determined by R and y and is given by (2.9). The expression for H now follows immediately from Lemma 1. □

It now follows that the same sequence of orthogonal transformations used to update R to U can be used to update R^{-1} to U^{-1}.

Theorem 4. (See also Morf and Kailath [39]). *Let R denote an $n\times n$ nonsingular upper triangular matrix and let y denote an n-vector. If Q denotes the product of a sequence of plane rotations used to solve the updating problem for R and y^{T}, i.e. if*

$$Q\begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U \\ 0^{\mathrm{T}} \end{bmatrix} \tag{2.17}$$

where U is upper triangular, then

$$Q\begin{bmatrix} R^{-\mathrm{T}} \\ 0^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U^{-\mathrm{T}} \\ u^{\mathrm{T}} \end{bmatrix} \tag{2.18}$$

where u is given by

$$u = -R^{-1}a/\delta \tag{2.19}$$

with $a = R^{-\mathrm{T}}y$ and $\delta = \sqrt{1+\|a\|^2}$.

The vector u in (2.19), which is obtained as a by-product of updating the inverse Cholesky factor, has a special significance in certain signal processing applications. In particular, a scaled form of u arises in the *Kalman sequential filter* which has been established as a fundamental tool in the analysis and solution of linear estimation problems (see, e.g. [1,5,30,33]). Consider the least squares problem (1.2) and suppose the observation

$$y^{\mathrm{T}}w = \sigma \tag{2.20}$$

is being added. Then the vector

$$K = \frac{(X^{\mathrm{T}}X)^{-1}y}{1+y^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}y} \tag{2.21}$$

is called the *Kalman gain vector* [33]. It weights the predicted residual $\sigma - y^{\mathrm{T}}w$, where w solves the original least squares problem (1.2). The solution w' to the updated problem is given by

$$w' = w + (\sigma - y^{\mathrm{T}}w)K. \tag{2.22}$$

But now observe that since $X^{\mathrm{T}}X = R^{\mathrm{T}}R$ and $y = aR^{\mathrm{T}}$,

$$K = \frac{R^{-1}R^{-T}y}{1+\|R^{-\mathrm{T}}y\|^2} = \frac{R^{-1}a}{\delta^2},$$

where $\delta = \sqrt{1+\|a\|^2}$. Thus the Kalman gain vector K is a scaled form of u given in Theorem 4, namely

$$K = -u/\delta. \tag{2.23}$$

Moreover, it follows now from (2.22) and (2.23) that the updated least squares solution vector w' is given by

$$w' = w - \frac{(\sigma - y^{\mathrm{T}} w)}{\delta} u. \tag{2.24}$$

We next proceed to describe the algorithm for updating the inverse Cholesky factor. The algorithm is based upon the application of orthogonal plane rotations to $\left[\begin{smallmatrix} R^{-\mathrm{T}} \\ 0^{\mathrm{T}} \end{smallmatrix}\right]$ to compute $\left[\begin{smallmatrix} U^{-\mathrm{T}} \\ u^{\mathrm{T}} \end{smallmatrix}\right]$.

In Theorem 4 it was shown that the same sequence of orthogonal plane rotations used to update $\left[\begin{smallmatrix} R \\ y^{\mathrm{T}} \end{smallmatrix}\right]$ to $\left[\begin{smallmatrix} U \\ 0^{\mathrm{T}} \end{smallmatrix}\right]$ can be applied to update $\left[\begin{smallmatrix} R^{-\mathrm{T}} \\ 0^{\mathrm{T}} \end{smallmatrix}\right]$ to $\left[\begin{smallmatrix} U^{-\mathrm{T}} \\ u^{\mathrm{T}} \end{smallmatrix}\right]$. However, the upper triangular matrix R is generally not available so that in constructing the rotation parameters the data in R cannot be used. However, the vector $a = R^{-\mathrm{T}} y$ contains all the information needed to compute the necessary rotation parameters.

We consider Givens rotation matrices of the form

$$G(i, j) = \begin{bmatrix} I & \vdots & & \vdots & \\ \cdots & c & & -s & \cdots \\ & & & \vdots & \\ \cdots & s & \cdots & c & \cdots \\ & \vdots & & \vdots & I \end{bmatrix}$$

where the c and s entries are in the (i, i)th, (i, j)th, (j, i)th and (j, j)th positions and $c^2 + s^2 = 1$. It is a simple matter to zero a selected entry of a p-dimensional vector x using $G(i, j)$. For example if $x_i \neq 0$, then setting

$$h = \sqrt{x_i^2 + x_j^2}, \qquad c = x_j / h, \qquad s = x_i / h,$$

it follows that the ith component of $G(i, j)x$ is zero.

We consider $(n+1) \times (n+1)$ orthogonal plane rotation matrices defined by

$$Q_k \equiv G(k, n+1) = \begin{bmatrix} I & \cdots & & \cdots \\ \cdots & c_k & \cdots & -s_k \\ & \cdots & I & \cdots \\ \cdots & s_k & \cdots & c_k \end{bmatrix} \tag{2.25}$$

where the c_k and s_k are chosen so that

$$Q_n \cdots Q_2 Q_1 \begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U \\ 0^{\mathrm{T}} \end{bmatrix}. \tag{2.26}$$

The components c_k and s_k are determined explicitly in the following lemma.

Lemma 5. *Let $Q \equiv Q_n \cdots Q_2 Q_1$ be a product of orthogonal plane rotation matrices of the form* (2.26), *chosen so that*

$$Q \begin{bmatrix} -a \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \delta \end{bmatrix} \tag{2.27}$$

where $a = R^{-\mathrm{T}} y$ and $\delta = \sqrt{1 + \| a \|^2}$, as before. Then

$$Q \begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} U \\ 0^{\mathrm{T}} \end{bmatrix}. \tag{2.28}$$

Proof. The proof follows by direct verification. □

We are now ready to state the orthogonal plane rotation algorithm designed to update the lower triangular matrix R^{-T} to U^{-T}.

Algorithm O-IU (Orthogonal–Inverse Updating). Given R^{-1} and y:
Step 1. Form the matrix–vector product

$$a = R^{-T}y.$$

Step 2. For $k = 1, \ldots, n$, determine orthogonal plane rotation matrices Q_k of the form (2.25) so that

$$Q_n \cdots Q_2 Q_1 \begin{bmatrix} -a \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \delta \end{bmatrix}$$

with $\delta = \sqrt{1 + \| a \|^2}$, and apply to $[\begin{smallmatrix} R^{-T} \\ 0^T \end{smallmatrix}]$ forming

$$Q_n \cdots Q_2 Q_1 \begin{bmatrix} R^{-T} \\ 0^T \end{bmatrix} = \begin{bmatrix} U^{-T} \\ u^T \end{bmatrix}.$$

We note the similarity of the general procedure of Algorithm O-IU to an orthogonal downdating algorithm devised by Saunders [43] and analyzed by Stewart [49]. However, in our case the vector a need not be completely determined before Step 2 begins. Thus the computation of the components

$$a_k = (R^{-T})_k y, \quad k = 1, \ldots, n,$$

where $(R^{-T})_k$ denotes the kth row of R^{-T}, can be incorporated into Step 2. Moreover the Givens parameters c_k and s_k for Q_k can be given explicitly as

$$c_k = \alpha_{k-1}/\alpha_k, \qquad s_k = -a_k/\alpha_k,$$

where $\alpha_0 = 1$, $\alpha_k = \sqrt{1 + a_1^2 + \cdots + a_k^2}$, $k = 1, \ldots, n$.

Observe that Step 1 of Algorithm O-IU is ideally suited for implementation on parallel processors. It turns out that Step 2 is also amenable to parallel implementation (see [31,32]). Some comments on such parallel implementations are given in Section 4.

It is easy to see that Algorithm O-IU, based upon orthogonal plane rotations, can be implemented with $\frac{5}{2}n^2 + O(n)$ floating-point multiplications and $\frac{3}{2}n^2 + O(n)$ floating-point additions. If, however, a generally less stable Cholesky-type algorithm based upon the implicit formation and factorization of $I + aa^T$ is used, then the number of multiplications can be reduced by the factor $\frac{1}{2}n^2$. In this scheme the computation of A^{-1} given in (2.5) is combined with the computation of $U^{-T} = A^{-T}R^{-T}$. The Kalman gain vector also results as a by-product of these computations [5].

3. Downdating R^{-1}

The downdating process for the Cholesky factor is similar in certain ways to the updating process and, consequently, we will omit several proofs and comments in this section. However, some of the procedures are different and need explanation. Whereas orthogonal transformation matrices were used in updating, we will be concerned primarily with the case of pseudo-orthogonal hyperbolic transformation matrices in downdating. We show how the same sequence of hyperbolic rotations used to downdate R to D can be used to downdate R^{-1} to D^{-1}, thus extending the results in Section 2 and in [39] to downdating. In particular, if H is pseudo-orthogonal with respect to $S = \operatorname{diag}(I_n, -1)$, and

$$H\begin{bmatrix} R \\ z^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D \\ 0^{\mathrm{T}} \end{bmatrix}, \tag{3.1}$$

then

$$H\begin{bmatrix} R^{-\mathrm{T}} \\ 0^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D^{-\mathrm{T}} \\ v^{\mathrm{T}} \end{bmatrix}, \tag{3.2}$$

where z corresponds to the observation being removed. As in the case of u in (2.2), the vector v in (3.2) can be related to Kalman filtering schemes in certain signal processing applications.

Recall that in downdating the Cholesky factor R of $X^{\mathrm{T}}X$ by removing a row vector z^{T} from X, one seeks the Cholesky factor D of $X'^{\mathrm{T}}X'$, where $X = [{X' \atop z^{\mathrm{T}}}]$. In this case

$$D^{\mathrm{T}}D = X'^{\mathrm{T}}X' = X^{\mathrm{T}}X - zz^{\mathrm{T}} = R^{\mathrm{T}}R - zz^{\mathrm{T}} = R^{\mathrm{T}}(I - bb^{\mathrm{T}})R,$$

with $b = R^{-\mathrm{T}}z$. The following general lemma provides an expression for the Cholesky factor B of $I - bb^{\mathrm{T}}$, along with an expression for B^{-1}.

Lemma 6. *Let b denote an n-vector and set*

$$\beta_0 = 1, \qquad \beta_k = \sqrt{1 - b_1^2 - \cdots - b_k^2}, \quad k = 1, \ldots, n. \tag{3.3}$$

Then the Cholesky factor B of $I - bb^{\mathrm{T}}$ is given by $B = (b_{ij})$ where

$$b_{ij} = \begin{cases} -\dfrac{b_i b_j}{\beta_{i-1}\beta_i}, & i < j, \\[2ex] \dfrac{\beta_i}{\beta_{i-1}}, & i = j. \end{cases} \tag{3.4}$$

Moreover, the inverse, B^{-1}, is given by $B^{-1} = (b_{ij}^{(-1)})$ with

$$b_{ij}^{(-1)} = \begin{cases} \dfrac{b_i b_j}{\beta_i \beta_j}, & i < j, \\[2ex] \dfrac{\beta_{i-1}}{\beta_i}, & i = j. \end{cases} \tag{3.5}$$

As was the case with Lemma 2, the proof of this lemma is easily obtained by direct verification. Also, the downdated Cholesky factor D for R is given by

$$D = BR. \tag{3.6}$$

Pseudo-orthogonal transformations used in solving the Cholesky downdating problem can be related to corresponding orthogonal transformations. In particular, we give expressions for H, pseudo-orthogonal with respect to $S = \operatorname{diag}(I_n, -1)$, and a corresponding orthogonal matrix Q such that

$$H\begin{bmatrix} R \\ z^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D \\ 0^{\mathrm{T}} \end{bmatrix} \tag{3.7}$$

and

$$Q\begin{bmatrix} R \\ 0^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D \\ z^{\mathrm{T}} \end{bmatrix}. \tag{3.8}$$

Theorem 7. *The matrices Q and H in (3.7) and (3.8) are uniquely determined by R and z. Moreover*

$$H = \begin{bmatrix} B^{-\mathrm{T}} & -Bb/\gamma^2 \\ -b^{\mathrm{T}}/\gamma & 1/\gamma \end{bmatrix} \tag{3.9}$$

and

$$Q = \begin{bmatrix} B & -Bb/\gamma \\ b^{\mathrm{T}} & \gamma \end{bmatrix}, \tag{3.10}$$

where B is given by (3.4) and the vector b is the solution to $R^{\mathrm{T}}b = z$, with $\gamma = \sqrt{1 - \|b\|^2}$.

The proof of Theorem 7 is similar to the proof of Theorem 3 for updating. We next relate downdating R to D to downdating R^{-1} to D^{-1}.

Theorem 8. *Let R denote an $n \times n$ nonsingular upper triangular matrix and let z denote an n-vector. If H is pseudo-orthogonal with respect to $S = \operatorname{diag}(I_n, -1)$ and*

$$H\begin{bmatrix} R \\ z^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D \\ 0^{\mathrm{T}} \end{bmatrix}, \tag{3.11}$$

where D is the upper triangular downdated Cholesky factor, then

$$H\begin{bmatrix} R^{-\mathrm{T}} \\ 0^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D^{-\mathrm{T}} \\ v^{\mathrm{T}} \end{bmatrix}, \tag{3.12}$$

where v is given by

$$v = -R^{-1}b/\gamma \tag{3.13}$$

with $b = R^{-\mathrm{T}}z$ and $\gamma = \sqrt{1 - \|b\|^2}$.

The proof here is analogous to the proof of Theorem 4 and is omitted. The vector v in (3.13) is also related to a Kalman "loss" vector $\bar{K}$, given by

$$\bar{K} = \frac{(X^{\mathrm{T}}X)^{-1}z}{1 - z^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}z}. \tag{3.14}$$

(However, we have not encountered the situation of Kalman filtering techniques with downdating in the literature. There an exponential "forgetting factor" is generally used to remove the effect of deleting an observation [1,5,30].)

The vector v in (3.12) is related to $\bar{K}$ in (3.14) by

$$\bar{K} = -v/\gamma.$$

Moreover, if w solves the original least squares problem (1.2), then the solution to the downdated problem where the observation (equation) $z^{\mathrm{T}}w = \eta$ is removed from (1.2) is given by

$$w' = w - \left(\frac{\eta - z^{\mathrm{T}}w}{\gamma}\right)v. \tag{3.15}$$

We now describe some algorithms that can be used to downdate R^{-1} to D^{-1}. The first algorithm involves the use of hyperbolic plane rotations and is based upon an application of Theorem 7.

We consider hyperbolic plane rotation matrices of the form

$$H(i, j) = \begin{bmatrix} I & \vdots & & \vdots & \\ \cdots & c & \cdots & -s & \cdots \\ & \vdots & & \vdots & \\ \cdots & -s & \cdots & c & \cdots \\ & \vdots & & \vdots & I \end{bmatrix} \tag{3.16}$$

where the c and s entries are in the (i, i)th, (i, j)th, (j, i)th and (j, j)th positions and

$$c^2 - s^2 = 1.$$

Here $H(i, j)$ is pseudo-orthogonal with respect to the signature matrix S obtained from I by negating the jth diagonal element.

In contrast to the case for orthogonal plane rotations, matrices of the form (3.16) cannot always be used to zero a selected non-zero component of a vector x. However, if $|x_i| > |x_j|$, then letting

$$h = \sqrt{x_i^2 - x_j^2}, \qquad c = x_i/h, \qquad s = x_j/h,$$

it follows that $H(i, j)$ is pseudo-orthogonal and that the jth component of $H(i, j)x$ is zero. Hyperbolic plane rotations can thus be constructed according to the following scheme, whenever $|x_i| > |x_j|$: set

$$t = x_j/x_i, \qquad c = 1/\sqrt{1 - t^2}, \qquad s = ct.$$

The application of hyperbolic plane rotations to reduce $[{R \atop z^{\mathrm{T}}}]$ to $[{D \atop 0^{\mathrm{T}}}]$ was evidently first discussed by Golub [22]. Error analyses have been provided for these methods. Alexander et al. [2,3] have shown that the use of hyperbolic plane rotations for Cholesky downdating is forward (weakly) stable, in the sense that acceptable results can always be expected when the downdating problem is not too ill-conditioned. Bojanczyk et al. [7] have established a mixed stability result for Golub's modified algorithm.

We consider $(n+1) \times (n+1)$ hyperbolic plane rotation matrices defined by

$$H_k \equiv G(k, n+1) = \begin{bmatrix} I & \cdots & & \cdots \\ \cdots & c_k & \cdots & -s_k \\ & \cdots & I & \cdots \\ \cdots & -s_k & \cdots & c_k \end{bmatrix} \tag{3.17}$$

which are pseudo-orthogonal with respect to $S = \text{diag}(I_n, -1)$. Here the c_k and s_k are chosen so that the application of the H_k reduces $[{}^{R}_{z^{\mathrm{T}}}]$ to $[{}^{D}_{0^{\mathrm{T}}}]$. As was the case for orthogonal plane rotations for the updating problem, the c_k and s_k can be given explicitly without having R available.

Lemma 9. *Let $H \equiv H_n \cdots H_2 H_1$ be a product of hyperbolic plane rotation matrices of the form (3.17) chosen so that*

$$H\begin{bmatrix} b \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \gamma \end{bmatrix} \tag{3.18}$$

where $b = R^{-\mathrm{T}}z$ and $\gamma = \sqrt{1 - \|b\|^2}$. Then

$$H\begin{bmatrix} R \\ y^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D \\ 0^{\mathrm{T}} \end{bmatrix}.$$

The hyperbolic plane rotation algorithm for downdating the lower triangular matrix $R^{-\mathrm{T}}$ to $U^{-\mathrm{T}}$ can now be given.

Algorithm H-ID (Hyperbolic–Inverse Downdating). Given R^{-1} and z:
Step 1. Form the matrix–vector product $b = R^{-\mathrm{T}}y$.
Step 2. For $k = 1, \ldots, n$, determine hyperbolic plane rotation matrices H_k of the form (3.17) so that

$$H_n \cdots H_2 H_1 \begin{bmatrix} b \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \gamma \end{bmatrix}$$

with $\gamma = \sqrt{1 - \|b\|^2}$ and apply to $[{}^{R^{-\mathrm{T}}}_{0^{\mathrm{T}}}]$ forming

$$H_n \cdots H_2 H_1 \begin{bmatrix} R^{-\mathrm{T}} \\ 0^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} D^{-\mathrm{T}} \\ v^{\mathrm{T}} \end{bmatrix}.$$

As with Algorithm O-IU, for orthogonal updating, the matrix–vector multiplication in Step 1 is suited for parallel implementation and need not be completed before Step 2 begins. The hyperbolic rotation parameters c_k and s_k for H_k can be given explicitly as

$$c_k = \beta_{k-1}/\beta_k, \qquad s_k = a_k/\beta_k,$$

where $\beta_0 = 1$, $\beta_k = \sqrt{1 - b_1^2 - \cdots - b_k^2}$, $k = 1, \ldots, n$.

It should be noted that for better stability properties, a modified algorithm for applying hyperbolic plane rotations (see Algorithm 2′ in [2]) should be used in Step 2. An indication of the possible superiority of Algorithm 2′ in [2] over the standard implementation of hyperbolic rotations (Algorithm 2 in [2]) is given in [7].

Further, a perfectly parallel version of Step 2 can be given, and this will be discussed in Section 4. Algorithm H-ID can be implemented with $\frac{5}{2}n^2 + O(n)$ multiplications and $\frac{3}{2}n^2 + O(n)$ additions, but many of these operations can be performed in parallel (see Section 4).

As was the case for updating, the number of multiplications required for downdating R^{-1} can be reduced by a factor $\frac{1}{2}n^2$ if a generally less stable Cholesky-type algorithm is used. Here, the scheme is based upon the implicit formation and Cholesky factorization of $I - bb^{\mathrm{T}}$, $b = R^{-\mathrm{T}}z$. The computation of B^{-1} in (3.5) is combined with the computation of $D^{-\mathrm{T}} = B^{-\mathrm{T}}R^{-\mathrm{T}}$. The Kalman "loss" vector in (3.14) also results as a by-product of these computations. Such an algorithm can be derived in a manner similar to the derivation of a corresponding algorithm for downdating R to D given in [41].

4. Summary and parallel implications

We have attempted to provide some new insights into the processes of updating and downdating least squares computations. One key feature of the approach is that highly serial triangular solves can be avoided entirely by working with the inverse Cholesky factor. This concept has been used extensively in signal processing [5,33,35,39,40,50]. Another key feature is that the same sequence of orthogonal (hyperbolic) rotations used to update (downdate) the Cholesky factor R can also be used to update (downdate) its inverse, R^{-1}.

We now summarize the least squares updating and downdating schemes and point out some parallel implications and directions for future work. As before, $\min_w \| s - Xw \|$ denotes the original least squares problem. We seek to update w to w' by successively adding or deleting observations, $y^T w = \sigma$ or $z^T w = \eta$, respectively. In windowed, recursive least squares computations, observations are alternately added and then deleted for stability reasons [2,42]. A general summary of the combined methods is given next, beginning with the updating algorithm.

Algorithm LS-IU (Least Squares–Inverse Updating). Given the current least squares estimator vector w, the current inverse Cholesky factor R^{-1} and the observation $y^T w = \sigma$ being added, the algorithm computes the updated least squares estimator w' and the updated inverse Cholesky factor U^{-1}.

Step 1. Form the matrix–vector product $a = R^{-T} y$.

Step 2. Apply Step 2 of Algorithm O-IU to compute

$$Q \begin{bmatrix} -a & R^{-T} \\ 1 & 0^T \end{bmatrix} = \begin{bmatrix} 0 & U^{-T} \\ \delta & u^T \end{bmatrix}, \quad Q^T Q = I. \tag{4.1}$$

where $\delta = \sqrt{1 + \| a \|^2}$.

Step 3. Set

$$w := w - \frac{(\sigma - y^T w)}{\delta} u, \qquad R^{-1} := U^{-1}. \tag{4.2}$$

The downdating algorithm is given next. Again, the hyperbolic plane rotations should be applied according to the scheme in [22,34], as discussed in [2,7].

Algorithm LS-ID (Least Squares–Inverse Downdating). Given the current least squares estimator vector w, the current inverse Cholesky factor R^{-1} and the observation $z^T w = \eta$ being deleted, the algorithm computes the downdated least squares estimator w' and the downdated inverse Cholesky factor U^{-1}.

Step 1. Form the matrix–vector product $b = R^{-T} z$.

Step 2. Apply Step 2 of Algorithm H-ID to compute

$$H \begin{bmatrix} b & R^{-T} \\ 1 & 0^T \end{bmatrix} = \begin{bmatrix} 0 & D^{-T} \\ \gamma & v^T \end{bmatrix}, \quad H^T S H = S, \tag{4.3}$$

with $S = \operatorname{diag}(I_n, -1)$ and where

$$\gamma = \sqrt{1 - \| b \|^2}.$$

Step 3. Set

$$w := w - \left(\frac{\eta - z^{\mathrm{T}} w}{\gamma} \right) v, \qquad R^{-1} := D^{-1}.$$

Each of these two algorithms requires up to $\frac{5}{2}n^2 + \mathrm{O}(n)$ multiplications for implementation, which is just as computationally efficient as the standard serial least squares updating and downdating algorithms implemented in LINPACK [15]. However, our primary purpose here is to develop algorithms for efficient implementation on modern parallel or vector architectures. Significantly, these approaches are completely free of triangular solves.

On conventional serial computers, the solution of triangular systems is often thought of as a generally trivial extra computation (relative to matrix factorization computations). However, on multiprocessor architectures, particularly distributed memory systems, the triangular solution phase of solving systems of linear equations or least squares problems can require a significant portion of the total computation, although some progress has been made recently in efficiently implementing triangular solves on hypercube type architectures [16,36]. In view of the difficulty in efficiently implementing triangular solves on modern multiprocessor systems, the fact that the least squares updating and downdating schemes discussed herein completely avoid such situations is especially significant. Step 1 in these algorithms requires only a matrix–vector multiplication, while in Step 3 the modified least squares estimators are also obtained without the use of triangular solves and is thus highly amenable to parallel implementation.

In addition, Henkel et al. [32] have recently devised, in conjunction with a parallel implementation of the least squares downdating routine CHDD in LINPACK [15], an efficient parallel scheme for computations quite similar to (4.1) and (4.3). These computations can be organized so that no communication between processors is necessary. Work is continuing in efficiently implementing Algorithms LS-IU and LS-ID on distributed memory systems. In particular, Henkel and Plemmons [31] have recently developed an efficient parallel implementation of Algorithm LS-IU on an iPSC/2 hypercube with 64 processors. This 64-mode system is measured to execute the algorithm over 48 times as fast as a single processor when the problem size for the system is fixed, and over 60 times as fast (scaled speedup) when the problem size per processor is fixed.

Acknowledgements

The authors would like to thank Professors P. van Dooren, T. Kailath and G. Verghese for several helpful comments on a preliminary version of this paper presented at the International Conference on Linear Algebra and Applications in Valencia, Spain, September 1987.

References

[1] S.T. Alexander, *Adaptive Signal Processing* (Springer, New York, 1986).

[2] S.T. Alexander, C.-T. Pan and R.J. Plemmons, Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing, *Linear Algebra Appl.* **98** (1988) 3–40.

[3] S.T. Alexander, C.-T. Pan and R.J. Plemmons, Numerical properties of a hyperbolic rotation method for windowed RLS filtering, in: *Proc. IEEE Int. Conf. Acous., Speech and Sig. Proc.*, Dallas (1987) 423–426.

[4] M. Annaratone, E. Arnould, T. Gross, H.T. Kung, M. Lam, O. Menzilcioglu, K. Sarocky and J. Webb, Warp architecture and implementation, in: *Proc. IEEE 13th Int. Symp. on Comp. Arch.* (1986) 346–356.

[5] G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation* (Academic Press, New York, 1977).

[6] A. Björck, Least squares methods, in: P. Ciarlet and J. Lions, Eds., *Handbook of Numerical Analysis, Vol. 1* (North-Holland, Amsterdam, 1989) 465–653.

[7] A.W. Bojanczyk, R.P. Brent, P. van Dooren and F.R. de Hoog, A note on downdating the Cholesky factorization, *SIAM J. Sci. Statist. Comput.* **8** (1987) 201–221.

[8] J.M. Chambers, Regression updating, *J. Amer. Statist. Assoc.* **66** (1971) 744–748.

[9] J. Chun, T. Kailath and H. Lev-Ari, Fast parallel algorithms for QR and triangular factorization, *SIAM J. Sci. Statist. Comput.* **8** (1987) 399–413.

[10] J.M. Cioffi and T. Kailath, Fast transversal recursive least squares filters for adaptive filtering, *IEEE Trans. Acoust. Speech Signal Process.* **32** (1984) 304–338.

[11] J.M. Cioffi and T. Kailath, Windowed fast transversal filters, adaptive algorithms with normalization, *IEEE Trans. Acoust. Speech Signal Process.* **33** (1985) 607–625.

[12] G. Cybenko, Fast Toeplitz orthogonalization using inner products, *SIAM J. Sci. Statist. Comput.* **8** (1987) 734–740.

[13] J.-M. Delosme and I.C.F. Ipsen, Parallel solution of symmetric positive definite systems with hyperbolic rotations, *Linear Algebra Appl.* **77** (1986) 75–112.

[14] J.E. Dennis and R.B. Schnabel, *Quasi-Newton Methods for Nonlinear Problems* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[15] J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, *LINPACK Users Guide* (SIAM Publications, Philadelphia, PA, 1978).

[16] S.C. Eisenstat, M.T. Heath, C.S. Henkel and C.R. Romine, Modified cyclic algorithms for solving triangular systems on distributed-memory multiprocessors, *SIAM J. Sci. Statist. Comput.* **9** (1988) 589–600.

[17] R. Fletcher and M. Powell, On the modification of LDL^T factorizations, *Math. Comp.* **28** (1974) 1067–1087.

[18] D. Foulser and R. Schreiber, The Saxpy Matrix-1: A general-purpose systolic computer, *IEEE Comput. J.* **20** (1987) to appear.

[19] N. Ghosh and S. Mukhergee, A new boundary element method formulation for three dimensional problems in linear elasticity, *Acta Mech.*, to appear.

[20] P.E. Gill, W. Murray and M.A. Saunders, Methods for computing and modifying the LDU factors of a matrix, *Math. Comp.* **29** (1975) 1051–1077.

[21] P.E. Gill, G.H. Golub, W. Murray and M.A. Saunders, Methods for modifying matrix factorizations, *Math. Comp.* **28** (1974) 505–535.

[22] G.H. Golub, Matrix decompositions and statistical calculations, in: R.C. Milton and J.A. Nelder, Eds., *Statistical Computation* (Academic Press, New York, 1969) 365–395.

[23] G.H. Golub and R.J. Plemmons, Large-scale geodetic least squares adjustment by dissection and orthogonal decomposition, *Linear Algebra Appl.* **34** (1980) 3–27.

[24] G.H. Golub, R.J. Plemmons and A. Sameh, Parallel block schemes for large-scale least squares computations, in: *Proc. Workshop on Sci. Appl. and Algebra Design for High Speed Comp.*, Univ. Illinois (1986).

[25] G.H. Golub and C. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, MD, 1983).

[26] W.W. Hager, Updating the inverse of a matrix, preprint, Penn. State Univ., Dept. Math., 1987.

[27] W.W. Hager, Dual techniques for constrained optimizations, *J. Optim. Theory Appl.*, to appear.

[28] H.R. Hashemipour, S. Roy and A.J. Laub, Decentralized structures for parallel Kalman filtering, *IEEE Trans. Automat. Control*, to appear.

[29] H.R. Hashamipour, S. Roy and A.J. Laub, Square-root parallel Kalman filtering using reduced-order local filters, preprint, UCSB, Dept. ECE, 1987.

[30] S. Haykin, *Adaptive Filter Theory* (Prentice-Hall, Englewood Cliffs, NJ, 1986).

[31] C.S. Henkel and R.J. Plemmons, Recursive least squares computations on a hypercube multiprocessor using the covariance factorization, *SIAM J. Sci. Statist. Comput.*, to appear.

[32] C.S. Henkel, M.T. Heath and R.J. Plemmons, Cholesky downdating on the hypercube, in: *Proc. Hypercube 3 Conf.* (ACM, New York, 1988).

[33] R.E. Kalman, New methods in Wiener filtering theory, in: J.L. Bogdanoff and F. Kojin, Eds., *Proc. 1st Symp. Eng. Appl. Random Function Theory Probability* (Wiley, New York, 1963) 270–388.

[34] C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems* (Prentice-Hall, Englewood Cliffs, NJ, 1974).

[35] B. Levy, D. Castanon, G. Verghese and A. Willsky, A scattering framework for decentralized estimation problems, *Automatica* **19** (1983) 373–384.

[36] G. Li and T.F. Coleman, A new method for solving triangular systems on distributed memory message-passing systems, preprint, Cornell Univ., Comp. Sci. Dept., 1987.

[37] W. Miller and C. Wrathall, *Software for Roundoff Analysis of Matrix Algorithms* (Academic Press, New York, 1980).

[38] R.A. Monzingo and T.W. Miller, *Introduction to Adaptive Arrays* (Wiley, New York, 1980).

[39] M. Morf and T. Kailath, Square root algorithms and least squares estimation, *IEEE Trans. Automat. Control* **20** (1975) 487–497.

[40] C. Paige, Covariance matrix representation in linear filtering, *Contemp. Math.* **47** (1985) 309–321.

[41] C.-T. Pan, Hyperbolic rotations for downdating the Cholesky factor with applications to signal processing, Ph.D. Dissertation, NCSU Math. Dept., 1987.

[42] C. Rader and A.O. Steinhardt, Hyperbolic Householder transformations, *SIAM J. Matrix Anal. Appl.* **9** (1988) 269–290.

[43] M.A. Saunders, Large scale linear programming using the Cholesky factorization, Ph.D. Dissertation, Stanford Univ., Comp. Sci. Dept., 1972.

[44] R. Schreiber, Implementation of adaptive array algorithms, *IEEE Trans. Acoust. Speech Signal Process.* **34** (1986) 1038–1045.

[45] R. Schreiber, Systolic linear algebra machines: A survey, preprint, RPI, Comp. Sci. Dept., 1987.

[46] R. Schreiber and W.-P. Tang, On systolic arrays for updating the Cholesky-factorization, *BIT* **26** (1986) 451–466.

[47] H.W. Sorensen, Least squares estimation: From Gauss to Kalman, *IEEE Spectrum* **7** (1970) 63–68.

[48] J. Speiser and C. Van Loan, Beamforming algorithms using the generalized SVD, *Real-Time Sig. Proc. VII, SPIE Proc.* (1984).

[49] G.W. Stewart, The effects of rounding error on an algorithm for downdating a Cholesky factorization, *J. Inst. Math. Appl.* **23** (1979) 203–213.

[50] P. van Dooren, Personal correspondence, 1987.