# Econometric applications of positive rank-one modifications of the symmetric factorization of a positive semi-definite matrix[1]

**Enrique Sentana**

(CEMFI)

Revised: August 1998

[1]This is an extensively revised version of Sentana (1994). The author is grateful to Hans Dewachter, Juan J. Dolado, Prof. Golub and two anonymous referees for their comments. Of course, the usual caveat applies. Address for correspondence: Casado del Alisal 5, 28014 Madrid, Spain (tel.: +34 91 429 0551, fax: +34 91 429 1056, e-mail: sentana@cemfi.es).

**Abstract**

We present an algorithm for updating the symmetric factorization of a positive semi-definite matrix after a positive rank-one modification, which works even if the matrices involved do not have full rank. Recursive least squares and factor analysis provide two important econometric applications. An illustrative simulation shows that it can be potentially very useful in recursive situations.

**Keywords:** Recursive Least Squares, Factor Analysis, Cholesky Decomposition, Multicollinearity.

# 1 Introduction

The symmetric factorization of a positive semi-definite $n \times n$ matrix $A = A_L \cdot A_D \cdot A'_L$, with $A_L$ unit lower triangular and $A_D$ diagonal, is a well known numerical procedure with multiple applications in statistics and econometrics. The "innovation accounting" techniques invented by Sims (1980) provide a good example. Another useful application is as a computational shortcut that avoids unnecessary numerical errors associated with matrix inversions (see e.g. Bauer and Reinsch (1971)). For instance, in the heteroskedastic linear regression model:

$$y = X\delta + u, \quad E(u|X) = 0, \quad V(u|X) = diag(\omega) = \Omega$$

the weighted least squares (WLS) estimator of the $n$ regression coefficients in $\delta$ based on the whole sample of $T$ observations:

$$\hat{\delta}(T) = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}y$$

can be very quickly and accurately estimated in terms of the decomposition of the $n \times n$ matrix $M = X'\Omega^{-1}X$ as the solution of the (unit upper) triangular system of equations:[1]

$$M'_L \cdot \hat{\delta}(T) = M_D^{-1} \cdot q$$

where $q$ is in turn the solution of the (unit lower) triangular system:

$$M_L \cdot q = X'\Omega^{-1}y$$

Such a numerical procedure is implicitly estimating by OLS the transformed homoskedastic regression model with orthogonal covariates

$$y^* = X^*\delta^* + u^*$$

[1] If the design matrix, $X$, does not have full column rank $n$, the WLS estimator can be defined in (infinitely) many ways. One computationally attractive possibility in this framework is to use $M'_L \cdot \hat{\delta}(T) = M_D^+ \cdot q$, where $M_D^+$ is the Moore-Penrose inverse of $M_D$.

where $y^* = \Omega^{-\frac{1}{2}}y$, $u^* = \Omega^{-\frac{1}{2}}u$, $X^* = \Omega^{-\frac{1}{2}}XM_L'^{-1}$ and $\delta^* = M_L'\delta$, so that $q = X^{*\prime}y^*$ and $\hat{\delta}^*(T) = M_D^{-1} \cdot q$, and then transforming back the coefficient estimates.

Another example is provided by the log-likelihood function contribution from an observation on a $n \times 1$ Gaussian random vector $x_t$ with conditional mean $\mu_t$ and conditional covariance matrix $\Sigma_t$, which are functions of some parameter vector $\phi$:

$$\ell_t(x_t, \phi) = -\frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_t| - \frac{1}{2}(x_t - \mu_t)'\Sigma_t^{-1}(x_t - \mu_t)$$

where the evaluation of the determinant of the $n \times n$ covariance matrix $\Sigma_t$ and the quadratic form $(x_t - \mu_t)'\Sigma_t^{-1}(x_t - \mu_t)$ can be very efficiently and safely carried out as:

$$\ell(x_t; \phi) = -\frac{1}{2}\ln|\Sigma_{tD}| - \frac{1}{2}s_t'\Sigma_{tD}^{-1}s_t$$

with $s_t$ solving:

$$\Sigma_{tL} \cdot s_t = x_t - \mu_t$$

In fact, such a numerical procedure is implicitly performing a cross-sectional analogue of the prediction error decomposition, where $s_t$ contains the cross-sectional prediction errors, and $\Sigma_{tD}$ the corresponding variances (see Sentana (1997)).

In both examples, it is neither advisable nor necessary from a numerical point of view to compute the inverse explicitly. And even when the inverse is required, for instance to compute WLS standard errors, it is more appropriate to obtain it as $M_L^{-1\prime} \cdot M_D^{-1} \cdot M_L^{-1}$, where $M_L^{-1}$ is the solution to the special system of triangular linear equations $M_L \cdot M_L^{-1} = I_n$.

It is often the case that the original matrix $A$, whose symmetric decomposition is already available, is modified by a symmetric matrix of rank one to:

$$\bar{A} = A + \alpha \cdot zz' \tag{1}$$

with $\alpha$ a scalar and $z$ a $n \times 1$ vector. This situation commonly arises in numerical optimization procedures (see e.g. Gill et al. (1981)), but as we shall see in section

2, there are important econometric examples, such as recursive least squares and factor analysis, when it is also relevant. Given that each symmetric decomposition involves about $\frac{1}{6}n^3$ multiplications and additions, it would be desirable to exploit the existing $A_L \cdot A_D \cdot A_L$ factorization of $A$ to obtain the $\bar{A}_L \cdot \bar{A}_D \cdot \bar{A}'_L$ factorization of $\bar{A}$ in an efficient manner.

There are several algorithms for obtaining the matrix factorization of $\bar{A}$ from that of $A$ in only $O(n^2)$ multiplications and additions. The best-known ones, though, explicitly or implicitly make the assumption that both $A$ and $\bar{A}$ are *positive definite* (see e.g. Fletcher and Powell (1974), Gill et al. (1974), Gill et al. (1975), or Pan and Plemmons (1989)). Unfortunately, in some common econometric examples discussed in section 2, such an assumption cannot be made. For instance, in some recursive least squares applications, the design matrix is singular over the first part of the sample.

The main purpose of this paper is to make econometricians aware of the fact that there are variations of these algorithms which do not require such an assumption. In this respect, we present a slight modification of method C1 in Gill et al. (1974) for the case of $\alpha > 0$ which remains numerically stable when $A$ (and possibly $\bar{A}$) are positive semi-definite (see the appendix for its algebraic justification, implementation details, and a fully worked out numerical example). This modified algorithm turns out to be equivalent to an extension of the t-method discussed in section 4 of Fletcher and Powell (1974). We successfully apply it to an illustrative econometric example in section 3.

# 2 Econometric Motivation

Recursive (weighted) least squares provides an obvious and increasingly important econometric example of rank-one modifications of positive (semi-) definite matrices. Its recursive nature is derived from the fact that the estimator based on the first $t-1$ observations:

$$\hat{\delta}(t-1) = (\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1} (\sum_{s=1}^{t-1} \omega_s^{-1} x_s y_s)$$

is updated as each subsequent observation is added to the data set. In this context, the $n \times n$ matrix $(X'\Omega^{-1}X) = \sum_{s=1}^{T} \omega_s^{-1} x_s x_s'$ can be obtained by using expression (1) recursively, with $A^{(t)} = \sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s'$, $z = x_t$ and $\alpha = \omega_t^{-1}$.

Standard presentations of recursive least squares (see e.g. Harvey (1981a)), though, are often based on the Sherman-Morrison matrix inversion formula

$$\bar{A}^{-1} = A^{-1} - \frac{A^{-1} z z' A^{-1}}{\alpha^{-1} + z' A^{-1} z} \tag{2}$$

which in this case yields

$$(\sum_{s=1}^{t} \omega_s^{-1} x_s x_s')^{-1} = (\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1} - \frac{(\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1} x_t x_t' (\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1}}{\omega_t + x_t' (\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1} x_t} \tag{3}$$

and

$$\hat{\delta}(t) = \hat{\delta}(t-1) + \frac{(\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1} x_t}{\omega_t + x_t' (\sum_{s=1}^{t-1} \omega_s^{-1} x_s x_s')^{-1} x_t} \cdot [y_t - x_t' \hat{\delta}(t-1)]$$

where $y_t - x_t'\hat{\delta}(t-1)$ is the one-step ahead prediction error, or recursive residual, and the $n \times 1$ vector in front is known as the "Kalman gain". Although the direct evaluation of (3) is not recommended due to the accumulation of rounding errors, the Kalman gain can also be obtained as a by-product of the factorization update (see Pan and Plemmons (1989) for details).

Another example is given by a conditionally heteroskedastic, orthogonal, exact $k$-factor model (see e.g. Sentana (1997)), which assumes that the conditional covariance matrix of the $n \times 1$ random vector $x_t$ takes the form:

$$\Sigma_t = C \Lambda_t C' + \Gamma$$

where $C$ is a $n \times k$ full rank matrix of factor loadings, with $n \geq k$, $\Lambda_t$ a $k \times k$ diagonal matrix of time-varying factor variances, and $\Gamma$ a $n \times n$ diagonal matrix of idiosyncratic variances. Here, the $n \times n$ matrix $\Sigma_t$ can also be obtained iteratively on the basis of expression (1), with $A^{(j)} = \Gamma + \sum_{l=1}^{j-1} \lambda_{lt} c_l c_l'$, $z = c_j$ and $\alpha = \lambda_{jt}$, where $c_j$ is the $j^{th}$ column of $C$.[2,3]

Since only $n^2 + O(n)$ multiplications and additions are involved in each rank-one factorization update based on our proposed algorithm, as opposed to $\frac{1}{6}n^6$ in the direct factorization, it is clear that the updating procedure results in significant computational savings in a regression context, except when the number of regressors is very small, and the same is true in the factor model, except when $k$ is large relative to $n$.

The assumption of positive definiteness, though, cannot always be maintained. For instance, in recursive (weighted) least squares, the current design matrix, $X$, may not have full column rank. Such a situation trivially arises if we want to start the recursions from the very first observation, since $A = 0$ then. Harvey (1981b) suggests starting the recursions with $A = \kappa^{-1} I$, where $\kappa$ is a large number, on the presumption that the effect of the "ridge" parameter $\kappa^{-1}$ on estimates should generally be negligible after $n$ iterations, where $n$ is the number of regressors. However, if the regressors are highly collinear, such a fast convergence may be unwarranted, and the resulting estimator may be closer to a recursive version of the ridge estimator than to the recursive (weighted) least squares one for a quite a few observations.[4]

More importantly in practice, such a situation also arises if the regressors include variables which are perfectly collinear during the initial part of the sample. One such example is a linear regression/time series model with dummy variables

[2] In static factor models, $\lambda_{jt} = \lambda_j$ is usually set to 1 for scaling purposes, in which case $\alpha = 1$. Similarly, in recursive ordinary least squares, $\Omega = I$, so that $\alpha$ is again 1.

[3] A generalization of expression (2) can also be exploited to compute the log-likelihood function (see Sentana (1997) for details). In fact, such a procedure is more efficient than the factorization updates when $k$ is small relative to $n$.

[4] But see Koopman (1997) for exact algebraic expressions when $\kappa \to \infty$, and Sentana (1997) for a modification of (2) when $\bar{A}$ is regular but $A$ singular.

for infrequently occurring events. A related example from the structural breaks literature is given by the linear model:

$$y_t = \alpha + \beta d_t + u_t,\ \ u_t \sim i.i.d.(0, \sigma^2) \tag{4}$$

where $d_t$ is a dummy variable equal to 0 for the first $T_1$ observations, and to 1 afterwards (e.g. a German reunification dummy). The recursive OLS estimates of this model computed with $T_1 + t_2$ observations ($t_2 = 1, \ldots, T - T_1$) will be:

$$\hat{\alpha}(T_1 + t_2) = \frac{1}{T_1} \sum_{s=1}^{T_1} y_s$$

$$\hat{\beta}(T_1 + t_2) = \frac{1}{t_2} \sum_{s=T_1+1}^{T_1+t_2} y_s - \frac{1}{T_1} \sum_{s=1}^{T_1} y_s$$

so that $\hat{\alpha}(T_1+t_2)$ corresponds to the sample mean of the first regime and $\hat{\beta}(T_1+t_2)$ to the difference in means across regimes. Similarly, recursive residuals can be computed as:

$$\tilde{u}_{T_1+t_2} = y_{T_1+t_2} - \hat{\alpha}(T_1 + t_2 - 1) - \hat{\beta}(T_1 + t_2 - 1) d_{T_1+t_2}$$

from which CUSUM-type tests can be obtained to check if further structural breaks are present (see Harvey (1981a)).

But even though the design matrix is singular for $t_1 = 1, 2, \ldots, T_1$, we can still compute recursive residuals for these observations as:

$$\tilde{u}_{t_1} = y_{t_1} - \hat{\alpha}(t_1 - 1) - \hat{\beta}(t_1 - 1) d_{t_1}$$

where

$$\hat{\alpha}(t_1) = \frac{1}{t_1} \sum_{s=1}^{t_1} y_s$$

and $\hat{\beta}(t_1) = 0$ (say). To the best of our knowledge, though, standard econometric packages do not start the recursions until observation $T_1 + 1$, which is particularly disappointing when $T_1/T$ is close to 1. The only possibility is to estimate the restricted model $y_t = \alpha + u_t$ recursively for $t = 1, \ldots, T_1$ first, and then switch to (4). This is clearly impractical in more realistic situations.

Another common situation in practice where the initial matrix $A$ does not have full rank occurs in the evaluation of the likelihood function of an orthogonal exact factor model at a set of parameter values which includes some zero idiosyncratic variances (i.e. Heywood cases), so that $\text{rank}(\Gamma) = N_1 < N$ (see Sentana (1997)).

# 3 An Illustrative Application

We have used the updating method in the appendix to compute recursive least squares estimates of the model in equation (4) from a simulated sample generated as:

$$
\begin{aligned}
y_t &= \gamma_0 + \varepsilon_t \qquad \text{for} \qquad t = 1, \ldots, T_0 \\
y_t &= \gamma_1 + \varepsilon_t \qquad \text{for} \qquad t = T_0 + 1, \ldots, T_1 \\
y_t &= \gamma_2 + \varepsilon_t \qquad \text{for} \qquad t = T_1 + 1, , \ldots, T
\end{aligned}
$$

with $\varepsilon_t \sim i.i.d.\ N(0, 1); T_0 = 100, T_1 = 200, T = 300; \gamma_0 = \gamma_2 = 0$ and $\gamma_1 = 1$. Note that there are two structural breaks in the data generating process (at observations 101 and 201), but the econometrician is only aware of the second.

Figures 1 and 2 show the recursive OLS estimates of the parameters $\alpha$ and $\beta$. The behaviour of $\hat{\alpha}(t)$ from $t = 100$ onwards clearly shows the presence of a structural break "unnoticed" by the econometrician. More formally, the cumulative sum of recursive residuals test in Figure 3 confirms it at the 5% level. Please note that standard regression packages cannot start the recursions until observation 201, and hence have no chance of detecting the first structural break.

# 4 Conclusions

Here we present an algorithm for updating the symmetric factorization of a positive semi-definite matrix after a positive rank-one modification. Recursive least squares and factor analysis provide two potential econometric applications. Importantly, the algorithm works even if the matrices involved do not have full rank.

A numerical experiment shows that it is stable for singular as well as rather ill-conditioned matrices. An illustrative simulation shows that it can be potentially very useful in recursive situations.

Finally, the algorithm presented here could be easily modified to update the Cholesky decomposition of $A = A_C A_C'$, with $A_C$ lower triangular, since $A_C$ can be obtained from the $A_L, A_D$ factors as $A_C = A_L \cdot A_D^{1/2}$.

# References

Bauer, F.L. and Reinsch, C. (1971): "Inversion of positive definite matrices by the Gauss-Jordan method", in Wilkinson, J.H. and Reinsch, C. eds. *Handbook for Automatic Computation II: Linear Algebra*, Springer-Verlag, Berlin.

Fletcher, R. and Powell, M.J.D. (1974): "On the modification of $LDL^T$ factorizations", *Mathematics of Computation*, 28, 1067-1087.

Gill, P.E., Golub, G.H., Murray, W. and Saunders, M.A. (1974): "Methods for modifying matrix factorizations", *Mathematics of Computation*, 28, 505-535.

Gill, P.E., Murray, W. and Saunders, M.A. (1975): "Methods for computing and modifying the $LDV$ factors of a matrix", *Mathematics of Computation*, 29, 1051-1077.

Gill, P.E., Murray, W. and Wright, M.H. (1981): *Practical Optimization*, Academic Press, New York.

Harvey, A.C. (1981a): *The Econometric Analysis of Time Series*, Philip Allan, Oxford.

Harvey, A.C. (1981b): *Time Series Models*, Philip Allan, Oxford.

Koopman S.J. (1997): "Exact initial Kalman filtering and smoothing for nonstationary time series models", *Journal of the American Statistical Association*, forthcoming.

Martin, R.S.; Peters, G. and Wilkinson, J.H. (1971): "Symmetric decomposition of a positive definite matrix", in Wilkinson, J.H. and Reinsch, C. eds. *Handbook for Automatic Computation II: Linear Algebra*, Springer-Verlag, Berlin.

Pan, C.-T. and Plemmons, R.J. (1989): "Least squares modifications with inverse factorizations: parallel implications", *Journal of Computational and Applied Mathematics*, 27, 109-127.

Sentana (1994): "A positive rank-one modification of the symmetric factorization of a positive semi-definite matrix", CEMFI Working Paper 9421.

Sentana (1997): "The likelihood function of a conditionally heteroskedastic factor model", mimeo, CEMFI.

Sims, C.A. (1980): "Macroeconomics and reality", *Econometrica* 48, 540-552.

# Appendix

## Algebraic Justification of the Algorithm

It is well known that given a positive semi-definite symmetric matrix $A$ of dimension $n$, rank $n_1$, and nullity $n_2 = n - n_1$, it is possible to find a permutation matrix $P$ such that $A^* = PAP'$ can be written in the form:

$$A_L^* \cdot A_D^* \cdot A_L^{*\prime} = \begin{pmatrix} A_{L11}^* & 0 \\ A_{L21}^* & A_{L22}^* \end{pmatrix} \cdot \begin{pmatrix} A_{D1}^* & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} A_{L11}^{*\prime} & A_{L21}^{*\prime} \\ 0 & A_{L22}^{*\prime} \end{pmatrix}$$

where $A_{L11}^*$ and $A_{L22}^*$ are unit lower triangular matrices of dimensions $n_1$ and $n_2$ respectively, and $A_{D1}^*$ is a positive diagonal matrix of dimension $n_1$, with $A_{L22}^*$ usually set to $I_{n_2}$ for convenience, but without loss of generality. In fact, $P$ is such that $A_L = P' A_L^* P$ is unit lower triangular (and trivially $A_D = P' A_D^* P$ diagonal), so that $A = A_L \cdot A_D \cdot A_L'$. Furthermore, if $A$ is positive definite, then $P = I$, and $D$ is strictly positive.

Let $a_{Lj} = (a_{L1j}, \ldots, a_{Lnj})'$ denote the $j^{th}$ column of $A_L$, with $a_{Lij} = 0$ for $i < j$ and $a_{Ljj} = 1$, and let $a_{Dj}$ denote the $j^{th}$ diagonal element of $A_D$, so that we can write

$$A = \sum_{j=1}^{n} a_{Dj} a_{Lj} a_{Lj}'$$

Let's introduce the notation $w^{(1)} = z$ and $\alpha^{(1)} = \alpha$. The first step of the algorithm consists in obtaining a nonnegative scalar $\alpha^{(2)}$ and a vector $w^{(2)}$, with $w_1^{(2)} = 0$, such that

$$\bar{A} = \sum_{j=1}^{n} a_{Dj} a_{Lj} a_{Lj}' + \alpha^{(1)} w^{(1)} w^{(1)\prime} = \bar{a}_{D1} \bar{a}_{L1} \bar{a}_{L1}' + \sum_{j=2}^{n} a_{Dj} a_{Lj} a_{Lj}' + \alpha^{(2)} w^{(2)} w^{(2)\prime}$$

or equivalently

$$a_{D1} a_{L1} a_{L1}' + \alpha^{(1)} w^{(1)} w^{(1)\prime} = \bar{a}_{D1} \bar{a}_{L1} \bar{a}_{L1}' + \alpha^{(2)} w^{(2)} w^{(2)\prime} \tag{A1}$$

Equating the first columns of the left and right hand sides yields

$$a_{D1}a_{L1} + \alpha^{(1)}w_1^{(1)}w^{(1)} = \bar{a}_{D1}\bar{a}_{L1}$$

so that

$$\bar{a}_{D1} = a_{D1} + \alpha^{(1)}w_1^{(1)}w_1^{(1)}$$

and

$$\bar{a}_{L1} = \frac{a_{D1}}{\bar{a}_{D1}}a_{L1} + \frac{\alpha^{(1)}w_1^{(1)}}{\bar{a}_{D1}}w^{(1)}$$

if $\bar{a}_{D1} \neq 0$, and $\bar{a}_{L1} = (1, *, \ldots, *)'$ otherwise, where $*$ means indeterminate, although usually set to 0. In this respect, note that since $\alpha^{(1)} \geq 0$, $\bar{a}_{D_1} = 0$ requires both $a_{D_1} = 0$ and $\alpha^{(1)}w_1^{(1)}w_1^{(1)} = 0$, which in turn requires $\alpha^{(1)} = 0$ and/or $w_1^{(1)} = 0$.

Assuming that $\bar{a}_{D1} \neq 0$, then it is straightforward to prove that (A1) will be satisfied for

$$\alpha^{(2)} = \alpha^{(1)} \cdot \frac{a_{D1}}{\bar{a}_{D1}} \geq 0$$

and

$$w^{(2)} = w^{(1)} - w_1^{(1)}a_{L1}$$

Following Gil et al. (1974), though, we compute $\bar{a}_{L1}$ as

$$\bar{a}_{L1} = a_{L1} + \beta_1 w^{(2)}$$

where

$$\beta_1 = \frac{\alpha^{(1)}w_1^{(1)}}{\bar{a}_{D1}}$$

In any case, since the first row and column of the remainder matrix

$$\sum_{j=2}^{n} a_{Dj}a_{Lj}a'_{Lj} + \alpha^{(2)}w^{(2)}w^{(2)\prime}$$

are zero, the problem has been reduced to a similar one of dimension $n-1$, which can be solved iteratively in the same manner.

Apart from the trivial case in which $\alpha^{(1)} = 0$, when no action needs to be taken, the algorithm simplifies considerably if $w_1^{(1)} = 0$, in which case $\bar{a}_{D1} = a_{D1}$, $\bar{a}_{L1} = a_{L1}$, $\alpha^{(2)} = \alpha^{(1)}$ and $w^{(2)} = w^{(1)}$ irrespectively of the value of $a_{D1}$. Hence, we can easily handle those situations in which $\bar{a}_{D1} = 0$. Our other proposed modification stems from the fact that if $a_{D1} = 0$ but $w_1^{(1)} \neq 0$, then the above expressions reduce to

$$\begin{aligned} \bar{a}_{D1} &= \alpha^{(1)} w_1^{(1)} w_1^{(1)} \\ \bar{a}_{L1} &= \frac{1}{w_1^{(1)}} w^{(1)} \end{aligned}$$

and $\alpha^{(2)} = 0$, so that no further modifications are required.

Finally, it can be shown that the vector $p = (p_1, \ldots, p_n)'$, where $p_j = w_j^{(j)}$, is the solution to the (unit lower) triangular system of linear equations $A_L \cdot p = z$, and moreover, that the symmetric factorization of the matrix $\tilde{A} = A_D + \alpha \cdot pp'$ can be obtained with $\tilde{A}_D = \bar{A}_D$, and the subdiagonal elements of the $j^{th}$ column of $\tilde{A}_L$ being $\tilde{a}_{Lrj} = \beta_j p_r$, where $\beta_j = 0$ for $\bar{a}_{Dj} = 0$ (see Gill et al. (1974), Fletcher and Powell (1974) and Pan and Plemmons (1989)).

# Implementation Details

1. Define $\alpha^{(1)} = \alpha, w^{(1)} = z, j = 0$.
2. $j = j + 1$
3. If $w_j^{(j)} \neq 0$:

   a) If $a_{Dj} \neq 0$ compute

      $p_j = w_j^{(j)}$

      $\bar{a}_{Dj} = a_{Dj} + \alpha^{(j)} p_j^2$

      $\beta_j = p_j \alpha^{(j)} / \bar{a}_{Dj}$

      $\alpha^{(j+1)} = a_{Dj} \alpha^{(j)} / \bar{a}_{Dj}$

$$\left.\begin{aligned} w_r^{(j+1)} &= w_r^{(j)} - p_j a_{Lrj} \\ \bar{a}_{Lrj} &= a_{Lrj} + \beta_j w_r^{(j+1)} \end{aligned}\right\} r = j+1, \ldots, n$$

and if $j < n$ go to 2, otherwise stop

b) If $a_{Dj} = 0$ compute

$p_j = w_j^{(j)}$

$\bar{a}_{Dj} = \alpha^{(j)} p_j^2$

$\beta_j = 1/p_j$

$\bar{a}_{Lrj} = \beta_j w_r^{(j)} \quad r = j+1, \ldots, n$

$\bar{a}_{Di} = a_{Di} \quad i = j+1, \ldots, n$

$\bar{a}_{Lri} = a_{Lri} \quad i = j+1, \ldots, n; \; r = i+1, \ldots, n$

and stop

4. Else (i.e. if $w_j^{(j)} = 0$) set

$\bar{a}_{Dj} = a_{Dj}$

$\alpha^{(j+1)} = \alpha^{(j)}$

$$\left.\begin{aligned} w_r^{(j+1)} &= w_r^{(j)} \\ \bar{a}_{Lrj} &= a_{Lrj} \end{aligned}\right\} r = j+1, \ldots, n$$

and if $j < n$ go to 2, otherwise stop.

If $a_{Dj} > 0 \; \forall j$ this procedure is identical to Gill et al. (1974) method C1, and therefore entails the same number of operations (i.e. $n^2 + O(n)$ multiplications and additions). On the other hand, if at least one diagonal element of $A_D$ is 0, the computational burden is generally reduced (e.g. if $a_{D1} = 0$ but $z_1 \neq 0$ the number of operations is $O(n)$ only; see also Fletcher and Powell (1974)). Strictly speaking, method C1 could cope with a single $a_{Dj} = 0$ provided that $w_j^{(j)} \neq 0$, but it would involve unnecessary computations.

# A Numerical Example

All existing updating methods are numerically stable when $A$ (and hence $\bar{A}$) are well-conditioned positive definite matrices. Similarly, the method presented above works by hand for very simple examples of singular matrices. The crucial question is what its actual performance and numerical stability are when the matrices involved are ill-conditioned. Since the above method is equivalent to the method presented by Fletcher and Powell (1974), their general rounding-error analysis applies here as well. We complement their results by means of the following illustrative example. Starting from $A^{(0)} = \gamma I$, with $\gamma \geq 0$ but small, we generated the recursive sequence of matrices $A^{(k)} = A^{(k-1)} + 10^{k-1}\ell\ell'$, with $\ell$ a $n \times 1$ vector of ones. The resulting matrices take the simple form $A^{(k)} = \gamma I + \delta_k \ell\ell'$, where $\delta_k = 111\ldots111$ ($k$ ones), with eigenvalues $\gamma$ ($n-1$ times) and $\gamma + \delta_k(\ell'\ell)$ (once). Therefore, although $A^{(k)}$ remains positive definite for any $\gamma$ strictly positive (however small), it becomes ever more ill-conditioned as $k$ increases.

The factorization of $A^{(0)}$ is trivially $A_L^{(0)} = I, A_D^{(0)} = \gamma I$. The factorization of $A^{(k)}$ for any $\gamma > 0$ can also be found after simple but tedious algebraic manipulations. For the case of $n = 4$, we have:

$$A_L^{(k)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \theta_k & 1 & 0 & 0 \\ \theta_k & \theta_k/(1+\theta_k) & 1 & 0 \\ \theta_k & \theta_k/(1+\theta_k) & \theta_k/(1+\theta_k+\theta_k^2) & 1 \end{bmatrix}$$

$$A_D^{(k)} = diag\left(\delta_k + \gamma, \gamma(1+\theta_k), \gamma\frac{1+\theta_k+\theta_k^2}{1+\theta_k}, \gamma\frac{1+2(1+\theta_k+\theta_k^2)+\theta_k^4}{(1+\theta_k+\theta_k^2)(1+\theta_k)}\right)$$

with $\theta_k = \delta_k/(\delta_k + \gamma)$. As $k \to \infty, \theta_k \to 1$, and the above matrices converge to:

$$A_L^{(\infty)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1/2 & 1 & 0 \\ 1 & 1/2 & 1/3 & 1 \end{bmatrix} \quad A_D^{(\infty)} = diag(\infty, 2\gamma, \frac{3}{2}\gamma, \frac{4}{3}\gamma)$$

We applied a Fortran 77 version of the above algorithm to this problem for several very small values of $\gamma$ ($10^{-25}$, $10^{-50}$, $10^{-75}$, and $10^{-100}$) with $k$ up to 100 (the source code is available from the author on request).

Reassuringly enough, it provided the right answer in all cases. It is worth mentioning that $A^{(k)}$ is so ill-conditioned that a standard factorization algorithm such as Martin, Peters and Wilkinson (1971) found $A^{(1)}$ numerically singular, and thus was not able to find the desired factorization.

In the extreme case of $\gamma = 0$, then $A^{(k)}$ has rank 1 for all $k > 0$, with factorization:

$$A_L^{(k)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & * & 1 & 0 \\ 1 & * & * & 1 \end{bmatrix} \quad A_D^{(k)} = diag(\delta_k, 0, 0, 0)$$

Again, the algorithm worked perfectly fine.

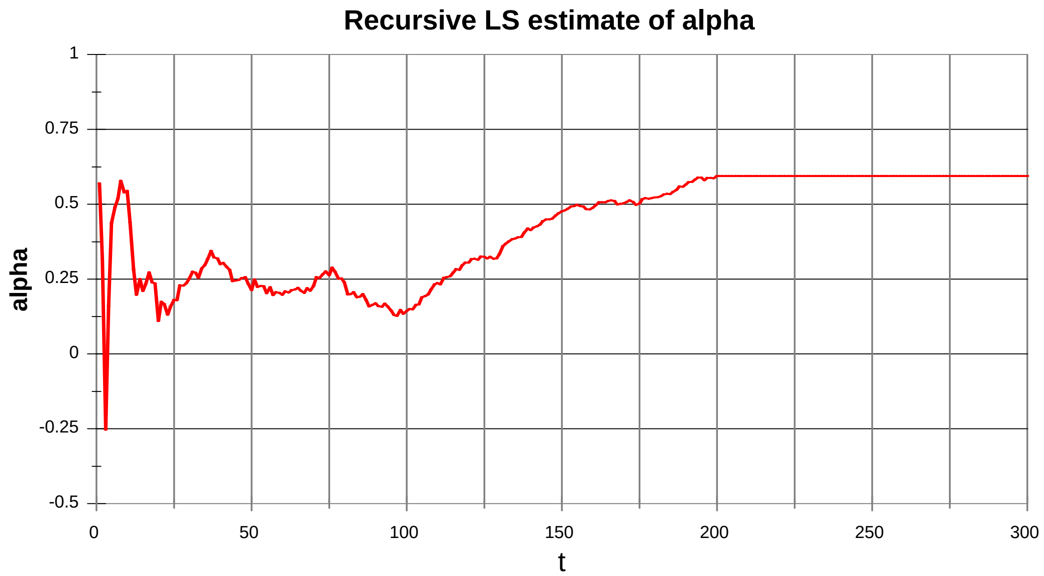


Figure 1:

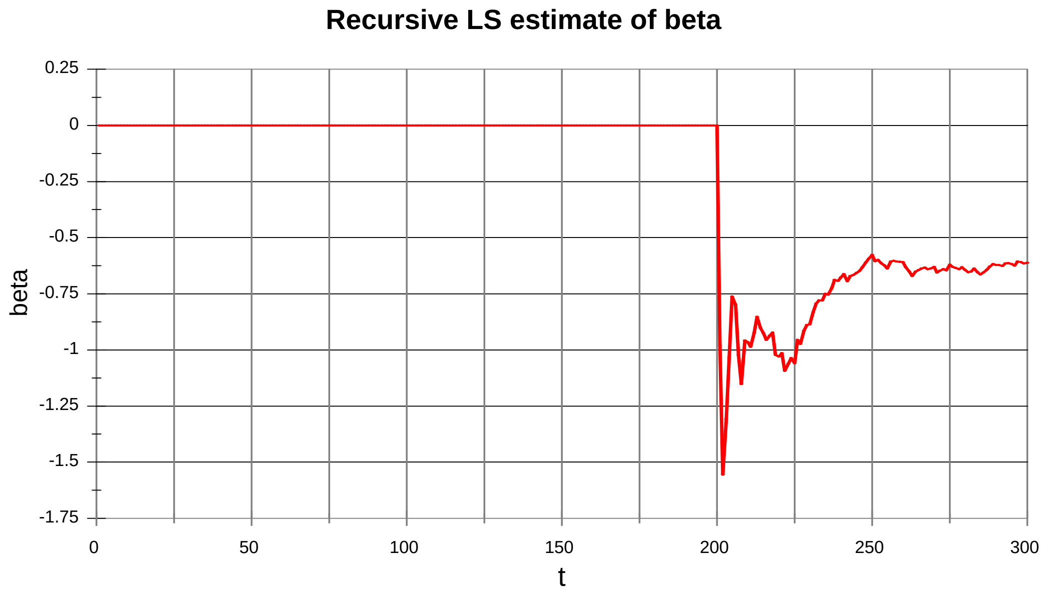


Figure 2:

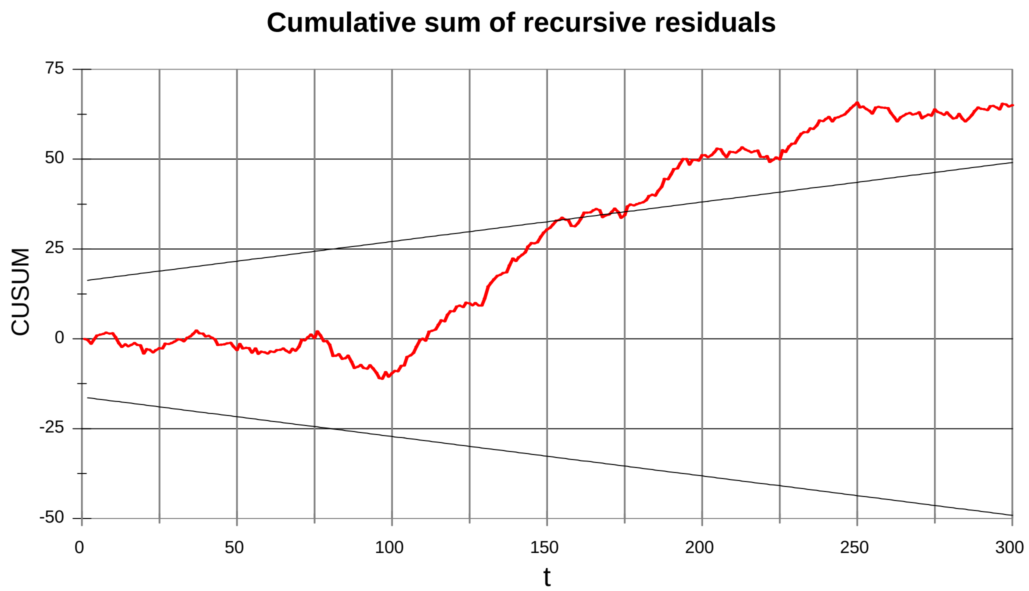
Cumulative sum of recursive residuals
75
50
25
0
-25
-50
CUSUM
0
50
100
150
200
250
300
t

Figure 3: