

A SQUARE-ROOT FIXED-INTERVAL DISCRETE-TIME SMOOTHER

TANIA PRVAN¹ AND M. R. OSBORNE²

(Received 30 September 1987)

Abstract

The square-root fixed-interval discrete-time smoother has been used extensively in discrete recursive estimation since it was first developed by Rauch, Tung and Streibel [10]. Various people, for example Bierman [2], [3], have recognized the inherent instability in employing this kind of smoother in its original form; they have investigated implementing the recursion more stably. Bierman's paper [3] is one such contribution. In this paper we plan to present a more comprehensive development of Bierman's approach, and to show that this algorithm can be implemented more stably as a square-root smoother. Throughout this paper the fixed-interval discrete-time smoother will be referred to as the RTS smoother. Numerical results are given for the usual form of the RTS smoother, Bierman's algorithm and our square-root formulation of his algorithm. These confirm that the square-root formulation is more desirable than Bierman's algorithm, which performs better than the usual implementation of the RTS smoother.

1. Introduction

Reports of computational experience with the RTS smoother suggest that on occasions numerical problems can occur. Bierman [3] identifies these problems as being caused by computations involving the differences of positive definite matrices which can cause loss of significance due to cancellation, and to a lesser extent by the inversion of covariance matrices which can be ill conditioned. In addition, these explicit inversions can be computationally expensive. He suggested an alternative formulation which avoided these problems.

¹Dept. of Pure and Applied Mathematics, Washington State University, Pullman, 99164-2930, WA, USA.

²Dept. of Statistics, Institute of Advanced Studies, Australian National University, G.P.O. Box 4, Canberra, A.C.T. 2601, Australia.

© Copyright Australian Mathematical Society 1988, Serial-fee code 0334-2700/88

In this paper we plan to give a more comprehensive development of Bierman's embedded smoother approach and to show how the equations can be rearranged to exploit the inherent structure. It will be shown that his algorithm can be implemented in a numerically stable way as a square-root smoother. This is desirable because the positive definiteness of the smoothed covariances can be preserved. Numerical results are given for the usual form of the RTS smoother, Bierman's algorithm and our square-root formulation of his algorithm. These confirm the expected advantages.

An appropriate background reference for the filtering and smoothing algorithms to [1]. An accessible account of the Kalman filter is given in [6].

2. Preliminaries

The following signal model is considered

$$\mathbf{x}_j = T_j \mathbf{x}_{j-1} + \mathbf{w}'_j \quad (2.1)$$

$$y_j = \mathbf{h}_j^T \mathbf{x}_j + v_j \quad (2.2)$$

$$j = 1, \dots, n,$$

where $\mathbf{x}_j, \mathbf{w}'_j, \mathbf{h}_j \in \mathfrak{R}^p$, $y_j, v_j \in \mathfrak{R}$ and $T_j \in \mathfrak{R}^{p \times p}$. An initial estimate of the first state \mathbf{x}_1 , say $\hat{\mathbf{x}}_1$, is assumed to be known and likewise an estimate of its covariance S_1 . The following assumptions are made:

$$E(\mathbf{w}'_j) = 0, \quad E(\mathbf{w}'_j \mathbf{w}'_k{}^T) = \delta_{jk} \Omega_j \quad (2.3)$$

$$E(v_j) = 0, \quad E(v_j v_k) = \delta_{jk} \sigma_j^2 \quad (2.4)$$

$$E((\mathbf{x}_1 - \hat{\mathbf{x}}_1) \mathbf{w}'_j{}^T) = 0, \quad E((\mathbf{x}_1 - \hat{\mathbf{x}}_1) v_j) = 0 \quad (2.5)$$

where v_k, \mathbf{w}'_k and \mathbf{x}_1 are uncorrelated.

The Kalman filter recursion for the signal model (2.1) and (2.2) under assumptions (2.3), (2.4) and (2.5) is

$$\mathbf{x}_{j|j-1} = T_j \mathbf{x}_{j-1|j-1} \quad (\text{one step prediction estimate}) \quad (2.6a)$$

$$S_{j|j-1} = T_j S_{j-1|j-1} T_j^T + \Omega \quad (\text{estimate prediction error covariance}) \quad (2.6b)$$

$$\varepsilon_j = y_j - \mathbf{h}_j^T \mathbf{x}_{j|j-1} \quad (\text{innovations residual}) \quad (2.6c)$$

$$d_j = \mathbf{h}_j^T S_{j|j-1} \mathbf{h}_j + \sigma^2 \quad (\text{innovations residual covariance}) \quad (2.6d)$$

$$\mathbf{x}_{j|j} = \mathbf{x}_{j|j-1} + S_{j|j-1} \mathbf{h}_j \varepsilon_j / d_j \quad (\text{estimate update}) \quad (2.6e)$$

and

$$S_{j|j} = S_{j|j-1} - S_{j|j-1} \mathbf{h}_j \mathbf{h}_j^T S_{j|j-1} / d_j \quad (\text{estimate update error covariance}) \quad (2.6f)$$

$$j = 1, \dots, n.$$

The RTS smoother is

$$\mathbf{x}_{j|n} = \mathbf{x}_{j|j} + A_j(\mathbf{x}_{j+1|n} - \mathbf{x}_{j+1|j}) \quad (\text{smoothed estimate}) \quad (2.7a)$$

and

$$S_{j|n} = S_{j|j} + A_j(S_{j+1|n} - S_{j+1|j})A_j^T \quad (\text{smoothed estimate error covariance}), \quad (2.7b)$$

where

$$A_j = S_{j|j}T_{j+1}^T(S_{j+1|j})^{-1} \quad (\text{smoother gain}) \quad (2.7c)$$

$$j = n, \dots, 1.$$

The quantities $\mathbf{x}_{n|n}$ and $S_{n|n}$ which initiate this backward recursion are obtained from the Kalman filter. It is in (2.7b) that the possibility of losing the property of positive definiteness for the smoothed covariance may arise because of the differencing of the positive definite matrices. Our algorithm exploits Bierman's equations for the RTS smoother by arranging it as a square-root smoother which ensures that the smoothed covariances remain positive definite. In some contexts it is desirable to maintain the positive definiteness of the smoothed covariances. Another distinct advantage of using a square-root implementation of the RTS smoother is that it is numerically stable for the covariance calculations. This was supported experimental by numerical results obtained from running the usual form of the RTS smoother, Bierman's formulation of the RTS smoother, and our square-root implementation of Bierman's formulation of the RTS smoother in both single and double precision to obtain estimates of the accuracy of the smoothed estimates and covariances for the three algorithms. The results are given in Section 5 for two examples.

3. Bierman's algorithm

In this section, a modified version of Bierman's approach to obtaining smoothed estimates and their respective covariances will be given.

As a preliminary the following signal model will be considered

$$\mathbf{x}_j = \mathbf{x}_{j-1} + \mathbf{b}w_j \quad j = 1, \dots, n, \quad (3.1)$$

which has no associated observation equation. The vectors \mathbf{x}_j and \mathbf{b} are of dimension p , and w_j is a scalar with zero mean and unit variance. In this special

case the Kalman filter collapses to the following equations

$$\mathbf{x}_{j+1|j} = \mathbf{x}_{j|j} \tag{3.2a}$$

$$\mathbf{x}_{j+1|j+1} = \mathbf{x}\mathbf{x}_{j+1|j} \tag{3.2b}$$

$$S_{j+1|j} = S_{j|j} + \mathbf{b}\mathbf{b}^T \tag{3.2c}$$

$$S_{j+1|j+1} = S_{j+1|j} \tag{3.2d}$$

$$j = 1, \dots, n - 1.$$

The RTS smoother becomes

$$\mathbf{x}_{j|n} = \mathbf{x}_{j|j} + A_j(\mathbf{x}_{j+1|n} - \mathbf{x}_{j+1|j+1}) \tag{3.3a}$$

$$S_{j|n} = S_{j|j} + A_j(S_{j+1|n} - S_{j+1|j+1})A_j^T, \tag{3.3b}$$

where

$$A_j = S_{j|j}(S_{j+1|j+1})^{-1} \tag{3.3c}$$

$$j = n - 1, \dots, 1.$$

Rearranging (3.2c) as an expression for $S_{j|j}$ and substituting this into (3.3c), we have that A_j is given by the rank I matrix

$$A_j = I - \mathbf{b}\mathbf{v}^T,$$

where

$$\mathbf{v} = (S_{j+1|j+1})^{-1}\mathbf{b}.$$

Substituting this expression for A_j into (3.3a) and (3.3b), using (3.2c) and simplifying gives

$$\mathbf{x}_{j|n} = \mathbf{x}_{j+1|n} + \mathbf{b}\mathbf{v}^T(\mathbf{x}_{j+1|j+1} - \mathbf{x}_{j+1|n}) \tag{3.4a}$$

and

$$S_{j|n} = (I - \mathbf{b}\mathbf{v}^T)S_{j+1|n}(I - \mathbf{b}\mathbf{v}^T)^T + (1 - \mathbf{v}^T\mathbf{b})\mathbf{b}\mathbf{b}^T. \tag{3.4b}$$

From (3.2a) and (3.2b) the term $\mathbf{x}_{j+1|j+1}$ is equivalent to $\mathbf{x}_{1|1}$. The formula for the smoothed covariance is a rank II matrix plus a correction term. This formulation is desirable because it lends itself to numerically stable implementation. The model (3.1), and recursion (3.2) and (3.4) are pivotal to Bierman's development of his algorithm.

To make use of (3.4), it is necessary to organise (2.1) into a series of steps, each of which has the form (3.1) with uncorrelated noise terms. Here this is done by rewriting the state equation (2.1) as

$$\mathbf{x}_j = T_j\mathbf{x}_{j-1} + L_j\mathbf{w}_j, \quad \mathbf{w}_j = L_j^{-1}\mathbf{w}'_j, \tag{3.5}$$

where L_j is the Cholesky factor of the covariance Ω_j which for the present is assumed to be positive definite. The quantity L_j^{-1} is never evaluated. It is worthwhile noting here that any square-root factorization of the covariance Ω_j

will suffice to make \mathbf{w}_j have identity covariance which is equivalent to the elements of \mathbf{w}_j being uncorrelated. We choose the Cholesky decomposition because it is readily calculated. Let $w_j(k)$ denote the k th element of the vector \mathbf{w}_j . The state equation (2.1) can be expressed in terms of $w_j(k)$ by considering operating on the columns of

$$L_j = [\mathbf{B}_{j1}, \dots, \mathbf{B}_{jp}],$$

where $\mathbf{B}_{jk} \in \mathfrak{R}^p$, $k = 1, \dots, p$. This gives

$$\mathbf{x}_j = T_j \mathbf{x}_{j-1} + \mathbf{B}_{j1} w_j(1) + \dots + \mathbf{B}_{jp} w_j(p), \tag{3.6}$$

which can be written recursively as

$$\mathbf{X}_j^D = T_j \mathbf{x}_{j-1} \tag{3.7a}$$

$$\mathbf{X}_{ji+1} = \mathbf{X}_{ji} + \mathbf{B}_{ji} w_j(i), \tag{3.7b}$$

where

$$\mathbf{X}_{j1} = \mathbf{X}_j^D. \tag{3.7c}$$

This gives a deterministic update (3.7a) and a noise process update (3.7b). The state equation (2.1) can now be replaced with (3.7a), (3.7b) and (3.7c). The recursion (3.7) considered as a state space formulation does not contain an observation equation.

To obtain the solution for one step prediction the special case Kalman Filter (3.2) is applied to (3.7) which gives

$$\hat{\mathbf{X}}_j^D = T_j \mathbf{x}_{j-1|j-1}, \quad \hat{\mathbf{S}}_j^D = T_j \mathbf{S}_{j-1|j-1} T_j^T \tag{3.8a,b}$$

$$\hat{\mathbf{X}}_{ji+1} = \hat{\mathbf{X}}_{ji}, \quad \hat{\mathbf{S}}_{ji+1} = \hat{\mathbf{S}}_{ji} + \mathbf{B}_{ji} \mathbf{B}_{ji}^T \tag{3.8c,d}$$

$$i = 1, \dots, p,$$

where

$$\hat{\mathbf{X}}_{j1} = \hat{\mathbf{X}}_j^D \tag{3.8e}$$

and

$$\hat{\mathbf{S}}_{j1} = \hat{\mathbf{S}}_j^D. \tag{3.8f}$$

Note that $\hat{\mathbf{X}}_{ji}$ is being used to represent $\mathbf{X}_{ij|j}$ and $\hat{\mathbf{S}}_{ji}$ to represent $\mathbf{S}_{ij|j}$. The recursion (3.8) gives

$$\hat{\mathbf{X}}_{jp+1} = \mathbf{x}_{j|j-1} \tag{3.8g}$$

and

$$\hat{\mathbf{S}}_{jp+1} = \mathbf{S}_{j|j-1}. \tag{3.8h}$$

Upper case bold letters are used to denote embedded recursions. The subscript j indicates that the j th state equation is of interest, while the subscript i indicates that the i th embedded recursion is of interest.

The special case RTS smoother (3.3a) is applied to (3.7) to obtain

$$\mathbf{X}_{jp+1|n} = \mathbf{x}_{j|n} \tag{3.9a}$$

$$\mathbf{X}_{ji|n} = \mathbf{X}_{ji+1|n} + \mathbf{B}_{ji} \mathbf{V}_{ji}^T (\mathbf{x}_{j|j-1} - \mathbf{X}_{ji+1|n}) \tag{3.9b}$$

$$i = p, \dots, 1,$$

$$\mathbf{x}_{j-1|n} = \mathbf{T}_j^{-T} \mathbf{X}_{j1|n} \tag{3.9c}$$

$$j = n, \dots, 1,$$

as the recursion for the smoothed estimate where

$$\mathbf{V}_{ji} = (\widehat{\mathbf{S}}_{ji+1})^{-1} \mathbf{B}_{ji}.$$

To obtain \mathbf{V}_{ji} , $i = 1, \dots, p$, a forward pass of the recursion (3.8) for the covariances is required. If a Cholesky decomposition is applied to $\widehat{\mathbf{S}}_{ji+1}$ then

$$\mathbf{B}_{ij} = \widehat{\mathbf{S}}_{ji+1}^{1/2} (\widehat{\mathbf{S}}_{ji+1}^{1/2})^T \mathbf{V}_{ji}$$

can be solved for the \mathbf{V}_{ji} by a forward and backward substitution, thus avoiding inverting $\widehat{\mathbf{S}}_{ji+1}$. The special case RTS smoother (3.3b) is applied to (3.7) to obtain

$$\mathbf{S}_{jp+1|n} = \mathbf{S}_{j|n} \tag{3.10a}$$

$$\mathbf{S}_{ji|n} = (\mathbf{I} - \mathbf{B}_{ji} \mathbf{V}_{ji}^T) \mathbf{S}_{ji+n|n} (\mathbf{I} - \mathbf{B}_{ji} \mathbf{V}_{ji}^T)^T + \eta_{ji} \mathbf{B}_{ji} \mathbf{B}_{ji}^T \tag{3.10b}$$

$$i = p, \dots, 1,$$

$$\mathbf{S}_{j-1|n} = \mathbf{T}_j \mathbf{S}_{j1|n} \mathbf{T}_j^{-T} \tag{3.10c}$$

$$j = n, \dots, 1,$$

as the backward recursion for the smoothed covariance where

$$\eta_{ji} = (\mathbf{I} - \mathbf{B}_{ji}^T \mathbf{V}_{ji}).$$

To identify (3.9) and (3.10) with the recursions obtained by Bierman, we make a LDL^T decomposition on the state covariances Ω_j where the lower triangular matrix L has unit entries along its principal diagonal and D is a diagonal matrix with elements (q_{j1}, \dots, q_{jp}) , and make the replacement

$$\mathbf{V}_{ji} \leftarrow \lambda_{ji} \mathbf{V}_{ji},$$

where

$$\lambda_{ji} = (\mathbf{I} + \mathbf{V}_{ji}^T \mathbf{B}_{ji} q_{ji})^{-1}.$$

As mentioned earlier, L_j^{-1} is never evaluated. Bierman's algorithm will now be modified to handle Ω_j being positive semi-definite, as follows. Recollect that, since by definition Ω_j is a covariance matrix, it must be symmetric. Thus there exists a nonsingular matrix G_j such that

$$G_j \Omega_j G_j^T = \Delta_j,$$

where Δ_j is a diagonal matrix. Furthermore, Ω_j being positive semi-definite restricts the entries of Δ_j to being non-negative which means that Ω_j possesses the square root $G_j^{-1}\Delta_j^{1/2}$. Since square roots of matrices are determined up to an orthogonal matrix we can convert $G_j^{-1}\Delta_j^{1/2}$ to a lower triangular matrix \bar{L}_j by appropriate multiplications by orthogonal transformations. If Ω_j is not of full rank, neither will \bar{L}_j be of full rank. Let

$$\bar{L}_j = [\mathbf{B}_{j1}, \dots, \mathbf{B}_{jp}].$$

We are now in the position to use the formulation of the Kalman filter and RTS smoother given in this section. The crucial step is that \mathbf{w}_j given in (3.5), even when Ω_j is positive semi definite, still has a covariance which is diagonal, albeit one, which may have some entries that have to be interpreted as approaching infinity (although we never need evaluate it), so the elements of \mathbf{w}_j are still equivalent to being uncorrelated. Bierman did not consider using his algorithm for the state transition covariance Ω_j being positive semi definite; he argued that if this occurred then it served as a warning that the state space formulation needed reformulation.

4. A square-root algorithm

As noted in the preceding section, the $\mathbf{V}_{ji}, i = 1, \dots, p$, are required to apply the recursions (3.9) and (3.10) for given j which in turn requires the quantities $\hat{\mathbf{S}}_{ji+1}, i = 1, \dots, p$. First of all a square-root algorithm for updating $\hat{\mathbf{S}}_{ji}$ will be developed. To do this the recursion (3.8b), (3.8d) and (3.8f) will be exploited.

A square-root Kalman filter is used initially to obtain the square roots of the covariances $(S_{j|j}^{1/2})^T$. The matrix $(S_{j|j}^{1/2})^T$ is kept in upper triangular form. The quantity $(\hat{\mathbf{S}}_{j1})^T$ initiates the forward recursion for $(\hat{\mathbf{S}}_{ji}^{1/2})^T$ (c.f. (3.8f)). The result

$$(\hat{\mathbf{S}}_{j1}^{1/2})^T = (S_{j-1|j-1}^{1/2})^T T_j \tag{4.1}$$

when computed is not necessarily an upper triangular matrix but it is determined up to an orthogonal matrix which can be chosen to restore upper triangular form. Thus (4.1) is premultiplied by orthogonal transformations to restore it to the required form. Using a similar approach orthogonal transformations are used to obtain $(\hat{\mathbf{S}}_{ji+1}^{1/2})^T$ by reducing

$$\begin{bmatrix} (\hat{\mathbf{S}}_{ji}^{1/2})^T \\ \mathbf{B}_{ji}^T \end{bmatrix} \text{ to } \begin{bmatrix} (\hat{\mathbf{S}}_{ji+1}^{1/2})^T \\ 0 \end{bmatrix}, \tag{4.2}$$

for $i = 1, \dots, p$, where $(\hat{\mathbf{S}}_{ji}^{1/2})^T$ and $(\hat{\mathbf{S}}_{ji+1}^{1/2})^T$ are constrained to be upper triangular matrices. The following notation will be useful. Let $H(a_{ii}, a_{ji})$ be the

elementary reflector which uses the (i, i) th element of the matrix A to zero out the (j, i) th element of A . Let A be the matrix being reduced in (4.2); then the reduction of this matrix to an upper triangular matrix can be done efficiently as follows:

$$\prod_{i=1}^p H(a_{ii}, a_{p+1,i})A.$$

More details about this can be found in Golub and Van Loan [5]. Then the following equation

$$\mathbf{B}_{ji} = \widehat{\mathbf{S}}_{ji+1}^{1/2} (\widehat{\mathbf{S}}_{ji+1}^{1/2})^T \mathbf{V}_{ji} \tag{4.3}$$

is solved for \mathbf{V}_{ji} by a backward and forward substitution thus avoiding inverting $\widehat{\mathbf{S}}_{ji}^{1/2}$.

To downdate $(\widehat{\mathbf{S}}_{ji+1}^{1/2})^T$, which is kept as an upper triangular matrix in the embedded recursion for the square root of the smoothed covariance, the square root of the leading term in (3.10b) is first formed. The quantity

$$(\widehat{\mathbf{S}}_{jp+1|n}^{1/2})^T = (S_{jn}^{1/2})^T \tag{4.4}$$

initializes this backward recursion, where $(S_{jn}^{1/2})^T$ is kept in upper triangular form. By inspection of (3.10b), the square root of the first term is

$$(S_{ji+1|n}^{1/2}) - (S_{ji+1|n}^{1/2})^T \mathbf{V}_{ji} \mathbf{B}_{ji}^T.$$

The term $(S_{ji+1|n}^{1/2})^T \mathbf{V}_{ji}$ can be transformed to $\sigma^2 \mathbf{e}_1$ by a series of orthogonal transformations, Q^T , in such a way that $Q^T (S_{ji+1|n}^{1/2})^T$ is upper Hessenberg. This is achieved by choosing Q^T such that

$$Q^T = \prod_{k=p}^2 H(((S_{ji+1|n}^{1/2})^T \mathbf{V}_{ji})_{k-1}, ((S_{ji+1|n}^{1/2})^T \mathbf{V}_{ji})_k).$$

Subtracting $(Q^T (S_{ji+1|n}^{1/2})^T \mathbf{V}_{ji}) \mathbf{B}_{ji}^T$ from $Q^T (S_{ji+1|n}^{1/2})^T$ only alters the first row of $Q^T (S_{ji+1|n}^{1/2})^T$, and so the quantity

$$Q^T (S_{ji+1|n}^{1/2})^T - Q^T (S_{ji+1|n}^{1/2})^T \mathbf{V}_{ji} \mathbf{B}_{ji}^T = \widehat{R} \tag{4.5}$$

is also upper Hessenberg. It only requires $O(n)$ operations to reduce an upper Hessenberg matrix \widehat{R} to an upper triangular matrix R by a series of elementary reflectors P^T ; that is,

$$P^T \widehat{R} = R. \tag{4.6}$$

For more details see Golub and Van Loan [5]. Once the square root of the leading term in (3.10b) is evaluated the square root of the smoothed covariance can be formed by reducing

$$\left[\begin{array}{c} R \\ \sqrt{\eta_{ji}} \mathbf{B}_{ji}^T \end{array} \right] \quad \text{to} \quad \left[\begin{array}{c} (S_{ji|n}^{1/2})^T \\ 0 \end{array} \right] \tag{4.7}$$

by a series of elementary reflectors as outlined for (4.2). After going through (4.5) to (4.7) which implements (3.10b) p times, the quantity $(S_{j1|n}^{1/2})^T$ is obtained. The final step computes

$$(S_{j-1|n}^{1/2})^T = (S_{j1|n}^{1/2})^T \mathbf{T}_j^{-T} \quad (4.8)$$

which is not necessarily in upper triangular form. This must be restored by orthogonal transformation. Note that both this reduction and that required following (4.1) are relatively expensive (of the same order as the internal recursion).

5. Examples

The RTS smoother, Bierman's [3] implementation of the RTS smoother and the square root formulation of the RTS smoother in Section 4 were run in single and double precision to obtain estimates of the precision of the smoothed estimates and covariances for two data sets. The data sets were from Gallant [4] (here $n = 72$) and the sunspot data in Pandit and Wu [9] (here $n = 176$).

The following covariance was used for the transition equation (2.1):

$$\Omega_{lk} = \lambda((2p + 1 - k - l)(p - l)!(p - k)!)^{-1} \\ l = 1, \dots, p, \quad k = 1, \dots, p,$$

for $j = 2, \dots, n$, where the algorithms were used for $p = 1, \dots, 5$. This covariance structure was chosen because we were interested in fitting to the data a polynomial smoothing spline employing a method which made use of the Kalman filter and RTS smoother. For more details refer to Wecker and Ansley [11] and Osborne and Prvan [7], [8] and references contained therein. The results in Table 1 and Table 2 are for values of λ which straddle the optimal λ for the smoothing spline. The smoothed estimates do not vary with the mode of calculation and we believe they reflect the attainable accuracy. Thus they provide a standard by which to judge the covariance calculations. From the results in Table 1 and Table 2 it is apparent that Bierman's algorithm is more accurate than the standard implementation of the RTS smoother, and that the square-root formulation of the RTS smoother is more accurate than either algorithm for $p \geq 3$. Thus our formulation of the RTS smoother has two distinct advantages over its competitors by being more accurate for large p , and by ensuring that the smoothed covariances are positive definite.

TABLE 1. Error estimates for smoothed estimates and smoothed covariances for the Gallant data.

p	λ		RTS Smoother	Bierman	Square root algorithm	
1	0.1	est	0.36×10^{-9}	0.36×10^{-9}	0.36×10^{-9}	
		cov	0.25×10^{-8}	0.36×10^{-9}	0.38×10^{-9}	
	0.01	est	0.88×10^{-9}	0.88×10^{-9}	0.88×10^{-9}	
		cov	0.18×10^{-8}	0.29×10^{-9}	0.43×10^{-9}	
	0.001	est	0.19×10^{-8}	0.19×10^{-9}	0.19×10^{-9}	
		cov	0.93×10^{-9}	0.28×10^{-9}	0.39×10^{-9}	
	0.0001	est	0.70×10^{-9}	0.70×10^{-9}	0.70×10^{-9}	
		cov	0.11×10^{-9}	0.12×10^{-10}	0.27×10^{-10}	
	2	0.1	est	0.16×10^{-10}	0.16×10^{-10}	0.16×10^{-10}
			cov	0.99×10^{-9}	0.10×10^{-8}	0.16×10^{-8}
		0.01	est	0.17×10^{-10}	0.17×10^{-10}	0.17×10^{-10}
			cov	0.93×10^{-9}	0.30×10^{-9}	0.15×10^{-8}
0.001		est	0.18×10^{-10}	0.18×10^{-10}	0.18×10^{-10}	
		cov	0.41×10^{-9}	0.40×10^{-9}	0.19×10^{-8}	
0.0001		est	0.25×10^{-10}	0.25×10^{-10}	0.25×10^{-10}	
		cov	0.63×10^{-9}	0.41×10^{-9}	0.16×10^{-10}	
3		0.1	est	0.13×10^{-10}	0.13×10^{-10}	0.13×10^{-10}
			cov	0.41×10^{-8}	0.30×10^{-8}	0.15×10^{-8}
		0.01	est	0.91×10^{-11}	0.91×10^{-11}	0.91×10^{-11}
			cov	0.17×10^{-8}	0.16×10^{-8}	0.74×10^{-9}
	0.001	est	0.60×10^{-11}	0.60×10^{-11}	0.60×10^{-11}	
		cov	0.16×10^{-8}	0.20×10^{-8}	0.77×10^{-9}	
	0.0001	est	0.26×10^{-12}	0.26×10^{-12}	0.26×10^{-12}	
		cov	0.16×10^{-8}	0.58×10^{-9}	0.42×10^{-9}	
	4	0.1	est	0.24×10^{-10}	0.24×10^{-10}	0.24×10^{-10}
			cov	0.51×10^{-7}	0.70×10^{-8}	0.26×10^{-8}
		0.01	est	0.42×10^{-11}	0.42×10^{-11}	0.42×10^{-11}
			cov	0.25×10^{-7}	0.67×10^{-8}	0.25×10^{-8}
0.001		est	0.30×10^{-11}	0.30×10^{11}	0.30×10^{-11}	
		cov	0.12×10^{-7}	0.46×10^{-8}	0.11×10^{-8}	
0.0001		est	0.56×10^{-12}	0.56×10^{-12}	0.56×10^{-12}	
		cov	0.74×10^{-8}	0.24×10^{-8}	0.13×10^{-8}	
5		0.1	est	0.21×10^{-10}	0.21×10^{-10}	0.21×10^{-10}
			cov	0.78×10^{-6}	0.45×10^{-7}	0.57×10^{-8}
		0.01	est	0.75×10^{-11}	0.75×10^{-11}	0.75×10^{-11}
			cov	0.45×10^{-6}	0.51×10^{-7}	0.35×10^{-8}
	0.001	est	0.94×10^{-12}	0.94×10^{-12}	0.94×10^{-12}	
		cov	0.32×10^{-6}	0.30×10^{-7}	0.31×10^{-8}	
	0.0001	est	0.56×10^{-12}	0.56×10^{-12}	0.56×10^{-12}	
		cov	0.18×10^{-6}	0.13×10^{-7}	0.32×10^{-8}	

TABLE 2. Error estimates for smoothed estimates and smoothed covariances for the sunspot data.

p	λ		RTS Smoother	Bierman	Square root algorithm
1	100	est	0.89×10^{-9}	0.89×10^{-9}	0.89×10^{-9}
		cov	0.16×10^{-7}	0.15×10^{-7}	0.79×10^{-7}
	10	est	0.17×10^{-8}	0.17×10^{-8}	0.17×10^{-8}
		cov	0.79×10^{-8}	0.71×10^{-8}	0.12×10^{-7}
	1	est	0.29×10^{-8}	0.29×10^{-8}	0.29×10^{-8}
		cov	0.11×10^{-7}	0.56×10^{-8}	0.63×10^{-8}
2	100	est	0.73×10^{-8}	0.73×10^{-8}	0.73×10^{-8}
		cov	0.14×10^{-8}	0.18×10^{-9}	0.19×10^{-9}
	10	est	0.32×10^{-8}	0.32×10^{-8}	0.32×10^{-8}
		cov	0.38×10^{-7}	0.31×10^{-7}	0.92×10^{-7}
	1	est	0.14×10^{-8}	0.14×10^{-8}	0.14×10^{-8}
		cov	0.11×10^{-7}	0.56×10^{-8}	0.63×10^{-8}
3	100	est	0.27×10^{-9}	0.27×10^{-9}	0.27×10^{-9}
		cov	0.26×10^{-8}	0.87×10^{-9}	0.13×10^{-8}
	10	est	0.95×10^{-9}	0.95×10^{-9}	0.95×10^{-9}
		cov	0.57×10^{-9}	0.65×10^{-9}	0.13×10^{-8}
	1	est	0.18×10^{-8}	0.18×10^{-8}	0.18×10^{-8}
		cov	0.45×10^{-6}	0.24×10^{-6}	0.43×10^{-7}
4	100	est	0.17×10^{-8}	0.17×10^{-8}	0.17×10^{-7}
		cov	0.87×10^{-7}	0.15×10^{-7}	0.74×10^{-8}
	10	est	0.14×10^{-8}	0.14×10^{-8}	0.14×10^{-8}
		cov	0.12×10^{-7}	0.47×10^{-8}	0.22×10^{-8}
	1	est	0.54×10^{-9}	0.54×10^{-9}	0.54×10^{-9}
		cov	0.29×10^{-8}	0.19×10^{-8}	0.85×10^{-9}
5	100	est	0.27×10^{-9}	0.27×10^{-9}	0.27×10^{-9}
		cov	0.68×10^{-5}	0.97×10^{-6}	0.80×10^{-7}
	10	est	0.22×10^{-8}	0.22×10^{-8}	0.22×10^{-8}
		cov	0.79×10^{-6}	0.11×10^{-6}	0.16×10^{-7}
	1	est	0.20×10^{-9}	0.20×10^{-9}	0.20×10^{-9}
		cov	0.10×10^{-6}	0.18×10^{-7}	0.48×10^{-8}
5	100	est	0.10×10^{-8}	0.10×10^{-8}	0.10×10^{-8}
		cov	0.27×10^{-7}	0.36×10^{-8}	0.16×10^{-8}
	10	est	0.27×10^{-9}	0.27×10^{-9}	0.27×10^{-9}
		cov	0.10×10^{-5}	0.23×10^{-4}	0.14×10^{-6}
	1	est	0.27×10^{-8}	0.27×10^{-8}	0.27×10^{-8}
		cov	0.10×10^{-4}	0.33×10^{-5}	0.22×10^{-7}
0.1	est	0.23×10^{-8}	0.23×10^{-8}	0.23×10^{-8}	
	cov	0.23×10^{-5}	0.21×10^{-6}	0.53×10^{-8}	
0.1	est	0.10×10^{-8}	0.10×10^{-8}	0.10×10^{-8}	
	cov	0.50×10^{-6}	0.23×10^{-7}	0.37×10^{-8}	

6. References

- [1] B. D. O. Anderson and J. B. Moore, *Optimal filtering* (Prentice-Hall, Englewood Cliffs, N. J., 1979).
- [2] G. J. Bierman, *Estimation methods for discrete sequential estimation* (Academic Press, 1977).
- [3] G. J. Bierman, "A new computationally efficient fixed-interval, discrete time smoother", *Automatica* **19** (5) (1983) 503–511.
- [4] T. M. Gerig and A. R. Gallant, "Computing methods for linear models subject to linear constraints", *J. Statist. Comput. Simulation* **3** (1975) 283–296.
- [5] G. H. Golub and C. F. Van Loan, *Matrix computations* (John Hopkins University Press, 1983).
- [6] D. G. Luenberger, *Optimization by vector space methods* (Wiley, 1969).
- [7] M. R. Osborne and Tania Prvan, "On algorithms for generalised smoothing splines", *J. Austral. Math. Soc. Ser. B* **29** (1988) 319–338.
- [8] M. R. Osborne and Tania Prvan, "Smoothness and conditioning in generalised smoothing spline calculations", *J. Austral. Math. Soc. Ser. B* **30** (1988) 43–56.
- [9] S. M. Pandit and S. M. Wu, *Time series and system analysis with applications* (Wiley, 1983).
- [10] H. E. Rauch, F. Tung and C. T. Streibel, "Maximum likelihood estimates of linear dynamic systems", *AIAA J.* **3** (8) (1965) 1445–1450.
- [11] W. E. Wecker and C. F. Ansley, "The signal extraction approach to nonlinear regression and spline smoothing", *J.A.S.A.* **78** (381) (1983) 81–89.