Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

CHAPTER 11
Numerical Methods

11.1 Numerical methods that use special structure.

We have noted that a basis for an $(n-m)$-dimensional subspace of span $R_{t_1}, \ldots, R_{t_n}$ in $W_m[0,1]$ may be obtained that has compact support. This results in a band structure for certain matrices, and numerical methods that make use of this structure should be faster than those that do not. See Reinsch (1971) for a fast algorithm for computing the univariate smoothing spline when λ is given. Various authors have come upon this special structure from different points of view. Looking at the $m-1$ fold integrated Weiner process $X(t)$ of (1.5.2) it can be seen that it is an m-ple Markov process in the sense of Dolph and Woodbury (1952), Hida (1960). This means that the prediction of $X(s)$ for any $s > t$, given $X(u)$, $u \in [0,t]$ is a function of $X^{(\nu)}(t), \nu = 0,1,\ldots,m-1$. Starting with this or similar reasoning, fast recursive formulas for the univariate polynomial smoothing spline have been obtained by various authors (see, e.g., Weinert and Kailath (1974)). Ansley and Kohn (1987) used this kind of reasoning to obtain a fast algorithm that included the computation of the GCV estimate of λ. Recently Shiau (1988) has used similar results to obtain fast algorithms for partial spline models in W_m with jumps. For some time it was an open question whether or not the special structure inherent in smoothing in W_m could be used to obtain an $O(n)$ algorithm for computing the univariate polynomial smoothing spline along with the GCV estimate of λ. Hutchinson and deHoog (1985) and O'Sullivan (1985b) appear to be the first to provide such an algorithm. A fast, accurate, user-friendly code based on the Hutchinson and deHoog paper, with some improvements, has been implemented by Woltring (1985, 1986) and may be obtained from netlib over the internet, as may O'Sullivan's code. Netlib is a robot library system run by Jack Dongarra and Eric Grosse and its internet address is netlib@research.att.com. If you write to netlib with "send index" in the body of the message, the robot mailserver will return instructions for using the system. O'Sullivan's and Woltring's code may be obtained this way. Code for generating B-splines based on deBoor's book may also be obtained this way. Earlier, Utreras (1983) provided a cheap way of evaluating trace $A(\lambda)$ in the equally spaced data case in W_m, based on good approximate values for the relevant eigenvalues. This method is implemented in Version 8 et seq of IMSL (1986).

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

11.2 Methods for unstructured problems.

Basic references for numerical linear algebra and optimization methods are Golub and Van Loan (1983) and Gill, Murray, and Wright (1981), respectively. The LINPACK manual (Dongarra et al. (1979)) is a good place to read about the singular value decomposition, the QR decomposition, Cholesky decomposition, Householder transformations, and other useful tools for computing the estimates in this book. See also the EISPACK manual (Smith et al. (1976)). Generally, the most computer-intensive part of the calculation of cross-validated estimates discussed in this book is going to be the calculation of trace $I - A(\lambda)$ in the denominator of the GCV function. The optimization of that calculation will usually dictate the manner in which the remaining calculations are done.

Considering the case where N basis functions are used, as in Secton 7.1, let

$$A(\lambda) = X(X'X + n\lambda\Sigma)^{-1}X'$$

$$= \tilde{X}(\tilde{X}'\tilde{X} + n\lambda I)^{-1}\tilde{X}'$$

where $\tilde{X} = X\Sigma^{-1/2}$, where $\Sigma^{-1/2}$ is any square root of Σ^{-1}. For small-to-moderate problems $\Sigma^{-1/2}$ may be obtained numerically as L^{-1} where LL' is the Cholesky decomposition of Σ, L being lower triangular, and hence numerically easy to invert (if it is not too ill-conditioned). Then

$$\operatorname{tr}(I - A(\lambda)) = n - \sum_{\nu=1}^{N} \frac{s_\nu^2}{s_\nu^2 + n\lambda}$$

where the s_ν's are the singular values of $\tilde{X}$. The singular value decomposition (SVD) in LINPACK (Dongarra et al. (1979)) can be used to compute the s_ν's. Elden (1984) proposed a method for computing trace $A(\lambda)$ based on a bidiagonalization of $\tilde{X}$ that is much faster than using the singular value decomposition of $\tilde{X}$ (see also Elden (1977)). Bates and Wahba (1982) proposed a method for computing trace $A(\lambda)$ based on truncating a pivoted QR decomposition of $\tilde{X}$, $\tilde{X}$ =QR. (Recall that Q is $n \times n$ orthogonal and R is $n \times N$ upper triangular; see Dongarra et al., Chap. 9 for the pivoted QR decomposition.) The pivoted QR decomposition permutes the columns of $\tilde{X}$ so that R has the property that its entries r_{ij} satisfy

$$r_{kk}^2 \geq \sum_{i=k}^{j} r_{ij}^2.$$

If R is replaced by R_{trunc}, which is obtained by setting r_{ij} to zero for $i = k+1, \ldots, N$, then the Weilandt–Hoffman theorem (Golub and Van Loan (1983, p.270) says that

$$\operatorname{trace}(\tilde{X} - \tilde{X}_{\text{trunc}})(\tilde{X} - \tilde{X}_{\text{trunc}})' = \sum_{i=k+1}^{N} \sum_{j=i}^{N} r_{ij}^2 = \tau, \text{ say.}$$

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Thus k can be chosen so that the tolerance τ is less than a prespecified small amount. This method can be expected to be useful when n and N are large and $\tilde{X}$ has a large number of eigenvalues near zero. In these kinds of cases it has sometimes been found that the LINPACK SVD will converge quickly on R_{trunc} when it converges slowly or not at all on $\tilde{X}$. It is implemented in GCVPACK (Bates et al. (1987)) and the code is available through netlib. GCVPACK also has a code for partial spline models where the smooth part is a thin-plate spline. Hutchinson and Bischof (1983) and Hutchinson (1984, 1985) have developed transportable code for thin-plate splines using the thin-plate basis functions of Wahba (1980b) described in (7.1.4) and (7.1.5). Recently, Girard (1987a,b) has proposed an ingenious method for estimating trace $A(\lambda)$ when n is very large, based on the fact that, if $\epsilon \sim \mathcal{N}(0, I)$, then $E\epsilon' A(\lambda)\epsilon = \text{trace}A(\lambda)$. A random vector ϵ is generated and $A(\lambda)\epsilon$ obtained in $O(n)$ operations by solving a linear system. A formula for the standard deviation of $\epsilon' A(\lambda)\epsilon$ is given, and if this standard deviation is too large, then the average of k replications of this estimate can be taken; the standard deviation will go down as $1/\sqrt{k}$. Hutchinson (1989) has studied this approach with the ϵ_i plus or minus one with probability $\frac{1}{2}$.

Gu et al. (1989) have considered the general unstructured case when V is defined by (4.3.1) with A as in (1.3.23),

$$V(\lambda) = \frac{1}{n} z'(\tilde{\Sigma} + n\lambda I)^{-2} z \Big/ \left(\frac{1}{n} \operatorname{tr} (\tilde{\Sigma} + n\lambda I)^{-1}\right)^2 \tag{11.2.1}$$

where $\tilde{\Sigma} = Q_2' \Sigma Q_2$ and $z = Q_2' y$. An outline of the major steps goes as follows.

(1) Tridiagonalize $\tilde{\Sigma}$ as

$$U' \tilde{\Sigma} U = \Delta$$

where U is orthogonal and Δ is tridiagonal. This can be done by successively applying the Householder transformation (see Dongarra et al. (1979)). A distributed truncation method is provided in Gu et al. (1988) for speeding up this step. Letting $x = Uz$, then

$$V(\lambda) = \frac{1}{n} x'(n\lambda I + \Delta)^{-2} x \Big/ \left(\frac{1}{n} \operatorname{tr} (n\lambda I + \Delta)^{-1}\right)^2.$$

(2) Compute the Cholesky decomposition $(n\lambda I + \Delta) = C'C$, where

$$C = \begin{bmatrix} a_1 & b_1 & & & & \\ & a_2 & b_2 & & & \\ & & \ddots & \ddots & & \\ & & & & a_{n-M-1} & b_{n-M-1} \\ & & & & & a_{n-M} \end{bmatrix}$$

is upper diagonal.

(3) Calculate $\text{tr}(C^{-1}C^{-1'})$ using a trick due to Elden (1984). Letting the ith row of C^{-1} be $\mathbf{c}_i'$, then we have $\text{tr}(C^{-1}C^{-1'}) = \sum_{i=1}^{n-M} \|\mathbf{c}_i\|^2$.

From

$$C^{-1'}C' = (\mathbf{c}_1, \ldots \mathbf{c}_{n-M}) \begin{bmatrix} a_1 & & & & & \\ b_1 & a_2 & & & & \\ & b_2 & & & & \\ & & \ddots & & & \\ & & & \ddots & a_{n-M-1} & \\ & & & & b_{n-M-1} & a_{n-M} \end{bmatrix} = I$$

we have

$$\begin{array}{lll} a_{n-M} & \mathbf{c}_{n-M} & = \mathbf{e}_{n-M} \\ a_i & \mathbf{c}_i & = \mathbf{e}_i - b_i \mathbf{c}_{i+1}, \quad i = n-M-1, \ldots, 1 \end{array}$$

where the $\mathbf{e}_i$'s are unit vectors. Because $C^{-1'}$ is lower triangular, $\mathbf{c}_{i+1}$ is orthogonal to $\mathbf{e}_i$, giving the recursive formula

$$\begin{aligned} \|\mathbf{c}_{n-M}\|^2 &= a_{n-M}^{-2}, \\ \|\mathbf{c}_i\|^2 &= (1 + b_i^2 \|\mathbf{c}_{i+1}\|^2) a_i^{-2}, \ i = n-M-1, \ldots, 1. \end{aligned}$$

11.3 Methods for multiple smoothing parameters, with application to additive and interaction splines

The algorithm of Gu et al. (1989) has been used as a building block by Gu and Wahba (1991) in an algorithm for minimizing $V(\lambda, \theta)$ of (10.1.7), thus allowing the calculation of additive and interaction splines with multiple smoothing parameters chosen by GCV. The software appears in Gu (1989) and can be obtained from netlib. Here recall that $V(\lambda, \theta)$ is given by (11.2.1) with $\tilde{\Sigma}$ replaced by $\theta_1 \tilde{\Sigma}_1 + \ldots + \theta_p \tilde{\Sigma}_p$,

$$V(\lambda, \theta) = \frac{\frac{1}{n} z'(\theta_1 \tilde{\Sigma}_1 + \ldots + \theta_p \tilde{\Sigma}_p + n\lambda I)^{-2} z}{\left(\frac{1}{n} \operatorname{tr} (\theta_1 \tilde{\Sigma}_1 + \ldots + \theta_p \tilde{\Sigma}_p + n\lambda I)^{-1}\right)^2}. \tag{11.3.1}$$

As noted previously, all sets (θ, λ) with $\lambda_\beta = \lambda/\theta_\beta$ are equivalent. However a minimization of (11.3.1) in λ is "cheap" compared to a minimization in components of θ. We briefly describe the algorithm. The algorithm works iteratively by first fixing θ and minimizing $V(\lambda|\theta)$ by the algorithm in Gu et al. (1988). Then, for fixed λ, the gradient and Hessian of $V(\theta|\lambda)$ are evaluated with respect to the variables $\rho_\beta = \log \theta_\beta$, and the ρ_β are updated by a modified Newton method. In this sort of optimization it is important to get good (relative) starting values for the θ's. The default starting values in Gu and Wahba (1988) are obtained as follows. The initial θ_β's are taken as

$$\theta_\beta^{(0)} = (\operatorname{tr} \tilde{\Sigma}_\beta)^{-1}.$$

Then $V(\lambda|\theta)$ is minimized. (By this we mean $V(\lambda, \theta)$ is considered as a function of λ with θ given.) This results in a trial $f_{\lambda,\theta}$, with $\|P^\beta f_{\lambda,\theta}\|^2 = (\theta_\beta^{(0)})^2 c' \tilde{\Sigma}_\beta c$,

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

from (10.1.9). New starting values $\theta_\beta^{(1)}$ of the θ_β's are taken as

$$\theta_\beta^{(1)} = (\theta_\beta^{(0)})^2 c' \tilde{\Sigma}_\beta c,$$

where c is associated with the trial $f_{\lambda,\theta}$. $V(\lambda|\theta^{(1)})$ is then minimized with respect to λ via the algorithm of Gu et al. (1989). $\theta_\beta^{(2)}$ and successive θ_β's are obtained by the modified Newton update, alternating minimizations with respect to λ. The algorithm has been observed to converge rapidly in a number of examples.

Here convergence is defined in terms of minimizing $V(\theta, \lambda)$, and is declared to have occurred if the gradient is small and V no longer decreases. In examples this has also been seen to drive the predictive mean-square error $T(\lambda, \theta)$ to a minimum. This does not *necessarily* mean that a particular $\lambda_\beta = \lambda/\theta_\beta$ has converged, since if the predictive mean-square error $T(\lambda, \theta)$ is insensitive to certain variations in the λ_β, then so will be $V(\lambda, \theta)$, and these cannot be sorted out by minimizing $V(\lambda, \theta)$. For predictive mean-square error purposes, one would presumably be indifferent to them. Loosely speaking, λ_α will be more or less distinguishable from λ_β, according to whether $\operatorname{Tr} \tilde{\Sigma}_\alpha \tilde{\Sigma}_\beta$ is small or large.

In Figures 11.1 and 11.2 we present the results of a Monte Carlo example of an additive model with $d = 4$, $m = 2$, $n = 100$. The data were generated by

$$y_i = f(\mathbf{x}(i)) + \epsilon_i, \ i = 1, \ldots, n$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 1$, and $\mathbf{x} = (x_1, x_2, x_3, x_4)$ with

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= 10 \sin(\pi x_2) + \exp(3x_3) \\ &\quad + 10^6 x_4^{11}(1 - x_4)^6 + 10^4 x_4^3 (1 - x_4)^{10} \\ &= f_2(x_2) + f_3(x_3) + f_4(x_4), \text{ say.} \end{aligned}$$

Thus the truth is additive with no parametric or main effects component for x_1. The $\mathbf{x}(i)$ were random uniform variables on the unit 4-cube. Figure 11.1 gives a scatter plot of the $x_\alpha(i)$, $i = 1, 2, 3, 4$ and y_i. The dashed lines in Figure 11.2 are the $f_\alpha(x_\alpha)$, with $f_1(x_1) \equiv 0$, and the solid lines are the estimates. The main effects are uniquely determined by the fact that their average values are zero. They have been shifted in the plots to match the means of the f_α's. (The constant component of the model was estimated to better than the visual resolution of the picture.)

Table 11.1 gives the value of $V(\lambda, \theta)$ and $T(\lambda, \theta)$ after each iteration cycle (after the λ-step). It can be seen that convergence was quite rapid and T appears to decrease along with V.

11.4 Applications to components of variance, a problem from meteorology

A similar algorithm for minimizing the GML function

$$M(\theta, \lambda) = \frac{z'(\theta_1 \tilde{\Sigma}_1 + \cdots + \theta_p \tilde{\Sigma}_p + n\lambda I)^{-1} z}{[\det(\theta_1 \tilde{\Sigma}_1 + \cdots + \theta_p \tilde{\Sigma}_p + n\lambda I)^{-1}]^{1/n-M}} \tag{11.4.1}$$

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

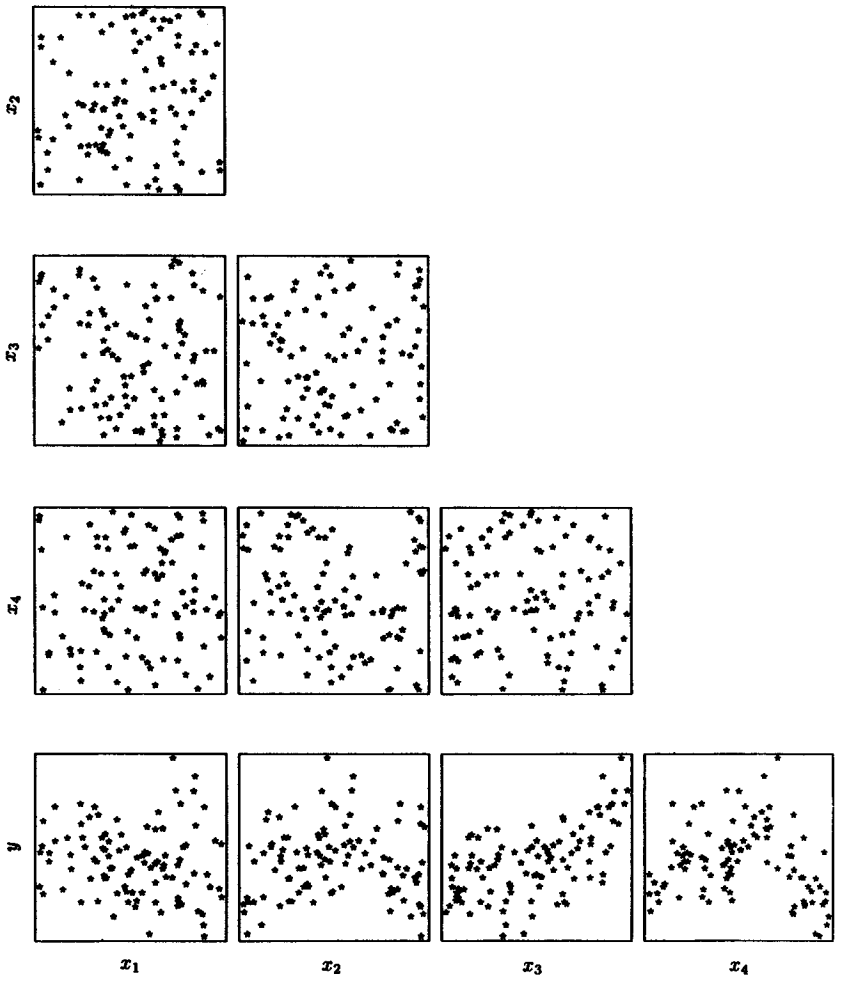

FIG. 11.1. *Scatter plot matrix for the additive model.*

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

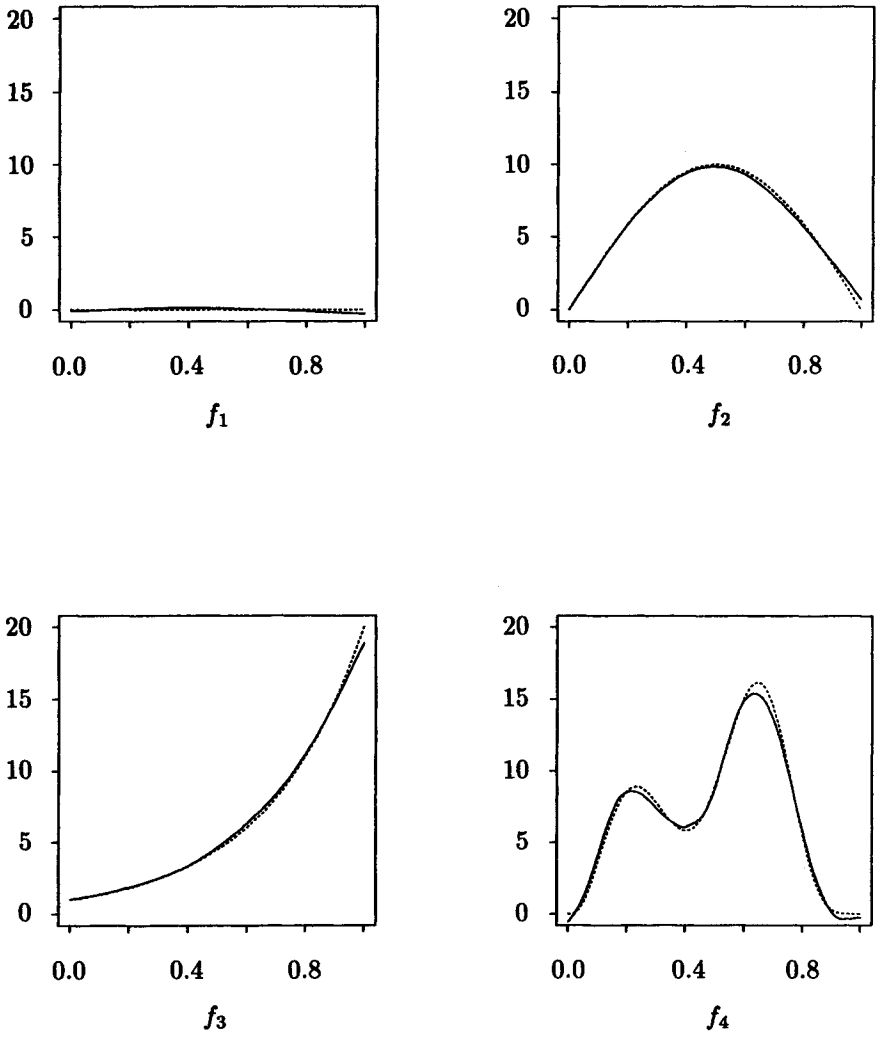

FIG. 11.2. *Estimates (solid) and "truth" (dashed) for the additive model.*

TABLE 11.1
V and T, iterated values.

Iteration No.	V	T
0	1.50409	.291176
1	1.50180	.232195
2	1.45412	.273181
3	1.41040	.243224
4	1.40893	.234954
5	1.40893	.234726

(compare (4.8.4)), is given in Gu et al. (1989) and Gu and Wahba (1988). The problem of minimizing $M(\theta, \lambda)$ comes up in components of variance.

An important example comes from meteorology. We outline the mathematical ideas here, and further details can be found in Wahba (1989).

Let f be a meteorological field of interest (for example, atmospheric temperature, wind, geopotential, humidity), and let the data from the βth data source (sensor or forecast) be

$$y_{i\beta} = L_i^\beta f + \epsilon_i^\beta \ , \ i = 1, \ldots, n_\beta, \ \beta = 0, 1, \ldots, q, \tag{11.4.2}$$

and suppose the $\epsilon^\beta = (\epsilon_1^\beta, \ldots, \epsilon_{n\beta}^\beta)'$ are independent, $\mathcal{N}(0, \omega_\beta \Sigma_\beta)$, where the Σ_β are known. For purposes of estimating f, it is desirable to know the relative values of the weights ω_β for the different data sources. If these relative values were known, all of the ingredients would be available to estimate f from the pooled data of (11.4.2) by minimizing the appropriate weighted residual sum of squares plus a penalty functional.

Suppose we can find $q+1$ matrices B_β of dimension $n \times n_\beta$, $\beta = 0, 1, \ldots, q$ such that

$$\sum_{\beta=0}^{q} B_\beta L^\beta = 0,$$

where $L^\beta = (L_1^\beta, \ldots, L_{n_\beta}^\beta)'$. Then

$$u \equiv \sum_{\beta=0}^{q} B_\beta y^\beta \sim \mathcal{N}\left(0, \sum_{\beta=0}^{q} \omega_\beta B_\beta \Sigma_\beta B_\beta'\right)$$

where $y^\beta = (y_1^\beta, \ldots, y_{n_\beta}^\beta)'$. Suppose that $B_0 \Sigma_0 B_0'$ is well-conditioned. Then taking the eigenvalue eigenvector decomposition $B_0 \Sigma_0 B_0' = UDU'$, and letting

$$z = D^{-1/2} U' u,$$

we have

$$z \sim \mathcal{N}(0, \omega_1 \tilde{\Sigma}_1 + \cdots + \omega_q \tilde{\Sigma}_q + \omega_0 I),$$

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $\tilde{\Sigma}_\beta = D^{-1/2}U'\Sigma_\beta U D^{-1/2}$. Letting $\omega_0 = \sigma^2 n\lambda$ and $\omega_\beta = \sigma^2\theta_\beta$, one can minimize the negative log-likelihood for z with respect to σ^2 explicitly. Substituting in the resulting $\hat{\sigma}^2$, one is left with an expression of the form of (11.4.1) to minimize. In order to get good estimates of $\lambda_\beta = \lambda/\theta_\beta$, it is necessary that the correlation structure of the different data sets $B_\beta y^\beta$ be sufficiently different. Some examples where this is likely to happen in practice are given in Lonnberg and Hollingsworth (1986). For some other examples of the use of multiple smoothing parameters in meteorological applications, see Wahba (1982d), Hoffman (1984, 1985), and Legler, Navon, and O'Brien (1989). A. Hollingsworth (1989) informs us that development of new data analysis for estimating initial conditions for use in the European Center for Medium-Range Weather Forecasts is following along lines suggested by Wahba (1982d).

Downloaded 09/05/14 to 128.104.46.206. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php