An iterative procedure for calculating minimum generalised cross validation smoothing splines

P. A. Hancock* M. F. Hutchinson†

(Received 1 June 2001)

Abstract

This study analyses a simple iterative procedure for estimating minimum generalised cross-validation (GCV) univariate smoothing splines. The results provide guidelines for the development of a similar methodology to estimate minimum GCV bivariate thin plate smoothing splines. The methodology is based on the techniques described in Hutchinson [*ANZIAM J*, 42(E):C774–C796, 2000], which uses nested grid SOR iterative methods in order to solve finite element thin plate smoothing spline systems efficiently for large data sets. The method also uses the stochastic approximation

*Centre for Resource and Environmental Studies, The Australian National University, Australia. mailto:penlepen@cres.anu.edu.au

†Centre for Resource and Environmental Studies, The Australian National University, Australia. mailto:hutch@cres.anu.edu.au

[0]See http://anziamj.austms.org.au/V44/CTAC2001/Hanc for this article, Published 1 April 2003. ISSN 1446-8735

to the GCV developed by Hutchinson [*Commun. Stats — Sim. and Comp.*, 18:1059–1076, 1989]. A double iteration is used to produce increasingly accurate estimates of the minimum GCV smoothing parameter and the smoothing spline. First and second derivatives of the GCV with respect to the smoothing parameter are used to update the smoothing parameter. Convergence of the SOR iteration is improved significantly by correcting the solution estimate after each smoothing parameter update using the estimate of the derivative of the solution with respect to the smoothing parameter.

Contents

1 Introduction

Thin plate smoothing splines are commonly used to fit smooth surfaces to scattered noisy data. Analytic methods of calculating thin plate smoothing splines require $\mathcal{O}(n^3)$ operations. A number of strategies for improving the computational efficiency of thin plate smoothing spline calculation exist [2, 3, 4]. Finite element techniques have also been developed [12, 10]. All of these methods tend to focus on the numeric-analytic properties of thin plate smoothing splines rather than their statistical properties, and do not incorporate a mechanism for optimising smoothness.

For practical spatial interpolation problems, surface smoothness is a central issue given that the data observations contain a significant noise component. One method of optimising smoothness is to minimise the generalised cross validation, which is a measure of predictive error. Minimising the generalised cross-validation (GCV) has been shown by Craven and Wahba [6] to be an accurate method of estimating the amount of smoothing corresponding to the minimum error solution. Minimum GCV thin plate smoothing splines have been widely used in spatial interpolation applications [11, 15, 23, 21, 19, e.g.]. The methods presented in this study are designed to efficiently compute accurate finite element approximations to univariate smoothing splines, incorporating procedures for optimising smoothness by minimising GCV. Estimation of the GCV requires the stochastic estimator of the trace of the influence matrix, developed by Hutchinson [13]. Analysis of the algorithm for univariate splines was intended to assist in the design of an optimal numerical procedure for estimating finite element minimum GCV bivariate thin plate smoothing splines.

Hutchinson [16] developed a simple multigrid based strategy which calculates finite element approximations to thin plate smoothing splines for elevation data in $\mathcal{O}(N)$ operations, where N is the number of grid points. This method emphasises the statistical framework of thin plate smoothing splines [14], and optimises the smoothness to yield a user specified residual sum of squares. This criterion is appropriate in the context of interpolating topography, where an estimate of the amount of noise is available [12]. The procedure presented here is a variation on the Hutchinson [16] method, in that it optimises smoothness by minimising GCV rather than prescribing a residual sum of squares from the data.

A hierarchial Quadratic B-spline framework was used to estimate the smoothing spline solution, to meet the natural requirement of first derivative continuity while maintaining compact support. The nested grid procedure used a double iteration to produce increasingly accurate estimates of the minimum GCV smoothing parameter and the smoothing spline. The optimal smoothing parameter was estimated using a Taylor's series expansion of the GCV in terms of the smoothing parameter. Methods described in Hutchinson [16] were used to calculate the derivatives of the solution estimate with respect to the smoothing parameter. Convergence was improved by making a first order correction to the solution as values of the smoothing parameter were updated. Further analysis of the performance of this algorithm led to modifications that increased the efficiency of the derivative estimations.

2 Quadratic B-splines

The B-spline framework was developed by Schoenberg [22] and is described in de Boor [5]. In one dimension, the rth normalised B-spline $B_{r,k}(x)$ of degree k, and 'order' $k+1$, with knots $\gamma_r, \ldots, \gamma_{r+k+1}$

is defined recursively (de Boor [5]):

$$B_{r,0}(x) = \begin{cases} 1\,, & \text{if } \gamma_r \le x \le \gamma_{r+1}\,, \\ 0\,, & \text{otherwise}\,; \end{cases} \tag{1}$$

$$B_{r,k+1}(x) = \frac{x-\gamma_r}{\gamma_{r+k+1}-\gamma_r} B_{r,k}(x) + \frac{\gamma_{r+k+2}-x}{\gamma_{r+k+2}-\gamma_{r+1}} B_{r+1,k}(x)\,. \tag{2}$$

The B-spline functions $B_{r,k}$ form a basis for the vector space of spline functions of degree k where $r = 0, 1, \ldots, N+k-1$ and N is the number of knot intervals. The basic properties of this B-spline basis are

$$B_{r,k}(x) \ge 0\,, \quad \text{for all } x \in \mathcal{R}\,; \tag{3}$$

$$\operatorname{supp} B_{r,k} = [\gamma_r, \gamma_{r+k+1}]\,; \tag{4}$$

$$\sum_{r=0}^{N+k-1} B_{r,k}(x) = 1\,. \tag{5}$$

Quadratic B-splines, or B-splines of order 3, can be simply defined in terms of first order B-splines using (2). They are continuous functions with continuous first derivatives and support of 3 knot intervals. This leads to 5-banded systems of discretised smoothing spline equations, as discussed in Section 4. While they require more storage than the diagonal systems arising from first order B-spline (or finite difference) discretisations, the results presented in Section 6 show that the ability of quadratic B-splines to produce smooth functions at coarse discretisations is essential to the development of an efficient minimum GCV algorithm.

3 Hierarchial B-splines

A hierarchial spline space can be defined as a linear span of B-splines with nested knot sequences [20]. The hierarchial spline space provides a natural framework for defining different resolutions of B-spline discretisations for use in a multigrid scheme. Consider the knot sequences

$$\gamma_{r,l} = a + r2^l h\,, \quad r = 0, \ldots, N_l + k + 2\,, \quad l = 0, 1, 2, \ldots \tag{6}$$

where a is the first knot in the sequence, l is the grid level, h is the width of the knot intervals on the finest level and N_l is the number of knot intervals on level l. The width of the knot intervals doubles as l is incremented. The B-spline $B^l_{r,k}$ of level l is the B-spline to the knot sequence $\gamma_{r,l}, \ldots, \gamma_{r+k+1,l}$ with $\operatorname{supp} B^l_{r,k} = [\gamma_{r,l}, \gamma_{r+k+1,l}]$. Consider the spaces

$$\mathcal{S}_l = \operatorname{span}\{B^l_{r,k}\} = \left\{ s^l \in S_l \mid s^l = \sum_{r=1}^{d_l} \alpha_r B^l_{r,k}\,;\ \alpha_r \in \mathcal{R} \right\},$$

where d_l is the dimension of level l. These spaces form a sequence of nested subspaces of $\mathcal{S}$, such that

$$\mathcal{S}_p \subset \mathcal{S}_{p-1} \subset \cdots \mathcal{S}_1 \subset \mathcal{S}_0\,. \tag{7}$$

This means that any element in $\mathcal{S}_l$ can be written as a linear combination of the basis vectors $B^{l-1}_{r,k}$ of $\mathcal{S}_{l-1}$, although this representation would be redundant since the dimension of $\mathcal{S}_l$ is half that of $\mathcal{S}_{l-1}$. B-splines are therefore 'refinable' [7], in that each one can be re-expressed as a linear combination of one or more 'smaller' basis functions. In the multigrid context, this means that standard intergrid transfer operators are not required. In the case of nested grid, refinement occurs by expressing each coarse grid basis element in terms of the fine grid basis elements:

$$B^l_{r,k} = \sum_{i=2r-2}^{2r+1} \beta_i B^{l-1}_{i,k}(x)\,, \tag{8}$$

where $(\beta_{2r-2}, \beta_{2r-1}, \beta_{2r}, \beta_{2r+1}) = (1/4, 3/4, 3/4, 1/4)$ [7].

4 Quadratic B-spline approximation of univariate smoothing splines

For the purposes of obtaining a smooth univariate representation of noisy data, standard practice is to choose the function $y(x)$ that minimises

$$\frac{1}{n}\sum_{i=1}^{n}(z_i - y(x_i))^2 + \lambda \int_a^b [y''(x)]^2\,dx\,, \quad x \in [a,b]\,, \tag{9}$$

where z_i are the data observations at locations x_i, n is the number of data points, and λ is a positive smoothing parameter [8]. The minimiser, $y(x)$, of expression (9) over $C^2[a,b]$ (the space of all functions that are continuous, and have continuous first and second derivatives on the interval $[a,b]$) is a natural cubic spline. $\mathcal{O}(n)$ algorithms exist for calculating the univariate analytic spline solution [17]. However, this study investigates numerical approximations to univariate splines that can be extended to higher dimensions.

A quadratic B-spline approximation $f(x)$ of the function $y(x)$ in equation (9) is

$$f(x) = \sum_{r=0}^{N+1} \alpha_r B_{r,2}(x)\,. \tag{10}$$

Thus f is a spline function composed of quadratic B-spline elements. We substitute the approximation f into equation (9) and choose the coefficients α_r to minimise (9). This requires differentiating f with respect to x. For this, the relation

$$\frac{d}{dx}\left(\sum_{r=0}^{N+1}\alpha_r B_{r,k}\right) = \sum_{r=1}^{N+1}\frac{k(\alpha_r - \alpha_{r-1})}{\gamma_{r+k} - \gamma_r} B_{r,k-1}(x)\,, \tag{11}$$

was used [5]. See that the first derivative of a spline function f is found simply by differencing its B-spline coefficients. For quadratic splines with knots equally spaced at intervals of length h this gives

$$f'(x) = \sum_{r=1}^{N+1} \frac{(\alpha_r - \alpha_{r-1})}{h} B_{r,1}(x)\,, \tag{12}$$

$$f''(x) = \sum_{r=2}^{N+1} \frac{\alpha_r - 2\alpha_{r-1} + \alpha_{r-2}}{h^2} B_{r,0}(x)\,. \tag{13}$$

Thus the coefficients of the second derivative of a quadratic B-spline are given by the second difference of its coefficients. The simplicity of this relationship further motivates the choice of quadratic B-splines. The second term of the minimisation problem in equation (9) is then

$$\begin{aligned} \lambda \int \left(f''(x)\right)^2 dx &= \lambda h \sum_{r=2}^{N+1} \left(\frac{\alpha_r - 2\alpha_{r-1} + \alpha_{r-2}}{h^2} \right)^2 \\ &= \frac{\lambda}{h^3} \sum_{r=2}^{N+1} (\alpha_r - 2\alpha_{r-1} + \alpha_{r-2})^2\,. \end{aligned} \tag{14}$$

The discretised minimisation problem is then

$$\|P\alpha - z\|^2 + \frac{\lambda}{h^3} \|Q\alpha\|^2\,, \tag{15}$$

where α is a vector containing the $N+2$ quadratic B-spline coefficients α_r and z is a vector of length n containing the data observations z_i. The matrix P operates on α to calculate values of $f(x)$ at data point locations i and is

$$[P]_{ir} = B_{r,2}(x_i)\,. \tag{16}$$

As there are at most 3 non-zero B-splines in any knot interval, the matrix P has no more than 3 non-zero entries in each row. The matrix Q is a 5-banded finite difference matrix calculating the second differences in equation (14). Differentiating expression (15) with respect to α and equating to zero gives

$$\left(P^T P + \frac{\lambda}{h^3} Q^T Q\right) \alpha = P^T z = v^0 \,. \qquad (17)$$

Techniques for numerically solving this system whilst simultaneously optimising the smoothing parameter are discussed below. The conditioning of (17) had important ramifications in designing the numerical algorithm. The matrix $Q^T Q$ is rank deficient, which means the system is poorly conditioned for large values of λ/h^3. This situation occurs at fine grid spacings, especially if the optimal solution is highly smooth and has a large value of λ.

5 Iterate for optimum λ to minimise GCV

The techniques used in this study for optimising the parameter λ in equation (17) were based on an adaptive iterative strategy described in Hutchinson [16]. The process involves double iteration to produce increasingly accurate estimates of both the solution and the smoothing parameter. The method uses nested grid to iteratively solve equation (17) whilst periodically updating the estimate of λ using the current solution estimate.

5.1 Estimate the derivatives of the GCV with respect to the smoothing parameter

Estimates of the value of the smoothing parameter corresponding to minimum GCV were obtained by a second order Taylor's series approximation:

$$\textsc{gcv}(\theta) = \textsc{gcv}(\theta_q) + \frac{d\textsc{gcv}(\theta_q)}{d\theta}(\theta - \theta_q) + \frac{d^2\textsc{gcv}(\theta_q)}{d\theta^2}\frac{(\theta-\theta_q)^2}{2} \qquad (18)$$

where $\theta = \log\lambda$. The value of θ that minimises GCV is estimated by

$$\begin{aligned} \theta &= -b/2c\,, && (19)\\ \text{where}\quad b &= -\frac{d\textsc{gcv}(\theta_q)}{d\theta} + \frac{d^2\textsc{gcv}(\theta_q)}{d\theta^2}\theta_q\,, && (20)\\ c &= \frac{1}{2}\frac{d^2\textsc{gcv}(\theta_q)}{d\theta^2}\,. && (21)\end{aligned}$$

The derivatives of the GCV with respect to θ are calculated using

$$\textsc{gcv} = n\frac{R}{\mathrm{Tr}^2}\,, \qquad (22)$$

where R is the residual sum of squares $\sum_{i=1}^{n}(z_i - f(x_i))^2$ and Tr is trace$(I - A)$, where A is the influence matrix [14]. Differentiating expression (22) with respect to θ gives

$$\begin{aligned} \frac{dGCV}{d\theta} &= = \frac{n}{\mathrm{Tr}^4}\left(\frac{dR}{d\theta}\mathrm{Tr}^2 - 2\mathrm{Tr}\frac{d\mathrm{Tr}}{d\theta}R\right)\,, && (23)\\ \frac{d^2GCV}{d\theta^2} &= \frac{n}{\mathrm{Tr}^4}\left[\frac{d^2R}{d\theta^2}\mathrm{Tr}^2 - \left(2\mathrm{Tr}\frac{d\mathrm{Tr}}{d\theta} + 2\left(\frac{d\mathrm{Tr}}{d\theta}\right)^2\right)R\right]\\ &\quad - \frac{4n}{\mathrm{Tr}^5}\frac{d\mathrm{Tr}}{d\theta}\left(\frac{dR}{d\theta}\mathrm{Tr}^2 - 2\mathrm{Tr}\frac{d\mathrm{Tr}}{d\theta}R\right)\,. && (24)\end{aligned}$$

It is shown in Hutchinson [16] that

$$\begin{aligned} \frac{dR}{d\theta} &= 2(v^1)^T\frac{d\alpha}{d\theta}\,, && (25)\\ \text{and}\quad \frac{d^2R}{d\theta^2} &= 2(v^1)^T\frac{d^2\alpha}{d\theta^2} + 2\left(P^TP\frac{d\alpha}{d\theta}\right)^T\frac{d\alpha}{d\theta}\,, && (26)\\ \text{where}\quad v^1 &= P^T(P\alpha - z)\,. && (27)\end{aligned}$$

Hutchinson [16] further demonstrates that $d\alpha/d\theta$ satisfies the same system of equations as (17) with different right hand side vectors, so that

$$\left(P^TP + \frac{\lambda}{h^3}Q^TQ\right)\frac{d\alpha}{d\theta} = v^1\,, \tag{28}$$

and similarly it can be seen that

$$\left(P^TP + \frac{\lambda}{h^3}Q^TQ\right)\frac{d^2\alpha}{d\theta^2} = v^2\,, \tag{29}$$

where

$$v^2 = -v^1 + 2P^TP\frac{d\alpha}{d\theta}\,. \tag{30}$$

These equations are therefore solved iteratively with minimum additional storage requirement.

The estimates of Tr and its derivatives were obtained using the stochastic estimate developed by Hutchinson [13]. Hutchinson [13] showed that u^TAu is a minimum variance, unbiased estimator of $\mathrm{tr}A$, where $u = (u_1, \ldots, u_n)^T$ is a vector of n independent samples from the random variable U that takes the values ± 1 each with probability $1/2$. An estimate of Au is obtained by solving

$$\left(P^TP + \frac{\lambda}{h^3}Q^TQ\right)b = P^Tu = w^0\,. \tag{31}$$

An estimate of Au is then given by Pb. A similar stochastic estimator has been developed by Girad [9].

To obtain estimates of $d\mathrm{Tr}/d\theta$ and $d^2\mathrm{Tr}/d\theta^2$, a slightly different approach was taken to that used to solve for the derivatives of R. Differentiating equation (31) with respect to λ instead of θ gives

$$\left(P^TP + \frac{\lambda}{h^3}Q^TQ\right)\frac{db}{d\lambda} = -\frac{Q^TQ}{h^3}b = w^1\,. \tag{32}$$

Note that the right hand side of equation (32) has been expressed in terms of the solution vector b rather than the data vector u [16]. Differentiating $u^T Pb$ with respect to θ and using (31) and (32), the derivatives of Tr with respect to θ are

$$\frac{d\mathrm{Tr}}{d\theta} = \frac{\lambda}{h^3} b^T Q^T Q b\,, \tag{33}$$

$$\frac{d^2\mathrm{Tr}}{d\theta^2} = 2\frac{\lambda^2}{h^3}(Q^T Q b)^T \frac{db}{d\lambda} + \frac{\lambda}{h^3} b^T Q^T Q b\,. \tag{34}$$

These expressions use that the matrix $(P^T P + \frac{\lambda}{h^3} Q^T Q)$ is symmetric. Note that the expression for $d\mathrm{Tr}/d\theta$ does not involve $db/d\lambda$ and $d^2\mathrm{Tr}/d\theta^2$ only requires the first derivative of b with respect to λ. Tr and its first and second derivatives are therefore obtained from two systems of equations (31) and (32), rather than the three equations (17), (28) and (29) required to calculate the derivatives of R.

5.2 The algorithm

A nested grid procedure starting on a grid of coarseness 2^l and refining by a factor of 2 was constructed, using equation (19) to periodically update the θ estimate. Convergence of the smoothing parameter estimate on a given grid was determined by a criterion Q, and the refinement process was terminated by a criterion D. The algorithm is:

while $D >$ tol1
    $q = 1$
    while $Q >$ tol2
        for $m = 0$ to 2
            $\alpha_l(\theta_q) = S_l^{v_1}(\alpha_l^{(m)}(\theta_q), v_l^m)$

end
for $n = 0$ to 1
$b_l(\theta_q) = S_l^{\upsilon_1}(b_l^{(n)}(\theta_q), w_l^n)$
end
$\theta_{q+1} = -\frac{b}{2c}$
$q = q + 1$
end
for $m = 0$ to 2
$\alpha_{l-1}^{(m)} = T_l \alpha_l^{(m)}$
end
for $n = 0$ to 1
$b_{l-1}^{(n)} = T_l b_l^{(n)}$
end
$l = l - 1$
end
$q = 1$
while $Q >$ tol2
for $m = 0$ to 2
$\alpha_l(\theta_q) = S_l^{\upsilon_1}(\alpha_l^{(m)}(\theta_q), v_l^m)$
end
for $n = 0$ to 1
$b_l(\theta_q) = S_l^{\upsilon_1}(b_l^{(n)}(\theta_q), w_l^n)$
end
$\theta_{q+1} = -\frac{b}{2c}$
$q = q + 1$
end

where S_l is the SOR iteration matrix on a grid of coarseness l, θ_q is the qth update, υ_1 is the number of smoothing iterations per update, and $\alpha_l^{(m)}$ denotes the mth derivative of α_l with respect to θ. The matrix T_l is a prolongation operator corresponding to equation (8).

This operator is of the form

$$\frac{1}{4}\begin{pmatrix} 3 & 1 & & & & & \\ 1 & 3 & & & & & \\ & 3 & 1 & & & & \\ & 1 & 3 & & & & \\ & & & \ddots & & & \\ & & & & & 3 & 1 \\ & & & & & 1 & 3 \end{pmatrix}.$$

The criterion Q was set to the sum of the absolute differences between 4 consecutive θ updates. The resolution of the final grid level l was determined by a tolerance on the criterion D, where

$$D = \frac{\|g_l\|}{\|f_l\|}\,. \tag{35}$$

The function $g_l = f_l - f_r$, where f_r is the least squares restriction of f_l onto grid $l+1$. Thus $\|g_l\|$ is a measure of the fine scale information contributed by the finer grid. If this measure was less than the tolerance, it was considered unnecessary to continue refinement. This criterion is an objective method of matching the grid resolution to the scale of the data generation process. The results presented below show that it is important to avoid iterating on unnecessarily fine grids, because the conditioning of equations (17) deteriorates as the resolution becomes finer. Poor conditioning greatly decreases the efficiency of the algorithm, and can lead to non-convergence.

6 Performance

The algorithm was tested using a data set consisting of 101 noisy data points randomly perturbed from a test function $t(x)$ by values

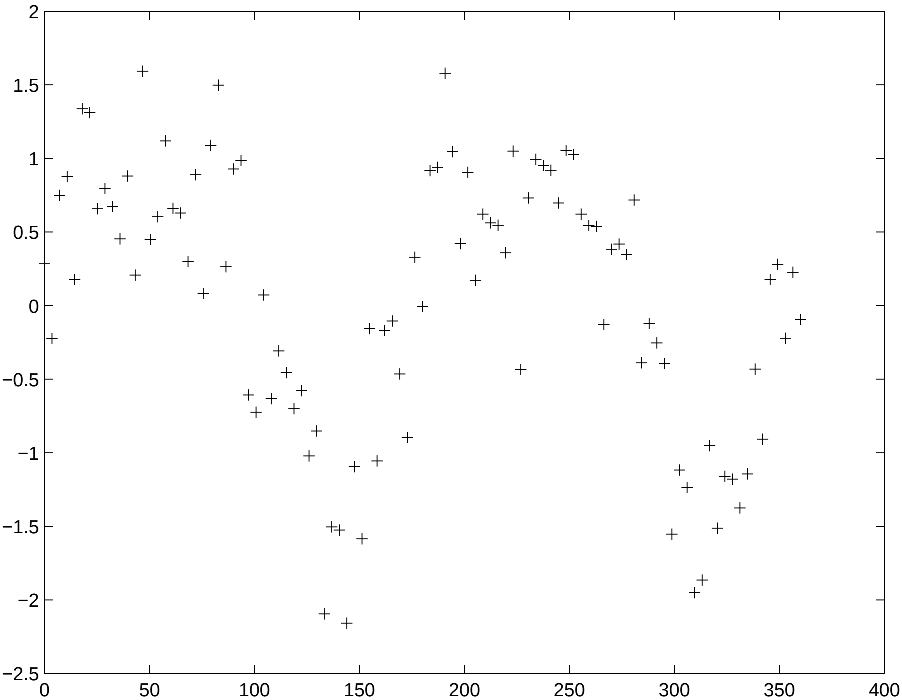

Figure 1: The simulated data, created from $t(x)$.

from a zero mean normal variable with standard deviation 0.5 . The function $t(x)$ was chosen to be

$$\sin\frac{2\pi x}{180} + 0.5\cos\frac{4\pi x}{180}\,, \qquad 0 \leq x \leq 360^\circ\,. \tag{36}$$

The data points were spaced at regular intervals of 3.6° . This data set is shown in Figure 1.

The algorithm was run with the initial solution estimate set to zero. The value of v_1 was optimised experimentally to 5 smoothing iterations per update. The smoothing parameter was initialised to the value that gave approximately equal weighting to the two terms

Table 1: Performance of the algorithm on each grid.

l	h^l	No. of updates	FLOPS
5	32	15	56 000
4	16	11	78 000
3	8	7	100 000
2	4	non-	convergence

on the left hand side of (17). This setting was chosen in order to give equal priority to smoothness and fidelity to the data, as recommended by Hutchinson and Gessler [18].

Results for the algorithm are shown in Table 1. On the three coarsest grids (levels 5, 4 and 3), the smoothing parameter estimate converges. A higher number of updates was required on the coarsest grid because the initial solution estimate contained no information. Iteration quickly became more computationally expensive as the grid resolution increased.

The failure of the algorithm to converge on level 2 is attributed to the deterioration in the conditioning of system (17) on fine grids. Poor conditioning leads to slow convergence of the SOR iteration. This results in higher error in the θ updates. It also means that the solution estimate is slow to respond to changes in the smoothing parameter estimate. The algorithm is therefore poorly synchronised, and eventually diverges. The inability of basic iteration to alter smooth components on fine grids is well known [1]. This emphasises the importance of using basis elements that represent the smooth components of the solution accurately on coarse grids.

Although it was found that $\|g_l\|$ was usually small enough to stop refinement before synchronisation issues became a problem, it was not the case for this particular random vector u. However, a first derivative correction to the solution estimate restored convergent

TABLE 2: Performance of the algorithm on each grid, using the first derivative correction to the solution estimate.

l	h^l	No. of updates	FLOPS
5	32	12	45 500
4	16	9	65 000
3	8	6	87 500
2	4	12	318 500

behaviour. The correction

$$\alpha(\theta_{q+1}) = \alpha(\theta_q) + \frac{d\alpha}{d\theta}(\theta_q)(\theta_{q+1} - \theta_q) \tag{37}$$

was added following each update, where equation (28) was used to estimate $d\alpha/d\theta$. The results in Table 2 show that the algorithm converged on grid 2. Refinement did not continue beyond this grid due to the criterion on $\|g_l\|$.

A plot of the solution overlaid on the analytic solution is shown in Figure 2. The statistics of the analytic thin plate spline and the finite element solution are given in Table 3. The statistics produced by the algorithm are subject to considerable stochastic error because the converged solution estimate depends on a stochastic estimate of Tr. The standard deviation of the Tr estimate is bounded by $(2/n)^{1/2}$ [13], which is high for a small data set. However, this standard error is insignificant in the case of the large data sets for which this algorithm was designed. For this analysis, the smoothing parameter has been underestimated and too much fine scale structure has been incorporated. This is reflected by the lower R and Tr values produced by the algorithm, in comparison to the analytic values.

A further reduction in computations was obtained by using a finite difference approximation of $d^2\text{GCV}/d\theta^2$ instead of using equation (24). This removed the need to iteratively solve for $d^2R/d\theta^2$

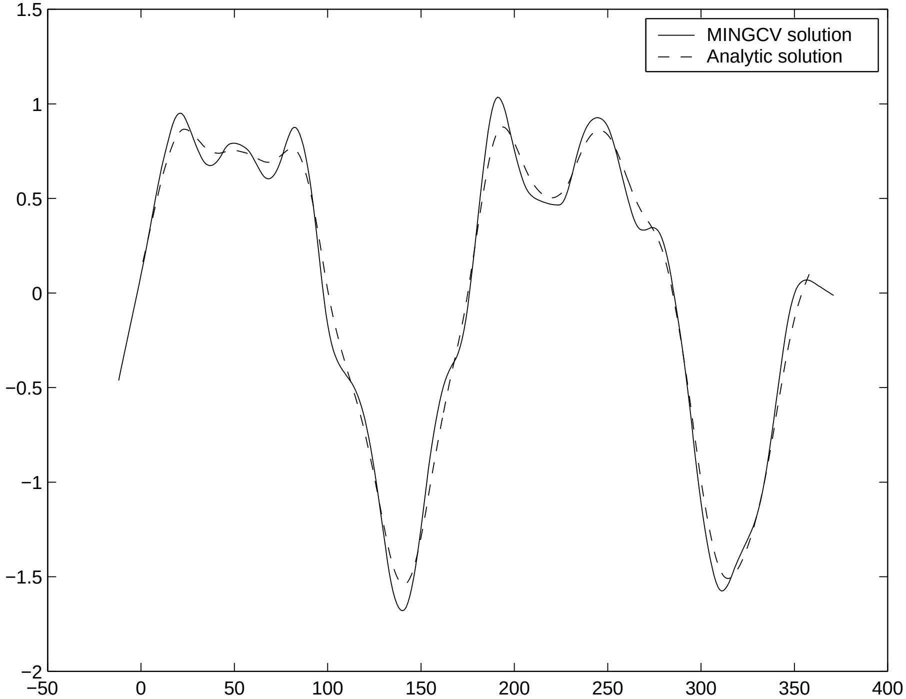

Figure 2: Solution estimate produced by the algorithm compared to the analytic solution.

Table 3: Comparison of the summary statistics for the analytic solution and the solution estimate obtained from the algorithm.

Analytic spline			MINGCV estimate		
GCV	R	Tr	GCV	R	Tr
0.221	14.75	82.1	0.226	12.64	75.1

Table 4: Performance of the algorithm on each grid, using a finite difference estimate of $d^2GCV/d\theta^2$.

l	h^l	No. of updates	FLOPS
5	32	23	50 400
4	16	8	35 100
3	8	7	60 000
2	4	16	249 900

and $d^2\mathrm{Tr}/d\theta^2$, almost halving the SOR workload required for each update. The finite difference procedure involved 'searching' empirically for a minimum GCV by iterating using the initial θ setting and periodically changing θ by 0.5 . When a minimum was detected across 3 consecutive GCV values, the finite difference formula

$$\frac{\frac{dGCV}{d\theta}(\theta_{m-1}) - \frac{dGCV}{d\theta}(\theta_{m+1})}{\theta_{m-1} - \theta_{m+1}} ,$$

was used to estimate $d^2\mathrm{GCV}/d\theta^2$, where θ_m is the θ estimate corresponding to the minimum GCV value.

The results of this procedure are shown in Table 4. As expected, the algorithm is almost twice as efficient, with only a few extra updates required due to the reduced accuracy of the $d^2\mathrm{GCV}/d\theta^2$ calculation. Clearly more updates are required while the algorithm searches for a minimum GCV, but this process is conducted on the coarse grid, where the extra updates are relatively inexpensive.

7 Conclusion

This unidimensional analysis has lead to the construction of an efficient algorithm for estimating minimum GCV univariate smoothing splines. The use of quadratic B-spline approximations on coarse grids allows accurate estimates of the minimum GCV solution to be obtained at low computational cost. Using finite difference approximation of the second derivative of the GCV, only 3 systems of equations need to be solved iteratively. Convergence was accelerated by using the estimate of $d\alpha/d\theta$ to correct the solution estimate after each θ update. These findings provide guidelines for the construction of a similar algorithm to estimate minimum GCV bivariate thin plate smoothing splines, constructed as tensor products of univariate biquadratic splines.

References

[1] W. L. Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, 88pp, 1987. C305

[2] R. K. Beatson and G. N. Newman. Fast evaluation of radial basis functions. I. *Comput. Math. Appl.* 24 7–19 1992. C292

[3] R. K. Beatson and M. J. D. Powell. An iterative method for thin plate spline interpolation that employs approximations to Lagrange functions. *Pitman Res. Notes Math. Ser.*, 303:17–39, 1994. C292

[4] R. K. Beatson and G. Goodsell and M. F. D. Powell. On multigrid techniques for thin plate spline interpolation in two dimensions. *The mathematics of numerical analysis (Park City, UT, 1995) (Providence RI), Lectures in Appl. Math.*, 32:77–97, 1996. C292

[5] C. de Boor. *A practical guide to splines*, Springer-Verlag Inc. 392, pp 1978. C293, C294, C296

[6] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979. C292

[7] D. R. Forsey and R. H. Bartels. Hierarchical B-Spline Refinement. *Computer Graphics*, 22(4):205–211, 1988. C295

[8] P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalised Linear Models*, Chapman and Hall, London 182 pp. 1994 C296

[9] D. Girad. A fast 'Monte-Carlo cross-validation' procedure for large least squares problems with noisy data. *Numerische Mathematik*, 56:1–23, 1989. C300

[10] M. Hegland and S. Roberts and I. Altas. Finite element thin plate splines for data mining applications. In: *Mathematical models for curves and surfaces II* editors M. Daehlen and T. Lyche and L. L. Schumaker, Vanderbilt University Press, Nashville TN, 245–252, 1998. Available as a Mathematical Research Report MRR057-97, School of Mathematical Sciences, The Australian National University 1998. C292

[11] M. F. Hutchinson. A new method for estimating the spatial distribution of mean seasonal and annual rainfall applied to the Hunter Valley. *Australian Meteorology Magazine*, 31:179–184, 1983. C292

[12] M. F. Hutchinson. A new method for gridding elevation and streamline data with automatic removal of spurious pits. *Journal of Hydrology*, 106:211–232, 1989. C292, C293

[13] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines.

Communications in Statistics — Simulation and Computation, 18:1059–1076, 1989. C292, C300, C306

[14] M. F. Hutchinson. On thin plate splines and kriging. *Computing and Science in Statistics*, 25:55–62, 1993. C293, C299

[15] M. F. Hutchinson. Interpolating mean rainfall using thin plate smoothing splines. *International Journal of Geographic Information Systems*, 9(4):385–403, 1995. C292

[16] M. F. Hutchinson. Optimising the degree of data smoothing for locally adaptive finite element bivariate smoothing splines. *ANZIAM Journal*, 42(E):C774–C796, 2000. http://anziamj.austms.org.au/v42/CTAC99/Hutch C292, C293, C298, C299, C300, C301

[17] M. F. Hutchinson and F. R. de Hoog. Smoothing Noisy Data with Spline Functions. *Numerische Mathematik* 47:99–106, 1985. C296

[18] M. F. Hutchinson and P. E. Gessler. Splines - more than just a smooth interpolator. *Geoderma*, 62:45–67, 1994. C305

[19] S. J. Jeffrey, J. O. Carter, K. B. Moodie and A. R Beswick. Using spatial interpolation to construct a comprehensive archive of Australian climate data. *Environmental Modelling and Software*, 16:309–330, 2001. C292

[20] M. Kohler. Universally Consistent Regression Function Estimation Using Hierarchical B-splines. *Journal of Multivariate Analysis*, 68:138–164, 1999. C294

[21] D. T. Price, D. W. Mckenney, I. A. Nalder, M. F. Hutchinson and J. L. Kesteven. A comparison of two statistical methods for spatial interpolation of Canadian monthly mean climate

data. *Agricultural and Forest Meteorology*, 101:81–94, 2000. C292

[22] I. J. Schoenberg. On spline functions. In: *Inequalities*, editor O. Shisha, Academic Press, New York 255–291 1967. C293

[23] X. Zheng and R. Basher. Thin plate smoothing spline modeling of spatial climate data and its application to mapping south pacific rainfalls. *Monthly Weather Review*, 123:3086–3120, 1995. C292