NEURAL COMPUTING
NCRG
RESEARCH GROUP

Neural Computing Research Group
Dept of Computer Science & Applied Mathematics
Aston University
Birmingham B4 7ET
United Kingdom
Tel: +44 (0)121 333 4631
Fax: +44 (0)121 333 4586
http://www.ncrg.aston.ac.uk/

Splines and Vector Splines

Dan Cornford
d.cornford@aston.ac.uk

Technical Report NCRG/97/024 March 13, 1998

1 Introduction

The thrust of this report concerns spline theory and some of the background to spline theory and follows the development in Wahba (1991). We also review methods for determining *hyper-parameters*, such as the smoothing parameter, by Generalised Cross Validation. Splines have an advantage over Gaussian process based procedures in that we can readily impose atmospherically sensible smoothness constraints and maintain computational efficiency. Equivalent methods are presented in (Amodei and Benbourhim, 1991) and (Wahba, 1982) by which we can penalise gradients of vorticity and divergence. Wahba (1982) provides a formulation in spherical coordinates (using the ideas of reproducing kernels) and Amodei and Benbourhim (1991) provide a Cartesian interpretation (couched in terms of Distributions - i.e. generalised functions). Both are summarised and improvements based on robust error functions and restricted numbers of basis functions given. A final, brief discussion of the application of vector splines to the problem of scatterometer data assimilation highlights the problems of ambiguous solutions.

2 Reproducing Kernels

Reproducing kernel Hilbert spaces define classes of functions that have the required properties for the solution of general spline smoothing problems where the penalty functional involves derivative of the fitted function. The data is assumed to come from the model:

$$z(x_i) = L_i f + \varepsilon_i \tag{1}$$

where the L_i is the evaluation functional (often $L_i f = f(x_i)$) allows a more general form for f. The noise model ε is assumed to be independently distributed (and usually Gaussian with variance σ_i^2). As we shall see later the Beppo-Levi space (Amodei and Benbourhim, 1991) is a special reproducing kernel Hilbert space with relevance to vector splines.

Having defined the properties of reproducing kernels (Wahba, 1991) we now need understand what form of kernel will be appropriate. Clearly we will be interested in functions involving derivatives and these functions will have to be continuous. An insight to one formulation of the spline problem comes from Taylors theorem with remainder which states:

$$f(x) = \left(\sum_{v=0}^{m-1} \frac{x^v}{v!} f^{(v)}(0) \right) + \left(\int_0^1 \frac{x - u_+^{m-1}}{(m-1)!} f^{(m)}(u) du \right)$$

where $f(x)$ is a real valued function on $[0,1]$ with $m-1$ continuous derivatives and $f^{(m)} \epsilon L_2[0,1]$, and (x_+) means x if $x \geq 0$ and 0 otherwise. If we now choose a class of functions B_m satisfying the boundary conditions $f^{(v)}(0) = 0$, $v = 0, ..., m-1$ then if $f \epsilon B_m$ we have:

$$f(x) = \left(\int_0^1 \frac{x - u_+^{m-1}}{(m-1)!} f^{(m)}(u) du \right) = \left(\int_0^1 G_m(x,u) f^{(m)}(u) du \right)$$

where $G_m(x,u)$ is the Green's function for the problem $D^m f = g$ with D^m the m^{th} derivative (Wahba, 1991).

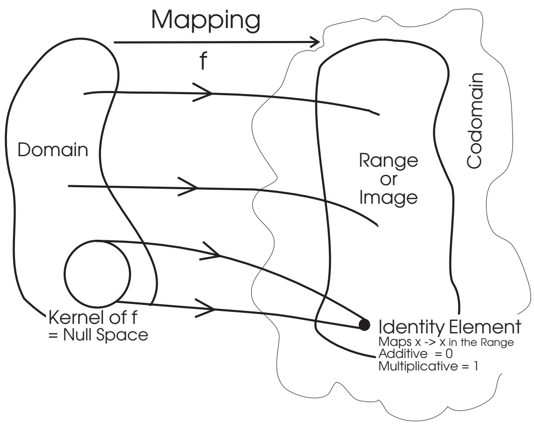

Figure 1: A reminder of the terminology of mappings and functions

If we now denote by W_m^0 the collection of all functions on [0,1] with $\{f : f \epsilon B_m, f, f', ..., f^{(m-1)}$ absolutely continuous, $f^{(m)} \epsilon L_2\}$ then W_m^0 is a Hilbert space with square norm $||f||^2 = \int_0^1 (f^{(m)}(x))^2 dx$. Furthermore it can be shown (Wahba, 1991) that W_m^0 is a reproducing kernel Hilbert space, with reproducing kernel:

$$R(x_i, x_j) = \int_0^1 G_m(x_i, u) G_m(x_j, u) du$$

It can be observed that the Hilbert space defined by W_m^0 does not contain the polynomials of degree less than $m-1$ - in other words they are in the null space. In most applications m is chosen to be small and thus low order polynomials are included in the solution.

Define $\phi_v(x) = x^{v-1}/(v-1)!$ for $v = 1, ..., m$ and denote the m-dimensional space of polynomials spanned by ϕ_v as H_0. This is a Hilbert space when given the squared norm $||\phi||^2 = \sum_{v=0}^{m-1} ((D^v \phi)(0))^2$ and has reproducing kernel:

$$R(x_i, x_j) = \sum_{v=1}^{m} \phi_v(x_i) \phi_v(x_j)$$

Now consider the Sobolev-Hilbert space W_m given by $\{f : f, f', ..., f^{(m-1)}$ absolutely continuous, $f^{(m)} \epsilon L_2\}$. This space includes a more general class of functions than W_m^0, however each element of W_m has a Taylor series expansion to order m and thus a unique decomposition $f = f_0 + f_1$, with $f_0 \epsilon H_0$ and $f_1 \epsilon W_m^0$. It can be shown that the two spaces H_0, W_m^0 are perpendicular and thus:

$$W_m = H_0 \oplus W_m^0$$

with the square norm $\|f\|^2 = \sum_{v=0}^{m-1}((D^v\phi)(0))^2 + \int_0^1 (f^{(m)}(x))^2 dx$. The reproducing kernel is the direct sum of the reproducing kernels of the perpendicular subspace and is thus:

$$R(x_i, x_j) = \sum_{v=1}^{m} \phi_v(x_i)\phi_v(x_j) + \int_0^1 G_m(x_i, u)G_m(x_j, u)du$$

This constructive definition of the spline interpolation problem is rather unusual in that we have said rather little about the problem we wish to address - that is of interpolation or approximation. However it will become clear that the smoothness penalty functional will simply be an orthogonal projection of the function f onto W_m^0 in W_m. Setting the problem in the space W_m as defined above allows us to formulate the spline problem relatively simply for general cases. This will be seen as we consider the 2 dimensional thin plate spline. Now we seek a solution to the general spline problem.

3 The General Spline Problem

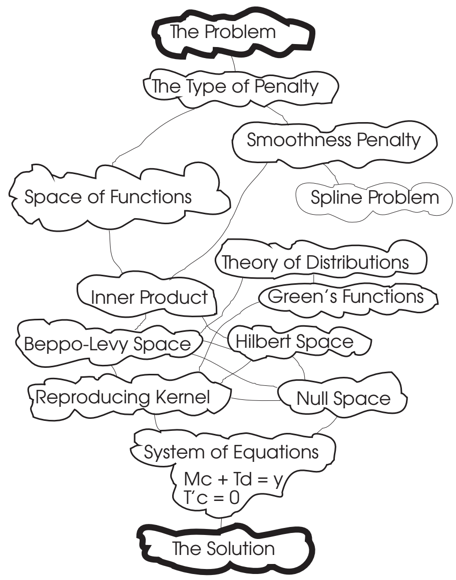

Figure 2: A summary of the links between the different elements in the spline smoothing / approximation problem.

Let the problem be again defined by the data model:

$$z_i = L_i f(x) + \varepsilon_i \qquad i = 1, ..., n$$

where $x \epsilon I_x$, I_x some index set with arbitrary range. The errors are assumed Gaussian iid. $f \epsilon H_R$, a given reproducing kernel Hilbert space, with decomposition:

$$H_R = H_0 \oplus H_1$$

with $dim(H_0) = M < n$. We find an estimate of f which minimises:

$$\frac{1}{n}\sum_{i=1}^{n}(L_i f - z_i)^2 + \lambda ||P_1 f||_R^2$$

where P_1 is the orthogonal projection of f onto H_1. With this type of penalty the reproducing kernel Hilbert space W_m is the natural setting.

Given a reproducing kernel it is always possible to obtain a representer η_i for any bounded linear functional L_i, that is:

$$\langle \eta_i, f\rangle = L_i f$$

and

$$\eta_i(x_j) = \langle \eta_i, R_{x_j}\rangle = L_i R_{x_j} = L_{i(.)} R_{(x_j,.)}$$

For example when $L_i f = f(x_i)$ then $\eta_i(x_j) = L_i R_{x_j} = R(x, x_j)|_{x=x_i}$ or if $L_i f = \frac{\partial^2 f(x_i)}{\partial x^2}$ then $\eta_i(x_j) = L_i R_{x_j} = \frac{\partial^2 R(x,x_j)|_{x=x_i}}{\partial x^2}$. The above definition is more general and allows more complex functionals L_i. We can now write the minimisation problem as:

$$\frac{1}{n}\sum_{i=1}^{n}(\langle \eta_i, f\rangle - z_i)^2 + \lambda ||P_1 f||_R^2 \tag{2}$$

If we again define a polynomial basis ϕ_i, $i = 1, ..., M$ to span the null space (kernel) of P_1 (i.e. H_0) and define $T_{i,j}$ as the n by M matrix of the j'th polynomial evaluated at the i'th data point. If T has full column rank (that is there is a non-degenerate least squares solution) then we can find the minimiser f_λ of (2) given by:

$$f_\lambda = \sum_{i=1}^{M} d_v \phi_v + \sum_{i=1}^{n} c_i \xi_i$$

The solution is given by:

$$\begin{aligned} \Sigma_s \boldsymbol{c} + T\boldsymbol{d} &= \boldsymbol{z} \\ T'\boldsymbol{c} &= 0 \end{aligned} \tag{3}$$

where:

$$\boldsymbol{d} = (d_1, \ldots, d_M)', \qquad \boldsymbol{c} = (c_1, \ldots, c_n)', \qquad \xi_i = P_1 \eta_i.$$

with $\Sigma_s = \Sigma + n\lambda I$, $\Sigma = \langle \xi_i, \xi_j\rangle$. Now we only need to determine ξ_i to solve the problem. (2) can then be written as:

$$\frac{1}{n}||\boldsymbol{z} - (\Sigma\boldsymbol{c} + T\boldsymbol{d})||^2 + \lambda \boldsymbol{c}'\Sigma\boldsymbol{c} \tag{4}$$

Recall that the H_r was defined as the sum of two orthogonal spaces, and thus the reproducing kernel of H_R is the sum of the reproducing kernels of the two subspaces. Then we have $\xi_i(x) = \langle \xi_i, R_x\rangle = \langle P_1\eta_i, R_x\rangle = \langle \eta_i, P_1 R_x\rangle = L_i R_x^1$ where R_x^1 is the representer of evaluation in H_1. Thus:

$$\langle \xi_i, \xi_j\rangle = L_i L_j R^1(x_i, x_j)$$

It can be seen that the ξ_i will be determined by the form of the penalty functional. In the next sections we will examine how.

4 Thin Plate Splines

Let us now define the data model to be:

$$z_i = L_i f(\boldsymbol{x}) + \varepsilon_i \qquad i = 1, ..., n \tag{5}$$

where $\boldsymbol{x} \in I_x$ is some index set with arbitrary range. In this application the index set I_x will be some small 'scene'over the ocean, with components either in Cartesian of spherical coordinates $\boldsymbol{x_i} = (x_i, y_i)$.

We will consider the 2 dimensional smoothing ('thin plate') spline. We seek a function f to minimise:

$$\frac{1}{n}\sum_{i=1}^{n}(L_i f - z_i)^2 + \lambda J_2(f) \tag{6}$$

with n observations (x_i, y_i, z_i), based on the model $z_i = f(x_i, y_i) + \varepsilon_i$ with the errors being independent of f and each other and with equal variance. L_i is a linear functional such that $L_i f \to f(x_i, y_i)$.

The smoothness penalty term $J_2(f)$ is given by:

$$J_2(f) = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \right) dx dy \tag{7}$$

Now define H to be a Hilbert Space of functions f such that $J_2(f)$ is finite. Choose some fixed (arbitrary) points $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M \epsilon \mathbb{R}^2$ such that

$$\sum_{v=1}^{M} a_v \phi_v(\boldsymbol{s}_j) = 0, \qquad j = 1, \ldots, M \Rightarrow a_v = 0$$

That is the ϕ_v are linearly independent. It will become clear later that these ϕ_v are simply the basis functions that span the null space of the smoothness penalty $J_2(f)$. An inner product is defined on H by:

$$\begin{aligned} \langle f, g \rangle = & \sum_{j=1}^{M} f(\boldsymbol{s}_j) g(\boldsymbol{s}_j) + \iint_{\mathbb{R}^2} \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 g}{\partial x^2} dx dy + \\ & + 2\iint_{\mathbb{R}^2} \frac{\partial^2 f}{\partial x \partial y}\frac{\partial^2 g}{\partial x \partial y} dx dy + \iint_{\mathbb{R}^2} \frac{\partial^2 f}{\partial y^2}\frac{\partial^2 g}{\partial y^2} dx dy \end{aligned} \tag{8}$$

and thus:

$$\| f \|^2 = \langle f, f \rangle = \sum_{j=1}^{M} f^2(\boldsymbol{s}_j) + J_2(f)$$

It can be seen that this space is similar to W_m as defined in the first section - that is it is a reproducing kernel Hilbert space. Again, using the Riesz representation theorem we associate with the linear functional L_i a representer $\eta_i \epsilon H$ such that:

$$L_i f = \langle \eta_i, f \rangle = f(x_i, y_i) = f(\boldsymbol{s}_i)$$

Now suppose we are given the η_i's. This implies we now want to minimise:

$$\frac{1}{n}\sum_{i=1}^{n}(\langle \eta_i, f \rangle - z_i)^2 + \lambda J_2(f) \tag{9}$$

Considering (9) we note that geometrically any f in the Hilbert Space H can be written as a linear combination of $\eta_1, \ldots, \eta_n, \phi_1, \ldots, \phi_M$ plus some function ρ which is perpendicular to each η_i and ϕ_v, that is:

$$f = \sum_{i=1}^{n} c_i \eta_i + \sum_{i=1}^{M} d_v \phi_v + \rho$$

for coefficients $\boldsymbol{c} = (c_1, \ldots, c_n)', \boldsymbol{d} = (d_1, \ldots, d_M)'$. Since we are interested in minimising (9), then ρ must be equal to zero as it is perpendicular to all the other terms in (9), but is otherwise arbitrary. We can also show that:

$$\langle \phi_v, \sum_{i=1}^{n} c_i \eta_i \rangle = 0, \qquad v = 1, \ldots, M \tag{10}$$

We still do not know what form the η_i's take, and to determine them we need to use the theory of reproducing kernel Hilbert spaces. Recall a Hilbert space H is said to possess a reproducing kernel if for all $\boldsymbol{x} \epsilon \mathbb{R}^2$ the functional $Lf = f(\boldsymbol{x})$ is a continuous linear functional. If this condition is met the there exists a representer $q_{\boldsymbol{x}} \epsilon H$ such that:

$$Lf = f(\boldsymbol{x}) = \langle q_{\boldsymbol{x}}, f \rangle$$

We define the function $Q(\boldsymbol{x_i}, \boldsymbol{x_j})$ by:

$$Q(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle q_{\boldsymbol{x_i}}, q_{\boldsymbol{x_j}} \rangle, \qquad \boldsymbol{x_i}, \boldsymbol{x_j} \epsilon \mathbb{R}^2$$

Q is called the reproducing kernel of H. The basic property of the reproducing kernel is that given Q one can find representers of any continuous functionals acting on elements of H. Thus the representers η_i are given by:

$$\eta_i(\boldsymbol{x}) = L_{i(\boldsymbol{x_j})} Q(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{11}$$

where $L_{i(\boldsymbol{x_j})}$ means the functional L is applied to what follows considered as a function of $\boldsymbol{x_j}$. Also:

$$\langle \eta_i, \eta_j \rangle = L_{i(\boldsymbol{x_i})} L_{j(\boldsymbol{x_j})} Q(\boldsymbol{x_i}, \boldsymbol{x_j})$$

Knowing the form of the inner product (8) it is possible to deduce the form of the reproducing kernel for certain special cases. These kernels are solutions to particular Green's functions - thus have a fundamental solution:

$$Q(\boldsymbol{x_i}, \boldsymbol{x_j}) = K(\boldsymbol{x_i}, \boldsymbol{x_j}) + P(\boldsymbol{x_i}, \boldsymbol{x_j})$$

where:

$$\begin{aligned} K(\boldsymbol{x_i}, \boldsymbol{x_j}) = E_m(\boldsymbol{x_i}, \boldsymbol{x_j}) &- \sum_{v=1}^{M} p_v(\boldsymbol{x_j}) E_m(\boldsymbol{x_{i_v}}, \boldsymbol{x_i}) - \\ &- \sum_{u=1}^{M} p_u(\boldsymbol{x_i}) E_m(\boldsymbol{x_j}, \boldsymbol{x_{i_u}}) \\ &+ \sum_{u,v=1}^{M} p_u(\boldsymbol{x_i}) p_v(\boldsymbol{x_j}) E_m(\boldsymbol{x_{i_u}}, \boldsymbol{x_{i_v}}) \end{aligned}$$

$$P(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{v=1}^{M} p_v(\boldsymbol{x_i}) p_v(\boldsymbol{x_j})$$

Here the $p_1, \ldots, p_M$ are M polynomials of degree less than 2 satisfying $p_v(\boldsymbol{x_{i_u}}) = 1$ when $u = v$ and equal to zero otherwise. $m = 2$ implies:

$$E_2(\boldsymbol{x_i}, \boldsymbol{x_j}) = E_2(\|\boldsymbol{x_i} - \boldsymbol{x_j}\|) = E_2(\boldsymbol{r}) = \theta \boldsymbol{r}^2 ln(\boldsymbol{r}) \tag{12}$$

where $\boldsymbol{r}$ is equal to the (Euclidean) distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ and

$$\theta = \frac{1}{8\pi}$$

from (Wahba, 1991). Using (10) and (11) we can show that:

$$\sum_{i=1}^{n} c_i \eta_i(\boldsymbol{x_j}) = \sum_{i=1}^{n} c_i \xi_i(\boldsymbol{x_j}) - \sum_{i=1}^{n} \sum_{v=1}^{M} c_i \xi_i(\boldsymbol{x_v}) p_v(\boldsymbol{x_j})$$

Since the double sum on the right is a polynomial of less than degree 2, and since these are in the null space of $J_2(f)$ the minimiser is:

$$f_{n,M,\lambda}(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \xi_i(\boldsymbol{x}) + \sum_{i=1}^{M} d_v \phi_v(\boldsymbol{x}) \tag{13}$$

where M = 3 $\phi_1 = \mathbf{1}$, $\phi_2 = \boldsymbol{x}$ and $\phi_3 = \boldsymbol{y}$.

What we have so far examined is the decomposition of the solution to the spline problem from one function space into 2 subspaces - defined by the range and the null space of the penalty functional. Because these spaces are orthogonal we simply add the solutions from the two subspaces to yield the final solution. This makes the problem tractable, however it is very difficult to see how one might alter the penalty functional to encompass some of the wind fields we might expect, for instance those which contai fronts (Cornford, 1997b).

5 Generalised Cross Validation

Generalised Cross Validation (GCV) is a method used to estimate the optimal value of a *hyper-parameter* such as the smoothing parameter of the thin plate spline λ on the basis of data. The idea is to determine the smoothing parameter which produces the smallest root mean squared error at data points which are withheld one at a time. The technique is useful because we can use mathematical tricks to compute the GCV quickly, which do not involve repetitive fitting of the splines.

If we consider equation (13) then we seek minimiser f_λ of the standard thin plate smoothing spline problem. But how do we set λ? One method is to use GCV. We want to find a λ to minimise:

$$V(\lambda) \propto \frac{1}{n} \sum_{k=1}^{n} (z_k - L_k f_\lambda)^2$$

where z_k is the k^{th} data value and $L_k f_\lambda$ evaluates f_λ at the point $\boldsymbol{x}_k$ (without using z_k to determine the spline parameters).

[Actually what we really are interested in is:

$$T(\lambda) \propto \frac{1}{n} \sum_{k=1}^{n} (L_k f_\lambda - L_k f)^2$$

the predictive mean square error. Wahba (1991) gives a 'weak GCV theorem' which says that there exists a sequence (as $n \to 0$) of minimisers $\hat{\lambda}$ of $E[V(\lambda)]$ that comes close to achieving $\min_\lambda E[T(\lambda)]$. See Wahba (1991), section (4.3) for more details.]

We actually compute:

$$V(\lambda) = \frac{\frac{1}{n}\sum_{k=1}^{n}(z_k - L_k f_\lambda)^2}{(1-\mu_1(\lambda))^2} = \frac{\frac{1}{n}\|(I - A(\lambda))\,\boldsymbol{z}\|^2}{\left(\frac{1}{n}\mathrm{tr}\,(I - A(\lambda))\right)^2} \tag{14}$$

where $A(\lambda)$ is the *influence matrix* given by:

$$\begin{bmatrix} L_1 f_\lambda \\ \vdots \\ L_n f_\lambda \end{bmatrix} = A(\lambda)\boldsymbol{z} = Td + \Sigma_s c \tag{15}$$

We can compute $A(\lambda)$ using a QR decomposition of T given by:

$$T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where Q_1 is $n \times M$, Q_2 is $n \times (n - M)$ and the square matrix $Q = [Q_1 \ Q_2]$ is orthogonal. R is upper triangular. Now:

$$I - A(\lambda) = n\lambda Q_2 (Q_2' \Sigma_s Q_2)^{-1} Q_2'$$

Bates *et al.* (1987) give an efficient algorithm for the computation of the GCV $V(\lambda)$ using the same QR decomposition, a Cholesky decomposition and a singular value decomposition, which also makes computation of the solution to the smoothing spline very efficient.

6 Splines and Random Field Models

There is a strong connection between spline techniques and random field models (Cornford, 1997a) - and this has been frequently explored in the geostatistics literature (Laslett, 1994). The duality can be seen by considering the the Hilbert space spanned by a random field and the associated reproducing kernel Hilbert space.

We start with a random field $Z(\boldsymbol{x})$, $\boldsymbol{x} \epsilon I_x \subset \mathbb{R}^2$ with $E[Z(\boldsymbol{x_i})Z(\boldsymbol{x_j})] = cov_Z[(\boldsymbol{x_i}, \boldsymbol{x_j})] = R(\boldsymbol{x_i}, \boldsymbol{x_j})$. We let H be the Hilbert space spanned by $Z(\boldsymbol{x})$ as defined in Parzen (1961). This Hilbert space is the collection of all random variables of the form:

$$Z_H = \sum_k a_k Z(\boldsymbol{x_k})$$

with $a_k \epsilon \mathbb{R}$, $\boldsymbol{x_k} \epsilon I_x$ and possesses an inner product defined by $\langle Z_1, Z_2 \rangle = E[Z_1 Z_2]$ where Z is understood to mean Z_H. To complete this Hilbert space H we also need the quadratic mean limits, that is $Z \epsilon H$ iff there is a sequence Z_l, $l = 1, 2, ...$ of random variables as defined above such that $lim_{l\to\infty} E[(Z - Z_l)^2] = \|Z - Z_l\|^2 \to 0$. If we now define H_R to be a reproducing kernel Hilbert space with reproducing kernel R as defined above then H_R has a one to one, inner product preserving correspondence with H, since

$$\langle Z(\boldsymbol{x_i}), Z(\boldsymbol{x_j}) \rangle = E[Z(\boldsymbol{x_i})Z(\boldsymbol{x_j})] = R(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle R_{\boldsymbol{x_i}}, R_{\boldsymbol{x_j}} \rangle$$

Optimisation of splines can be shown to correspond to Bayesian estimation on a random field. If we assume that $X(\boldsymbol{x})$, $\boldsymbol{x} \epsilon I_x$ is a stationary (zero-mean) Gaussian process with $E[Z(\boldsymbol{x_i})Z(\boldsymbol{x_j})] =$

$R(\boldsymbol{x_i}, \boldsymbol{x_j})$ as before. If we are now given $Z(\boldsymbol{x_i}) = z_i$ at points $\boldsymbol{x_i}$, $i = 1, \dots, n$ then for a fixed $\boldsymbol{x}$, $E[Z(\boldsymbol{x})|Z(\boldsymbol{x_i})]$, the kriging estimator (the best linear unbiased estimator - in a minimum variance sense) is given by:

$$\hat{f}(\boldsymbol{x}) = [R(\boldsymbol{x}, \boldsymbol{x_1}), \dots, R(\boldsymbol{x}, \boldsymbol{x_n})]R_{nn}^{-1}\boldsymbol{z}$$

where R_{nn} is the square, $n \times n$ covariance matrix $R_{nn}(\boldsymbol{x_i}, \boldsymbol{x_j})$ and $\boldsymbol{z}$ is a vector of the observed values. This is also true for the case when the process is not Gaussian - but the Gaussian assumption allows us to link the conditional expectation with $\hat{f}$.

If we now consider the problem set in the reproducing kernel Hilbert space H_R, then we must find some $f \epsilon H_R$ which minimises $||f||^2$ (recall the process is zero mean - hence this will correspond to minimum variance) subject to $f(\boldsymbol{x_i}) = z_i$. Thus the solution must be of the form:

$$f = \sum_{j=1}^{n} c_j R_{\boldsymbol{x_j}}$$

where this time the null space of the penalty functional is the empty set. Thus $||f||^2 = \boldsymbol{c}' R_{nn} \boldsymbol{c}$ with R_{nn} as previously. At the points $\boldsymbol{x_i}$ we have $\sum_{j=1}^{n} c_j R_{\boldsymbol{x_j}}(\boldsymbol{x_i}) = z_i$, or $R_{nn}\boldsymbol{c} = \boldsymbol{z}$ and f is given by:

$$f = \boldsymbol{z}' R_{nn}^{-1} [R_{\boldsymbol{x_1}}, \dots, R_{\boldsymbol{x_n}}]' \tag{16}$$

This is exactly equal to $\hat{f}$ and thus the penalty functional approach (using reproducing kernel Hilbert spaces) can be seen to be equivalent to the kriging approach. Of course we have considered a rather restricted class of problems here - with the assumption of stationarity being particularly troublesome in practice. Wahba (1991) claims that one cannot consistently estimate the parameters of the (generalised) covariance function (from the theory of Intrinsic Random Functions of order k (Matheron, 1973)) from data thus a spline prior is just as good unless one has prior knowledge on these parameters.

7 Vector Splines

One possible method for the inclusion of some smoothness constraint on the wind field is through the use of vector splines, which are reviewed in this section. We will be concerned with the estimation (smoothing) of a vector valued function (that is wind!). Cornford (1997b) gives a decomposition of the wind vector into two (hopefully uncoupled) scalars (Helmholtz theorem). In practice these scalars, the stream function and the velocity potential, are not totally uncoupled. Both Wahba (1982) and Amodei and Benbourhim (1991) give similar versions of vector splines specifically for winds.

7.1 Wahba's Vector Splines

Wahba (1982) gives a more flexible framework with the splines being fit over the surface of a sphere (the Earth!!). The penalty functional minimised is:

$$J_m(\Psi, \Phi) = \int \left((\nabla^2)^{\frac{m}{2}} \Psi(\boldsymbol{p})\right)^2 d\boldsymbol{p} + \frac{1}{\delta} \int \left((\nabla^2)^{\frac{m}{2}} \Phi(\boldsymbol{p})\right)^2 d\boldsymbol{p} \tag{17}$$

with the usual sum of (vector) squares error term. The actual computation is performed using an expansion in the eigen-functions of the Laplacian evaluated over the sphere (spherical harmonics).

They seek a solution of the form:

$$\Psi = \sum_{l=1}^{N} \sum_{s=-l}^{l} \alpha_{ls} Y_l^s \qquad \Phi = \sum_{l=1}^{N} \sum_{s=-l}^{l} \beta_{ls} Y_l^s \tag{18}$$

Reordering the indices using:

$$\tilde{N} = \sum_{l=1}^{N} \sum_{s=-l}^{l} 1 \tag{19}$$

and assuming observations (u_i, v_i), $i = 1, \ldots, n$, setting the $n \times \tilde{N}$ matrix X_ϕ with $(i, ls)^{th}$ entry:

$$X_\phi(i, ls) = \frac{1}{A} \frac{\partial}{\partial \phi} Y_l^s(\boldsymbol{p}_i) \tag{20}$$

where A is the radius of the earth, $\boldsymbol{p}_i$ is the location of the i^{th} observation and (ϕ, λ) are the latitude and longitude (recall we are working on the surface of a sphere) and Y_l^s are the normalised spherical harmonics (Wahba, 1982), together with:

$$X_\lambda(i, ls) = \frac{1}{A\cos(\phi_i)} \frac{\partial}{\partial \lambda} Y_l^s(\boldsymbol{p}_i) \tag{21}$$

we can create the $2n \times 2\tilde{N}$ matrix:

$$X = \begin{bmatrix} -X_\phi & X_\lambda \\ X_\lambda & X_\phi \end{bmatrix} \tag{22}$$

which is like the matrix of low order polynomials used in thin plate splines. It is clear that the matrix gives estimates of (u, v) based on fitting to (Φ, Ψ). Under this model the penalty matrix for (17) has a particularly simple form. Let D_i be the $\tilde{N} \times \tilde{N}$ diagonal matrix with $(ls, ls)^{th}$ entry $\lambda_{ls}(i)$ for $i = 1, 2$ with:

$$\lambda_{ls}(i) = (l(l+1))^m \qquad i = 1, 2 \tag{23}$$

Write:

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & \delta D_2 \end{bmatrix} \tag{24}$$

then with $\boldsymbol{u} = (u_1, \ldots, u_n, v_1, \ldots, v_n)'$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{\tilde{N}}, \beta_1, \ldots, \beta_{\tilde{N}})'$ we need to find $\boldsymbol{\alpha}$ to minimise:

$$\frac{1}{n} \|\boldsymbol{u} - X\boldsymbol{\alpha}\|^2 + \lambda_s \boldsymbol{\alpha}' D^{-1} \boldsymbol{\alpha} \tag{25}$$

Note that the formulation of the problem ensures that it is not necessary to consider the null space terms (particularly the orthogonality condition) as in thin plate splines, since these are automatically part of the solution, through the use of spherical harmonics. There remain two free parameters; δ which controls the ratio of vorticity to divergence and λ_s, which controls the degree of smoothing, both of which can be set from the data using Generalised Cross Validation, or fixed using historical data. Wahba (1982) suggests that the $\lambda_{ls}(i)$ could be chosen from historical data, by analysing the power spectrum (of winds or other fields) and fitting these with some form of model - not entirely unlike the method of moments used to fit variograms in random field models (Cornford, 1997a). With $m = 2$ these correspond to the div-curl splines of (Amodei and Benbourhim, 1991).

7.2 Amodei and Benbourhim's Vector Splines

Amodei and Benbourhim (1991) take a slightly different approach, in that the problem domain is posed in a Cartesian framework using the theory of Distributions (generalised functions). We shall now briefly review the method of Amodei and Benbourhim (1991), noting those areas where the so called div-curl splines may not meet our needs. A function is sought to minimise:

$$J_{\alpha,\beta}(\boldsymbol{u}) = \alpha \int \|\nabla(\nabla \cdot \boldsymbol{u})\|^2 dxdy + \beta \int \|\nabla(\nabla \times \boldsymbol{u})\|^2 dxdy \tag{26}$$

together with the interpolating constraints $\boldsymbol{u}(\boldsymbol{x_i}) = \boldsymbol{u_i}$ for $i = 1, \ldots, n$, where $\boldsymbol{u}$ is the wind vector (u, v). Thus the spline can be seen to penalise gradients in divergence and vorticity the relative importance of which are fixed by the choice of α and β. These can be chosen on the basis of what we already know about the behaviour of the winds (a prior in other words) or estimated from the data using cross validation. For our problem with noisy targets it might be unwise to try and choose both (or even either) the degree of smoothing and the ratio of divergence / vorticity from the data. In practice one chooses $\alpha \in [0, 1]$ and defines $\beta = 1 - \alpha$, with a smoothing parameter λ acting to weight the penalty function $J_{\alpha,\beta}$ as in the usual spline case.

The solution to the above problem is sought in the so called Beppo-Levy space of order 2. This is a (generalised) function space over $\mathbb{R}$ of Distributions (generalised functions) whose second (Distributional) derivatives are square integrable. The solution to this problem takes a form very similar to that of the thin-plate spline problem:

$$\begin{aligned}
u_{\alpha,\beta}(\boldsymbol{x}) &= \sum_{i=1}^{n} a_i \left(\frac{1}{\alpha} \frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x}_i)}{\partial x^2} + \frac{1}{\beta} \frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x}_i)}{\partial y^2} \right) \\
&\quad + \sum_{i=1}^{n} b_i \left(\frac{1}{\alpha} - \frac{1}{\beta} \right) \frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x}_i)}{\partial x \partial y} + \sum_{i=0}^{2} c_i \phi_i(\boldsymbol{x}) \\
v_{\alpha,\beta}(\boldsymbol{x}) &= \sum_{i=1}^{n} b_i \left(\frac{1}{\alpha} \frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x}_i)}{\partial y^2} + \frac{1}{\beta} \frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x}_i)}{\partial x^2} \right) \\
&\quad + \sum_{i=1}^{n} a_i \left(\frac{1}{\alpha} - \frac{1}{\beta} \right) \frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x}_i)}{\partial x \partial y} + \sum_{i=0}^{3} d_i \phi_{i-3}(\boldsymbol{x})
\end{aligned} \tag{27}$$

where ϕ_i are the polynomials of degree one or less as before. This solution can be seen to take the same form as the others outlined in previous sections. K is the reproducing kernel for the Beppo-Levy space defined above. It takes the form $K(\boldsymbol{r}) = \theta\|\boldsymbol{r}\|^4 ln\|\boldsymbol{r}\|$, with $\theta = -(\frac{1}{2})^7\pi$ and $\|\boldsymbol{r}\| = (x^2 + y^2)^{1/2}$. This can be derived from the fact that K is the fundamental solution of $(\nabla^2)^3 K = \delta$ with δ the Kronecker delta (this is where the Green's functions come in). The above equations together with the constraints:

$$\sum_{i=1}^{n} a_i \phi(\boldsymbol{x}_i) = \sum_{i=1}^{n} b_i \phi(\boldsymbol{x}_i) = 0 \tag{28}$$

for all ϕ as polynomials of degrees one or less define a system of linear equations:

$$\begin{bmatrix} \frac{1}{\alpha}K_{(xx)} + \frac{1}{\beta}K_{(yy)} & \frac{1}{\alpha}K_{(xy)} - \frac{1}{\beta}K_{(xy)} & \Phi & 0 \\ \frac{1}{\alpha}K_{(xy)} - \frac{1}{\beta}K_{(xy)} & \frac{1}{\alpha}K_{(yy)} + \frac{1}{\beta}K_{(xx)} & 0 & \Phi \\ \Phi' & 0 & 0 & 0 \\ 0 & \Phi' & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{29}$$

where $K_{(xx)}$, $K_{(xy)}$ and $K_{(yy)}$ are $n \times n$ matrices with entries:

$$\begin{aligned}
K_{(xx)i,j} &= \left.\frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x_i})}{\partial x^2}\right|_{\boldsymbol{x}=\boldsymbol{x_j}} = \theta\left(12\delta x_{i,j}^2 + 4\delta y_{i,j}^2\right)\log\left(\boldsymbol{r}_{i,j}\right) + 7\delta x_{i,j}^2 + \delta y_{i,j}^2 \\
K_{(xy)i,j} &= \left.\frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x_i})}{\partial xy}\right|_{\boldsymbol{x}=\boldsymbol{x_j}} = 8\theta\delta x_{i,j}\delta y_{i,j}\log\left(\boldsymbol{r}_{i,j}\right) + 6\delta x_{i,j}\delta y_{i,j} \\
K_{(yy)i,j} &= \left.\frac{\partial^2 K(\boldsymbol{x}-\boldsymbol{x_i})}{\partial y^2}\right|_{\boldsymbol{x}=\boldsymbol{x_j}} = \theta\left(4\delta x_{i,j}^2 + 12\delta y_{i,j}^2\right)\log\left(\boldsymbol{r}_{i,j}\right) + \delta x_{i,j}^2 + 7\delta y_{i,j}^2
\end{aligned}$$

with $\delta x_{i,j}$ meaning $x_i - x_j$ and other terms similarly. Φ is the $n \times 3$ matrix with vector components $[\mathbf{1}, \boldsymbol{x}, \boldsymbol{y}]$ at the data points while $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ are the parameter vectors and $\boldsymbol{u}, \boldsymbol{v}$ the data vectors. 0 is used to signify a zero array of appropriate dimension, while $\mathbf{0}$ is a zero vector, as usual.

The linear equations are solved in MATLAB using the \ operator. If the aim is to approximate rather than interpolate (as will be the case in our application) then it is necessary to add some constant which depends on the noise level (or degree of smoothing desired) to the diagonal of the covariance matrix. Denoting:

$$\Sigma = \begin{bmatrix} \frac{1}{\alpha}K_{(xx)} + \frac{1}{\beta}K_{(yy)} & \frac{1}{\alpha}K_{(xy)} - \frac{1}{\beta}K_{(xy)} \\ \frac{1}{\alpha}K_{(xy)} - \frac{1}{\beta}K_{(xy)} & \frac{1}{\alpha}K_{(yy)} + \frac{1}{\beta}K_{(xx)} \end{bmatrix} \tag{30}$$

then $\Sigma_s = \Sigma + n\lambda I$ where λ is the smoothing parameter of Wahba (1991) and I is the usual diagonal identity matrix with the same dimensions as Σ . Thus one simply replaces the Σ in (29) with Σ_s and solves as usual.

It is likely that vector splines will be flexible enough for many situations. The ratio of divergence on vorticity in the atmosphere typically varies with latitude (due to the rotation of the earth) with divergence dominating at low latitudes (tropics) and vorticity becoming more important pole ward (Cornford, 1997b). Thus we could set α and β as functions of latitude from climatological data, where α and β are estimated from cross-validation - although this has the disadvantage that we will necessarily be reflecting any model bias, since models would generate the climatology.

7.3 Div-curl splines using robust error functions

Instead of solving the system of linear equations using matrix inversions, the div-curl spline error function is minimised using a gradient based minimisation algorithm. This is necessary because the robust error function used means that the system of equations is no longer linear in parameters. The data fit part of the error function is given by:

$$V(t) = \beta_1 t^2 - ln\left(1 + \frac{\epsilon}{1-\epsilon}\sqrt{\frac{\beta_2}{\beta_1}}e^{t^2(\beta_1-\beta_2)}\right) \tag{31}$$

where t is the miss-fit between the predicted and observed values at the data points (Girosi, 1991). Thus with a spline which has a solution of the form $\Sigma_s[\boldsymbol{a};\boldsymbol{b}] + T[\boldsymbol{c};\boldsymbol{d}]$ (where Σ_s is a 'covariance' type matrix, derived from the non-local spline basis functions and T is the polynomial part which lies in the null space of the smoothness constraint) to the following minimisation problem:

$$E[(\boldsymbol{a};\boldsymbol{b}),(\boldsymbol{c};\boldsymbol{d})] = \sum_{i=1}^{n}\left(V\left((\boldsymbol{u_i},\boldsymbol{v_i}) - \Sigma_{s,i}(\boldsymbol{a};\boldsymbol{b}) - T_i(\boldsymbol{c};\boldsymbol{d})\right)\right) + \lambda(\boldsymbol{a};\boldsymbol{b})'\Sigma_s(\boldsymbol{a};\boldsymbol{b}) \tag{32}$$

together with the constraint $T'[\boldsymbol{a};\boldsymbol{b}] = 0$. When $V(x) = x^2$ then the solution is given by a system of linear equations which are best solved using matrix manipulation (in MATLAB) - that is the

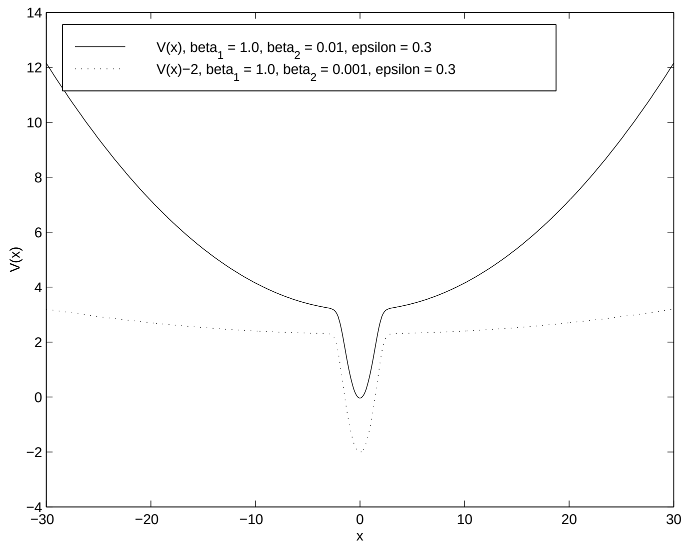

Figure 3: Plot of the robust error function

standard div-curl spline. Where we use more complex error functions (such as Equation 31) a matrix based minimisation is not possible. To solve this minimisation problem one needs to compute the derivatives of the error function with respect to the parameters and impose the constraint. The constraint may either be imposed using a penalty function approach (which penalises $T'[\boldsymbol{a};\boldsymbol{b}]$ in the solution in such a way that $T'[\boldsymbol{a};\boldsymbol{b}] \rightarrow 0$) or Lagrange multipliers [ref? - Ian's notes].

The error function (31) has a shaped illustrated in Figure 3. It can be seen that as β_2 gets smaller in relation to β_1, the error function becomes flatter away from a small region whose width is defined by β_1. The error function (especially with small β_2 values) essentially 'ignores' those errors greater than a certain threshold, since for these larger errors there is a much smaller gradient of the error function. Thus large outliers will have little effect on the final solution - but this makes the result rather sensitive to the initial conditions. It may be possible to get around this with sensible initialisations using parameters from a non-robust div-curl spline for example.

The advantage of this error function is that we can fit a sensible spline to the data we have using a smoothing parameter that is quite small, and performing the operation only once. In the version I have implemented I have used scaled conjugate gradient descent for the first 50 iterations and then a succession of quasi-Newton calls for succeeding iterations. These iterations are required since I have used a penalty function approach to ensure that $T'[\boldsymbol{a};\boldsymbol{b}] = 0$.

7.4 Regularisation networks

We may find that while div-curl splines are elegant, the placement of the kernel centres at the observations may be suboptimal and the option to choose fewer centres than we have observations will be advantageous to the quality of the results and the speed of computation - especially in the case where the centres are placed using some quick method prior to optimisation (for example evenly spaced over the problem domain, or placed using k-means algorithms). We are thus getting rather close to the area of Radial Basis Function networks or Regularisation Networks (Girosi *et al.*, 1995).

This has also been implemented. The user is free to choose the number of basis functions (derived from the div-curl penalty) and the centres of these are set using a k-means algorithm so that they are reasonably spaced with respect to the data locations. In general, even for the most complex wind fields less than 10 basis functions are required, and preliminary results suggest 3 may be sufficient for most cases. By fixing the centres we retain the simplicity of the linear solution and get around trying to set the smoothing parameter which is anyway rather difficult to determine. We are left with the problem of the number and placement of basis functions.

7.5 Dealing with Ambiguous Solutions

So far we have assumed that we have selected the best vector from the model output (that is the model or observation system that generates the initial wind vectors) and that this unique value will be used to determine the spline parameters. In the context of scatterometer data assimilation we must include at least 2 and possibly four solutions when estimating the full wind field, since we are not able to retrieve unique wind vectors due to the intrinsic multi-modality of the solution (Thiria *et al.*, 1993; Offiler, 1994). One method to try may be a random combination model such that we try all possible permutations of the first 2 solutions at each of the 361 points in the scene and pick the permutation with the smallest total penalty. This would involve fitting 2^{361} splines a somewhat large number and not currently computationally feasible. Of course even then we may get the 180 degree alias solution so we may well need some heuristics to choose which sense the flow should be (based on climatology).

Since what we actually produce is $P(\boldsymbol{u}$, other methods could be envisaged such as fitting the spline to the n_s most probable vectors (where $n_s << n$ the total number of observations) and then choosing the new vectors to add on the basis of the solution that best agrees with the current solution (although this would tend to propagate any errors in the initial n_s points. There is no principled basis for choosing n_S. In general using this ad-hoc approach will lead to a series of heuristics with little statistical basis to their use - although this has been the dominant means of treating the ambiguity problem to date. A review of the use of various ambiguity removal algorithms is not appropriate here and will be given elsewhere, as will a overview of the problem in the framework of scatterometer data assimilation.

8 Conclusion

This report has expanded on some of the ideas on which spline methods are based (particularly from the reproducing kernel perspective (Wahba, 1991)). The theory is useful to aid understanding, but not vital to the application, of any spline. Vector splines were introduced in two formulations, one over the sphere (Wahba, 1982) and the other in Cartesian space (Amodei and Benbourhim, 1991). Both are equivalent and it was shown that both could be modified by the addition of a robust error function, or a restriction on the number of basis functions used. Restricting the number of basis functions used places the technique into the Regularisation network framework of Girosi *et al.* (1995). The application of spline based techniques to scatterometer data was touched upon, but will be dealt with more fully elsewhere.

9 Acknowledgements

I would like to thank Chris Williams and Ian Nabney for constructive comments on this report. David Barber helped me understand the relationship between the reproducing kernel and the solution to the Green's problem. The work has been carried out as part of the European Union funded NEUROSAT program.

References

Amodei, L. and M. N. Benbourhim 1991. A vector spline approximation. *Journal of Approximation Theory* **67**, 51–79.

Bates, D. M., M. J. Lindstrom, G. Wahba, and B. S. Yandell 1987. Gcvpack - routines for generalised cross validation. *Communications in Statistics, Part B - Simulation and Computation* **16**, 263–297.

Cornford, D. 1997a. Random field models and priors on wind. Technical Report NCRG/97/023, Neural Computing Research Group, Aston University, Aston Triangle, Birmingham, UK.

Cornford, D. 1997b. Surface wind fields (on earth). Technical Report NCRG/97/022, Neural Computing Research Group, Aston University, Aston Triangle, Birmingham, UK.

Girosi, F. 1991. Models of noise and robust estimates. AI Memo 1287, Artifical Intelligence Laboratory, Massachusetts Institute of Technology.

Girosi, F., M. Jones, and T. Poggio 1995. Regularisation theory and neural network architectures. *Neural Computation* **7**, 219–269.

Laslett, G. M. 1994. Kriging and splines: An empirical comparison of their performance in some applications. *Journal of the American Statistical Society* **89**, 391–400.

Matheron, G. 1973. The intrinsic random functions and their applications. *Advances in Applied Probability* **5**, 439–468.

Offiler, D. 1994. The calibration of ers-1 satellite scatterometer winds. *Journal of Atmospheric and Oceanic Technology* **11**, 1002–1017.

Parzen, E. 1961. An approach to time series analysis. *Annals of MAthematical Statistics* **32**, 951–989.

Thiria, S., C. Mejia, F. Badran, and M. Crepon 1993. A neural network approach for modeling nonlinear transfer functions: Application for wind retrieval from spaceborne scatterometer data. *Journal of Geophysical Research* **98**, 22827–22841.

Wahba, G. 1982. Vector splines on the sphere, with application to the estimation of vorticity and divergence from discrete, noisy data. In W. Schempp and K. Zeller (Eds.), *Multivariate Approximation Theory, Vol. 2*, pp. 407–429. Basel: Birkhauser Verlag.

Wahba, G. 1991. *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics.