NPL Report CMSC 32/03

Report to the National Measurement System Directorate, Department of Trade and Industry

From the Software Support for Metrology Programme

Multivariate empirical models and their use in metrology

By
R Boudjemaa, A B Forbes and P M Harris
Centre of Mathematics and Scientific Computing, NPL
S Langdell, Numerical Algorithms Group Ltd

December 2003

Multivariate empirical models and their use in metrology

R Boudjemaa, A B Forbes and P M Harris
Centre of Mathematics and Scientific Computing, NPL
S Langdell, Numerical Algorithms Group Ltd

December 2003

ABSTRACT

In this report, we consider classes of the empirical functions available to the metrologist to model multivariate data and discuss the algorithmic requirements for using these models in data approximation. We first review common examples of empirical functions of one variable and describe their generic features relevant to models in higher dimensions. We then review common approaches to modelling multivariate data including approaches specific to data on a regular grid (e.g., tensor product polynomial and splines) and more general approaches (e.g., radial basis functions and support vector machines). For each type of model we highlight their advantages and disadvantages with respect to their applications in modelling metrological data. We also give an example application involving interferometric data. We conclude that i) radial basis functions are likely to become important tools for modelling multivariate systems in metrology, ii) much of the underlying technology of support vector machines – statistical information theory, reproducing kernel Hilbert spaces, etc., – are of potential value to metrology, particularly in situations in which the system under study is imperfectly understood, for example, in biotechnology.

ISSN 1471–0005

Authorised by Dr Dave Rayner,
Head of the Centre for Mathematics and Scientific Computing

National Physical Laboratory,
Queens Road, Teddington, Middlesex, United Kingdom TW11 0LW

Contents

1 Introduction

Mathematical modelling, in general, involves the assignment of mathematical terms for all the relevant components of a (measurement) system and the derivation of equations giving the relationships between these mathematical terms. In these equations, we can distinguish between terms that relate to quantities that are known or measured and those that are unknown or to be determined from measurement data. We will in general call the former terms *model variables* and use $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$, $\mathbf{y}$, etc., to denote them and the latter *model parameters* and denote them by $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, $\mathbf{b}$, etc.

A *physical model* is one in which there is a theory that defines how the variables and parameters depend on each other.

An *empirical model* is one in which a relationship between the variables is expected or observed but with no supporting theory. Many models have both empirical and physical components.

1.1 Empirical models in metrology

In this section, we give a few examples of the use of empirical models in metrology.

1.1.1 Response surfaces

A common use of univariate empirical models in metrology is the determination of calibration response curves associated with a measurement system. Typically the measurement system will produce a response y corresponding to a stimulus variable x and we wish to model y as function of x, for example length as a function of temperature. Clearly many systems depend on more than one variable and we want to be able to model the response of such systems as a function of all the relevant variables.

1.1.2 Properties of materials

Most measurement systems have some dependence on the properties of the material from which they are constructed or with which they interact. For example, the use of interferometric transducers for length measurement requires knowledge of the refractive index of the supporting medium. The

refractive index of air, for instance, depends on temperature, pressure, humidity and percentage of carbon dioxide.

1.1.3 Fields

Many quantities in metrology are represented as scalar fields (temperature, pressure) or vector fields (electric, magnetic, velocity) as functions of two or three spatial coordinates.

1.2 Report overview

In this report, we consider classes of the empirical functions available to the metrologist to model multivariate data and discuss the algorithmic requirements for using these models in data approximation. In section 2, we review common examples of empirical functions of one variable and in section 3 we describe their generic features that are relevant to models in higher dimensions. In sections 4–8 we review approaches to modelling multivariate data including approaches specific to data on a regular grid (tensor product polynomial and splines, for example) and more general approaches (e.g., radial basis functions). For each main type of model we highlight their advantages and disadvantages with respect to their applications in modelling metrological data. We give an example application involving interferometric data in section 9. Our summary and concluding remarks are presented in section 10.

2 Empirical functions of one variable

In this section, we give an overview of some of the main classes of empirical functions of one variable. This is done firstly to review some of the generic features of empirical models and secondly as input into the development of multivariate empirical models.

2.1 Polynomial curves

Polynomials provide a class of linear models that are used extensively as empirical models for experimental data. A polynomial of degree n can be written as

$$\phi(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = \sum_{j=0}^{n} a_j x^j = \sum_{j=0}^{n} a_j \phi_j(x),$$

where $\phi_j(x) = x^j$ are the *monomial basis* functions. A polynomial of degree 1 is a straight line, degree 2 a quadratic curve, etc. The immediate appeal of polynomials is that computation with polynomials requires only the arithmetic operations of addition and multiplication.

While the description of polynomials in terms of the monomial basis functions makes clear the nature of polynomial functions, the use of the monomial basis in numerical computation leads to severe numerical difficulties. A first difficulty is that for values of the variable x significantly greater than one in absolute value, the terms x^j will become very large as j increases. This problem is solved by working with a normalised variable $\hat{x}$. If x varies within the range $[x_{\min}, x_{\max}] = \{x : x_{\min} \leq x \leq x_{\max}\}$, then

$$\hat{x} = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}}, \tag{1}$$

and all its powers lie in the range $[-1, 1]$. For small degree polynomials ($n \leq 4$, say), this normalisation is sufficient to avoid most numerical difficulties.

The second difficulty arises from the fact that, especially for large j, the basis function ϕ_j looks very similar to ϕ_{j+2} in the range $[-1, 1]$. Figure 1 presents the graphs of $\phi_{2j}(x) = x^{2j}$, $j = 1, 2, 3, 4$. We can regard polynomial functions defined on $[-1, 1]$ as members of a vector space of functions. In this vector space, the inner product of two polynomials $p(x)$ and $q(x)$ can be defined in terms of an integral of the form

$$\langle p, q \rangle_w = \int_{-1}^{1} p(x) q(x) w(x) dx,$$

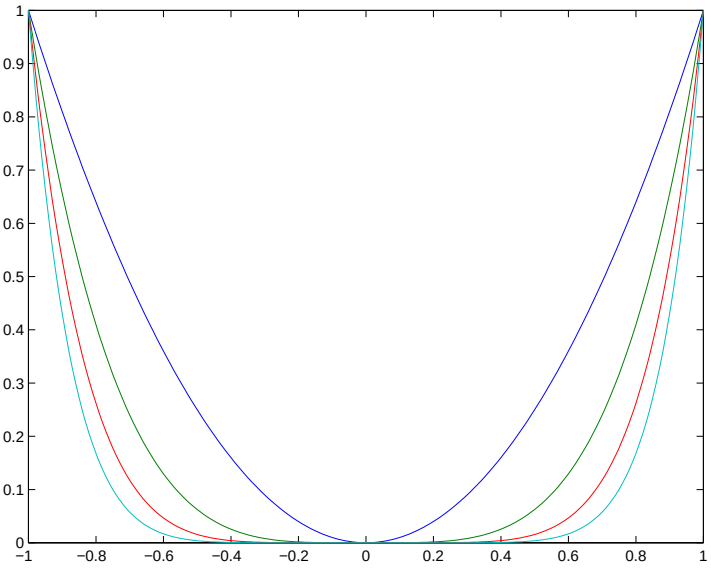

Figure 1: Graphs of x^{2j} $j = 1, 2, 3, 4$ on the interval [-1,1].

where $w(x)$ is a weighting function. Setting $\|p\|_w = \langle p, p\rangle_w^2$, the angle $\theta_{p,q}$ between p and q is defined from

$$\cos\theta_{p,q} = \frac{\langle p, q\rangle_w}{\|p\|_w \|q\|_w}.$$

With this definition of angle, it is straightforward to show that the monomial basis functions ϕ_j and ϕ_{j+2} point in roughly the same direction (in the sense that the angle between them is small), leading to ill-conditioning. This ill-conditioning worsens rapidly as the degree increases.

The problem of poor choice of basis functions can be illustrated for 3-dimensional Euclidean space. Suppose we take as basis vectors for three dimensional space $\mathbb{R}^3$ the vectors $\mathbf{e}_1 = (1, 0, 0)^{\mathrm{T}}$, $\mathbf{e}_2 = (1, 0.001, 0)^{\mathrm{T}}$ and $\mathbf{e}_3 = (1, 0, 0.001)^{\mathrm{T}}$. We note that all three vectors point approximately in the direction of the positive x-axis. Any point $\mathbf{y}$ in $\mathbb{R}^3$ can be written as a linear combination

$$\mathbf{y} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3.$$

For example,

$$\begin{aligned}(0.0, 1.0, 1.0)^{\mathrm{T}} &= -2000\mathbf{e}_1 + 1000\mathbf{e}_2 + 1000\mathbf{e}_3,\\ (0.0, 1.1, 1.1)^{T} &= -2200\mathbf{e}_1 + 1100\mathbf{e}_2 + 1100\mathbf{e}_3,\end{aligned}$$

showing that a change of the order of 0.1 in the point $\mathbf{y}$ requires a change of order 100 in the coefficients $\mathbf{a}$. This type of ill-conditioning means that up

to three significant figures of accuracy could be lost when using these basis vectors in numerical computation.

Alternative representations of polynomials can be derived by finding polynomial basis functions with better properties. The space $\mathcal{P}_n$ is an $n+1$-dimensional Euclidean space defined by the basis functions $\{\phi_j \in \mathcal{P}_n\}$ with an inner product. We can take this basis and produce from it a new set $\{T_j(x)\}$ of basis functions which span the same space but are orthogonal with respect to the inner product:

$$\int_{-1}^{1} T_j(x)T_k(x)w(x)dx = 1, \text{ if } j = k, \ 0 \text{ otherwise.}$$

(In the example of $\mathbb{R}^3$, this process would produce the orthogonal vectors $(1,0,0)^{\mathrm{T}}$, $(0,1,0)^{\mathrm{T}}$ and $(0,0,1)^{\mathrm{T}}$ from the basis vectors $\mathbf{e}_k$, $k = 1,2,3$.) $T_j(x)$ is a polynomial of degree j and, moreover, there is a three-term recurrence relationship of the form

$$T_j(x) = (A_{j-1}x + B_{j-1})T_{j-1}(x) - C_{j-1}T_{j-2}(x),$$

which can be used to evaluate the basis functions stably and efficiently [31].

The *Chebyshev* polynomials (of the first kind) $T_j(x)$ are one such set of basis functions and have the property that they are orthogonal to each other on the interval $[-1,\ 1]$ with respect to the weighting function $w(x) = 1/(1+x^2)^{1/2}$. They are defined by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad j \geq 2.$$

Figure 2 presents the graphs of $T_j(x)$, $j = 2, \ldots, 5$. Conventionally, T_0 is replaced by $T_0/2$ in the basis.

Using orthogonal polynomials in conjunction with the variable transformation formula (1) it is possible to use high degree polynomial models over any interval in a numerically stable way [25, 47].

Other classical orthogonal polynomials defined on $[-1,1]$ include the Legengre polynomials, $w(x) = 1$, the Chebyshev polynomials of the second kind, $w(x) = (1-x^2)^{1/2}$, and the Jacobi polynomials, $w(x) = (1-x)^{\alpha}(1-x)^{\beta}$, $\alpha, \beta > -1$. The Laguerre polynomials are defined over the interval $[0,\infty)$ and are orthogonal with respect to $w(x) = e^{-x}$. The Hermite polynomials are defined over the interval $(-\infty,\infty)$ and are orthogonal with respect to $w(x) = e^{-x^2}$.

There are other numerical approaches to representing polynomials for polynomial regression. Given data $\{(x_i, y_i)\}_1^m$ and weights $\mathbf{w} = (w_1, \ldots, w_m)^{\mathrm{T}}$,

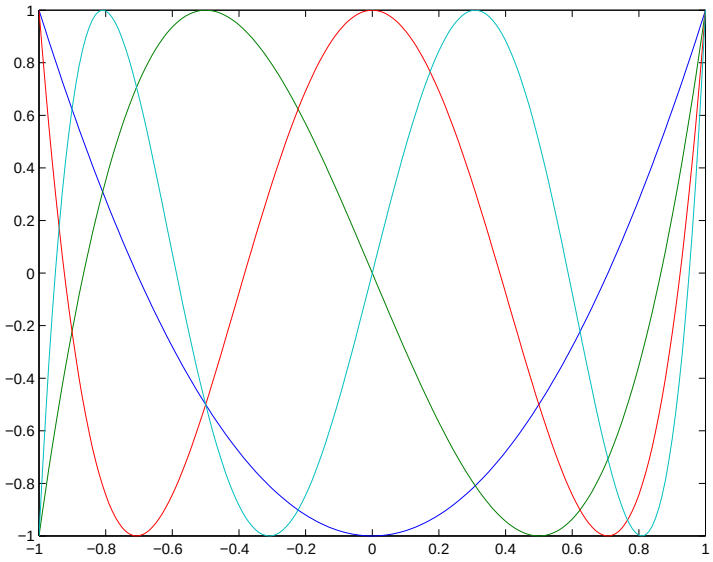

Figure 2: Chebyshev polynomials $T_j(x)$, $j = 2, \ldots, 5$.

the Forsythe method [25] implicitly determines a set of basis functions $\phi_j(x)$ that are orthogonal with respect to the inner product defined by

$$\langle f, g \rangle_{\mathbf{w}} = \sum_{i=1}^{m} w_i f(x_i) g(x_i).$$

The orthogonality of the basis functions can be exploited fully in determining the least squares best-fit polynomial to the data. The set of orthogonal polynomials is constructed specifically for the data $\{x_i\}$ and $\{w_i\}$. By contrast, the Chebyshev polynomials are more versatile since they are defined in the same way for all data sets.

As mentioned earlier, polynomials used over ranges different from [-1, 1] are defined in terms of a transformed variable $\hat{x}$ which depends on the range constants $x_{\min}$ and $x_{\max}$. In order to emphasize this dependency we sometimes write $\phi(x, \mathbf{a}|x_{\min}, x_{\max})$.

Let $\mathcal{P}_n$ be the collection of polynomial curves of degree n. Then $\mathcal{P}_0 \subset \mathcal{P}_1 \subset \ldots$ form a nested sequence of model spaces: if $\phi \in \mathcal{P}_n$, then $\phi \in \mathcal{P}_{n+1}$.

2.2 Polynomial spline curves

Like polynomials, polynomial spline curves - splines for short - are a class of linear models widely used for modelling discrete data. A spline $s(x)$ of order n defined over an interval $[x_{\min}, x_{\max}]$ is composed of sections of polynomial curves $p_k(x)$ of degree $n-1$ joined together at fixed points $\{\lambda_k\}_1^N$ in the interval. Practically all calculations using spline functions are performed using a B-spline representation of the form

$$s(x, \mathbf{a}|\boldsymbol{\lambda}) = \sum_{j=1}^{n+N} a_j N_{n,j}(x|\boldsymbol{\lambda}),$$

where n is the order of the spline, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ is the interior knot set satisfying

$$x_{\min} = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N < \lambda_{N+1} = x_{\max},$$

and $N_{n,j}(x, \boldsymbol{\lambda})$ are the B-spline basis functions of order n (i.e., degree $n-1$).

The basis functions $N_{n,j}(x|\boldsymbol{\lambda})$ are specified by the interior knot set $\boldsymbol{\lambda} = \{\lambda_k\}_1^N$, range limits

$$x_{\min} = \lambda_0, \quad \text{and} \quad x_{\max} = \lambda_{N+1},$$

and the additional exterior knots, λ_j, $j = 1-n, \ldots, -1$ and $j = N+2, \ldots, N+n$. These exterior knots are usually assigned to be

$$\lambda_j = \begin{cases} x_{\min}, & j < 0, \\ x_{\max}, & j > N+1. \end{cases}$$

With this choice, the basis functions are defined by the knots $\boldsymbol{\lambda}$. We use $q = n + N$ to denote the number of basis functions.

A common choice of order is $n = 4$, splines constructed from cubic polynomials – *cubic splines*. Figure 3 graphs a B-spline basis function for a cubic spline defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$. Figure 4 graphs all eight $= (n + N)$ basis functions for this knot set.

If the interior knots are distinct, a spline of order n has continuous derivatives of order $n-2$. So for example, a cubic spline (order 4) with distinct knots has continuous first and second derivatives.

A major feature of spline approximation algorithms is the exploitation of the banded structure of the observation matrices that arise. For a spline of order n, the only non-zero elements in each row occur in n adjacent columns and this sparsity structure can be fully exploited in the QR factorisation stage of the solution of the linear least squares system (section 3.2).

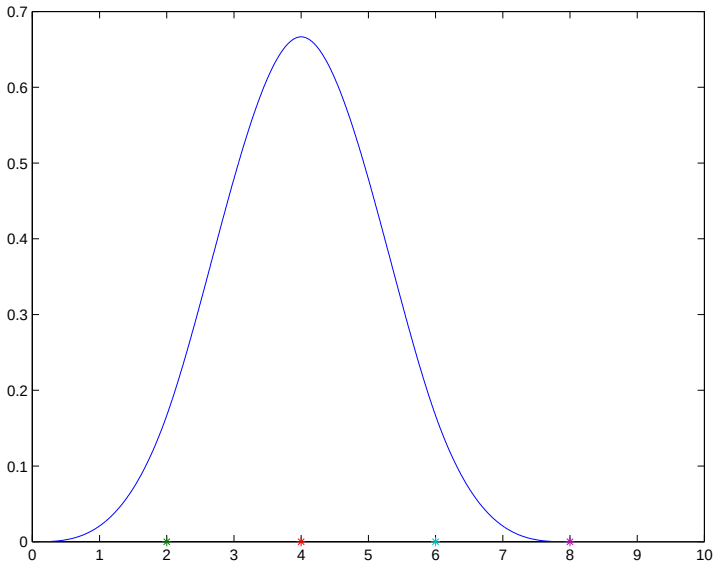

Figure 3: B-spline basis function $N_{4,4}(x, \boldsymbol{\lambda})$ defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$.

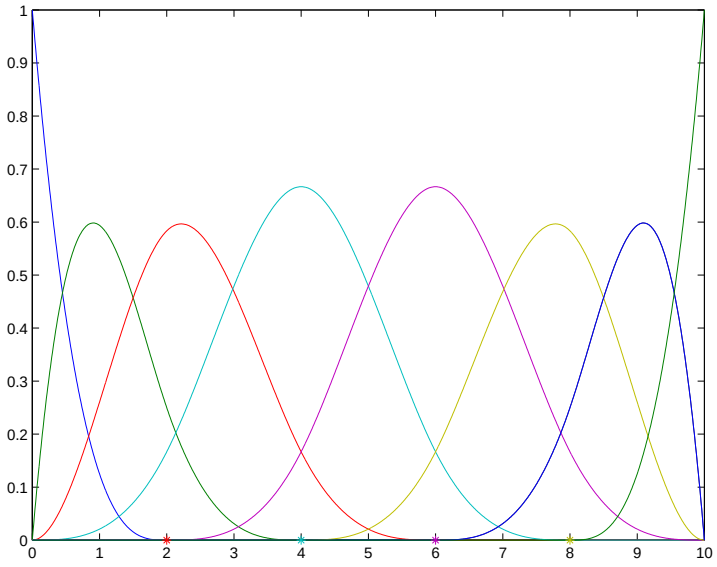

Figure 4: B-spline basis functions $N_{4,j}(x, \boldsymbol{\lambda})$ defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$.

Let $\mathcal{S}_{n|\boldsymbol{\lambda}}$ be the space of polynomial spline curves series of order n and knot set $\boldsymbol{\lambda}$. Then

$$\mathcal{S}_{n|\boldsymbol{\lambda}} \subset \mathcal{S}_{p|\boldsymbol{\lambda}}, \quad p > n, \quad \mathcal{S}_{n|\boldsymbol{\lambda}} \subset \mathcal{S}_{n|\boldsymbol{\mu}}, \quad \boldsymbol{\lambda} \subset \boldsymbol{\mu}.$$

2.3 Fourier series

A Fourier series of degree n is generally written as

$$\phi(x, \mathbf{a}) = \frac{a_0}{2} + \sum_{j=1}^{n} a_j \cos jx + \sum_{j=1}^{n} b_j \sin jx,$$

where $\mathbf{a} = (a_0, a_1, \ldots, a_n, b_1, \ldots, b_n)^{\mathrm{T}}$. We note the $\phi(x + 2\pi, \mathbf{a}) = \phi(x, \mathbf{a})$. To model functions with period $2L$, we modify the above to

$$\phi(x, \mathbf{a}|L) = a_0/2 + \sum_{j=1}^{n} a_j \cos j\pi x/L + \sum_{j=1}^{n} b_j \sin j\pi x/L.$$

Since

$$\int_{-\pi}^{\pi} \cos jx \cos kx \, dx = \int_{-\pi}^{\pi} \sin jx \sin kx \, dx = 0, \quad j \neq k,$$

and

$$\int_{-\pi}^{\pi} \cos jx \sin kx \, dx = \int_{-\pi}^{\pi} \cos jx \, dx = \int_{-\pi}^{\pi} \sin jx \, dx = 0,$$

the basis functions 1, $\cos jx$ and $\sin jx$ are orthogonal with respect to the unit weighting function over any interval of length 2π.

Let $\mathcal{F}_{n|L}$ be the space of Fourier series with period $2L$. Then

$$\mathcal{F}_{n|L} \subset \mathcal{F}_{p|L}, \quad p > n, \quad \mathcal{F}_{n|L} \subset \mathcal{F}_{n,L/K}, \quad K = 2, 3, \ldots.$$

2.4 Sums of exponentials with fixed time constants

A sum of exponentials with fixed time constants $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^{\mathrm{T}}$, $\lambda_j > \lambda_{j+1}$ is of the form

$$\phi(x, \mathbf{a}|\boldsymbol{\lambda}) = \sum_{i=1}^{n} a_j e^{\lambda_j x}.$$

Let $\mathcal{E}_{n|\boldsymbol{\lambda}}$ be the space of sums of exponentials with time constants $\boldsymbol{\lambda}$. Then

$$\mathcal{E}_{n|\boldsymbol{\lambda}} \subset \mathcal{E}_{p|\boldsymbol{\mu}}, \quad p > n \text{ and } \boldsymbol{\lambda} \subset \boldsymbol{\mu}.$$

2.5 Ordinary differential equations and orthogonal functions

Many sets of orthogonal functions of one variable arise as the solutions of ordinary differential equations (ODEs). We know from linear algebra that a symmetric $n \times n$ matrix A can be factored as

$$A = V \Lambda V^{\mathrm{T}},$$

its *eigen-decomposition*, where Λ is a diagonal matrix with $\Lambda_{jj} = \lambda_j$ and V is an $n \times n$ orthogonal matrix with $V^{\mathrm{T}}V = VV^{\mathrm{T}} = I$, the $n \times n$ identity matrix. The λ_j are known as eigenvalues and the columns of V the eigenvectors $\mathbf{v}_j$ and are such that $A\mathbf{v}_j = \lambda_j \mathbf{v}_j \;\; j = 1, \ldots, n$. Regarding differentiation as a linear mapping (or operator) on vector spaces of functions it is possible to generalise the concepts of eigenvectors and values, and show that for some classes of differential equations, their solutions generate a sequence of eigenfunctions orthogonal with respect to some function inner product.

2.5.1 Example: Sturm-Liouville problems

A Sturm-Liouville problem is a second order ordinary differential equation of the form

$$[u(x)\phi(x)']' + [v(x) + \lambda w(x)]\phi(x) = 0,$$

with boundary conditions

$$\begin{aligned} \alpha_1\phi(a) + \alpha_2\phi'(a) &= 0, \quad \alpha_1^2 + \alpha_2^2 > 0, \\ \beta_1\phi(b) + \beta_2\phi'(b) &= 0, \quad \beta_1^2 + \beta_2^2 > 0. \end{aligned}$$

Under mild restrictions on the functions u, v and assuming $w(x) > 0$, $x \in [a,\ b]$, then it can be shown there is an infinite number of values of λ_j (eigenvalues), all real, which give rise to nonzero solutions ϕ_j (eigenfunctions) and that the eigenfunctions are orthogonal with respect to the weighting function w on the interval $[a,\ b]$, i.e.,

$$\int_a^b \phi_j(x)\phi_l(x)w(x)\,dx = 0, \quad j \neq l.$$

3 Basic features of empirical models

The main concern of this report is empirical models of two or more variables. In particular, we are concerned with surfaces of the form

$$z = \phi(x, y, \mathbf{a}),$$

with z defined as a function of variables x and y or, more generally, hypersurfaces of the form

$$z = \phi(\mathbf{x}, \mathbf{a}), \quad \mathbf{x} \in \mathbb{R}^{p-1},$$

which defines a surface in $\mathbb{R}^p$.

In this section, we discuss the basic features of empirical models as illustrated by the empirical models of one variable (section 2) and the desirable properties we would like empirical models of two or more variables to have.

3.1 Basis functions

Empirical models are usually defined as linear combinations of basis functions

$$\phi(\mathbf{x}, \mathbf{a}) = \sum_{j=1}^{n} a_j \phi_j(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^{p-1}, \quad \mathbf{a} \in \mathbb{R}^n,$$

where Ω is the domain of definition of the basis functions. Let $\mathcal{F}_n$ be the collection of all such linear combinations. Then $\mathcal{F}_n$ is a vector space: if $\phi, \psi \in \mathcal{F}_n$ and $\alpha, \beta \in \mathbb{R}$, then

$$(\alpha\phi + \beta\psi)(\mathbf{x}) = \alpha\phi(\mathbf{x}) + \beta\psi(\mathbf{x})$$

is also in $\mathcal{F}_n$. The mapping $\mathbf{a} \mapsto \phi(\mathbf{x}, \mathbf{a})$ is a linear mapping from $\mathbb{R}^n$ to $\mathcal{F}_n$.

As a space of functions, we can associate to $\mathcal{F}_n$ an inner product of the form

$$\langle \phi, \psi \rangle_{\Omega,w} = \int_{\Omega} \phi(\mathbf{x})\psi(\mathbf{x})w(\mathbf{x})\, d\mathbf{x},$$

where $w(\mathbf{x}) \geq 0$ is a weighting function defined over Ω, and corresponding norm

$$\|\phi\| = \langle \phi, \phi \rangle_{\Omega,w}^{1/2}.$$

A set of n functions ϕ_j, $j = 1, \ldots, n$ form an *orthogonal set* if $\langle \phi_j, \phi_k \rangle = 0$, $j \neq k$. If, in addition $\|\phi_j\| = 1$, they are said to form an *orthonormal* set.

Given an orthonormal set ϕ_j for $\mathcal{F}_n$ and any $\psi \in \mathcal{F}_n$, then

$$\psi(\mathbf{x}) = \sum_{j=1}^{n} a_j \phi_j(\mathbf{x}), \quad a_j = \langle \psi, \phi_j \rangle.$$

The mapping $\mathbf{a} \mapsto \sum_j a_j \phi_j(\mathbf{x})$ is then an isomorphism between inner product spaces $\mathbb{R}^n$ and $\mathcal{F}_n$:

$$\left\langle \sum_j a_j \phi_j(\mathbf{x}), \sum_j b_j \phi_j(\mathbf{x}) \right\rangle_{\Omega,w} = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_j a_j b_j.$$

Any set of n linearly independent functions in $\mathcal{F}_n$ can be orthonormalised using the *Gramm-Schmidt* process [29] (not always in a numerically stable way).

Given a set of data points $X = \{\mathbf{x}_i \in \Omega, i = 1, \ldots, m\}$ and weights $w_i \geq 0$, $i = 1, \ldots, m$, we can also form a discrete inner product

$$\langle \phi, \psi \rangle_{X,\mathbf{w}} = \sum_{i=1}^{m} \phi(\mathbf{x}_i)\psi(\mathbf{x}_i)w_i.$$

3.2 Least-squares approximation

Suppose we wish to approximate a set of data points $X = \{(\mathbf{x}_i, z_i) \in \mathbb{R}^{p-1} \times \mathbb{R}\}$, $i = 1, \ldots, m$, by an empirical function of the form $z = \phi(\mathbf{x}, \mathbf{a}) = \sum_j a_j \phi_j(\mathbf{x})$. The observation matrix Φ associated with X and ϕ is

$$\Phi_{ij} = \phi_j(\mathbf{x}_i).$$

The least-squares best-fit[1] of the model ϕ are determined by the parameters $\mathbf{a}$ that solve

$$\min_{\mathbf{a}} f(\mathbf{a}) = \|\mathbf{z} - \Phi\mathbf{a}\|^2. \tag{2}$$

At the solution, it is known that the partial derivatives of f with respect to the parameters are zero, i.e.,

$$\frac{\partial f}{\partial a_j} = 0, \quad j = 1, \ldots, n,$$

and this leads to the system of linear equations of order n,

$$\Phi^{\mathrm{T}}\Phi\mathbf{a} = \Phi^{\mathrm{T}}\mathbf{z},$$

known as the *normal equations*. If Φ is full rank[2] so that $\Phi^{\mathrm{T}}\Phi$ is invertible, the solution parameters are given (mathematically) by

$$\mathbf{a} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{T}\mathbf{y}.$$

[1] We can also consider weighted least-squares problems; see, e.g., [18]

[2] If Φ is rank deficient, the singular value decomposition (SVD, [29]) can be used to provide a least-squares solution. Alternatively, regularisation techniques can be used to construct a full rank problem [48].

If Φ has orthogonal factorisation [18, 29]

$$\Phi = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

then, using the fact that $\|Q\mathbf{x}\| = \|\mathbf{x}\|$, we have

$$\|\mathbf{z} - \Phi\mathbf{a}\| = \|Q^{\mathrm{T}}\mathbf{z} - Q^{\mathrm{T}}\Phi\mathbf{a}\| = \left\| \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} - \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \mathbf{a} \right\|,$$

where $\mathbf{q}_1$ is the first n and $\mathbf{q}_2$ the last $m - n$ elements of $Q^{\mathrm{T}}\mathbf{z}$. From this it is seen that $\|\mathbf{z} - \Phi\mathbf{a}\|^2$ is minimised if $\mathbf{a}$ solves the upper triangular system

$$R_1\mathbf{a} = \mathbf{q}_1.$$

The numerical accuracy of the solution to (2) will depend on i) the accuracy to which the elements $\Phi_{ij} = \phi_j(\mathbf{x}_i)$ are evaluated, and ii) the condition of the matrix Φ.

If the basis functions ϕ_j are orthonormal with respect to the discrete inner product $\langle\ ,\ \rangle_X$ then Φ is an orthogonal matrix with $\Phi^{\mathrm{T}}\Phi = I$, so that the solution is given by $\mathbf{a} = \Phi^{\mathrm{T}}\mathbf{z}$. If the basis functions ϕ_j are orthonormal with respect to the continuous inner product $\langle\ ,\ \rangle_{\Omega,w}$ and the density of the data points $\mathbf{x}_i$ is approximately proportional to w, then Φ is expected to be approximately orthogonal and therefore well conditioned.

3.3 Subsidiary parameters

Often the basis functions depend on subsidiary parameters which we will denote by $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_s)^{\mathrm{T}}$. We denote this dependence, if required, as $\phi_j(\mathbf{x}) = \phi_j(\mathbf{x}|\boldsymbol{\lambda})$. An example of a set of subsidiary parameters is the vector of knots associated with a spline.

3.4 Partial ordering within families of empirical models

Given model spaces $\mathcal{F}_{n_1|\boldsymbol{\lambda}_1}$ and $\mathcal{F}_{n_2|\boldsymbol{\lambda}_2}$ from the same family of models, there are usually a set of straightforward conditions on n_k and $\boldsymbol{\lambda}_k$, $k = 1, 2$, to ensure that

$$\mathcal{F}_{n_1|\boldsymbol{\lambda}_1} \subset \mathcal{F}_{n_2|\boldsymbol{\lambda}_2}.$$

With this property, we can generate a sequence of model spaces

$$\mathcal{F}_{n_1|\boldsymbol{\lambda}_1} \subset \mathcal{F}_{n_2|\boldsymbol{\lambda}_2} \subset \ldots \mathcal{F}_{n_q|\boldsymbol{\lambda}_q} \subset \ldots$$

of increasing dimension. This is an important property since empirical models are used in situations where the degree of complexity of the system is not known *a priori* so we want a range of model spaces to try out and from this range select one which seems most appropriate. The selection of model from a range of models is discussed in [9, 16].

If there exists an orthogonal series of basis functions ϕ_i, $j = 1, 2, \ldots$, then defining $\mathcal{F}_n$ as the vector space of linear combinations of the first n basis functions produces a corresponding sequence

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \ldots$$

This is the ideal situation. The orthogonality of basis functions means that $\mathcal{F}_{n+1}$ is capable of describing behaviour totally absent from $\mathcal{F}_n$ in the sense that $\langle \phi_{n+1}, \psi \rangle = 0$ for all $\psi \in \mathcal{F}_n$.

Polynomials and Fourier series are examples where there is an obvious nested sequence of model spaces of increasing dimension.

The relationship between spline model spaces is more complicated since the splines depend on the choice of knots. An important issue in working with splines is to determine a sensible strategy for knot placement, see, e.g., [19].

3.5 Requirements for empirical models

For empirical models in general, we would like:

- Basis functions that are straightforward to evaluate in a numerically stable way.
- Well-conditioned observation matrices, ideally orthogonal with respect to some inner product.
- Straightforward strategies to determine suitable values of any subsidiary parameters.
- Nested sequences of model spaces allowing the user to balance flexibility of behaviour with economy of representation.

4 Tensor product surfaces

The simplest way to generate linear empirical models for surfaces is to construct them from linear empirical models for curves. Suppose

$$\begin{aligned} \phi(x, \mathbf{a}) &= a_1\phi_1(x) + \ldots + a_{n_1}\phi_{n_x}(x), \quad \text{and} \\ \psi(y, \mathbf{b}) &= b_1\psi_1(y) + \ldots + b_{n_2}\psi_{n_y}(y), \end{aligned}$$

are two linear models for curves. Then the functions $\gamma_{k\ell}(x, y) = \phi_k(x)\psi_\ell(y)$, $k = 1, \ldots, n_x$, $\ell = 1, \ldots, n_y$ form the *tensor product* set of basis functions for defining linear models for representing surfaces of the form

$$z = \gamma(x, y, \mathbf{a}) = \sum_{k=1}^{n_x} \sum_{\ell=1}^{n_y} a_{jk}\gamma_{k\ell}(x, y). \tag{3}$$

In particular, tensor products of Chebyshev polynomials and B-spline basis functions are used extensively: see below.

Tensor products are particularly useful representations for data (x_i, y_i, z_i) in which the behaviour of the surface is similar across the domain. They are less efficient in representing generally bland surfaces with local areas of large variations. A second (and related) disadvantage is that the number of basis functions is $n_x \times n_y$, so that to capture variation in both x and y a large number of basis functions can be required. On the positive side, if the data points (x_i, y_i) lie on or near a rectangular grid, the computations can be performed very efficiently [1]: see below.

Tensor product surfaces have been proposed [20] for modelling the kinematic behaviour of coordinate measuring machines (CMMs). An empirical model is used to describe the motion of the probe stylus assembly of the CMM (its location and orientation) in terms of three functions specifying a positional correction and three a rotational correction. Each correction is a function of three independent variables, the scale readings returned by the CMM, and is represented by a tensor product of polynomial spline curves.

Tensor product spline surfaces have also been used in the modelling of a photodiode response [34], in which the independent variables are time and active layer thickness. A spline surface approximation is used to smooth measurements of the response, represent concisely the very large quantities of measurements that are made, and to permit effective manipulation of the underlying function including obtaining derivatives and evaluating convolutions.

4.1 Orthogonality of tensor products

If $\{\phi_k\}$ and $\{\psi_l\}$ are orthonormal with respect to inner products

$$\langle p, q\rangle_u = \int_a^b p(x)q(x)u(x)\,dx, \quad \langle p, q\rangle_v = \int_c^d p(x)q(x)v(x)\,dx,$$

respectively, then $\{\gamma_{kl}(x, y) = \phi_k(x)\psi_l(y)\}$ are orthonormal with respect to the inner product

$$\langle p, q\rangle_w = \int_a^b \int_c^d p(x, y)q(x, y)w(x, y)\,dy\,dx,$$

where $w(x, y) = u(x)v(y)$.

4.2 Data approximation using tensor product surfaces

Given data points $(x_i, y_i, z_i),\ i = 1, \ldots, m$, the least-squares best-fit tensor product surface is found by solving

$$\min_{\mathbf{a}} \quad \sum_{i=1}^{m} (z_i - \gamma(x_i, y_i, \mathbf{a}))^2,$$

with $\gamma(x, y, \mathbf{a})$ defined by (3). In matrix terms, we solve

$$\min_{\mathbf{a}} \|\mathbf{z} - \Gamma\mathbf{a}\|^2,$$

where $\mathbf{z} = (z_1, \ldots, z_m)^{\mathrm{T}}$, Γ is an $m \times n_x n_y$ matrix of elements $\gamma_{k\ell}(x_i, y_i)$, and $\mathbf{a}$ is an $n_x n_y \times 1$ vector of elements $a_{k\ell}$. In this formulation, the order of the elements $a_{k\ell}$ in $\mathbf{a}$ (and the order of the corresponding columns of Γ) comes from a choice of ordering of the $n_x n_y$ basis functions $\gamma_{k\ell}(x, y)$.

In the case that the data points relate to measurements on a *grid* in the xy-domain, an alternative linear algebraic formulation is possible that exploits *separability* of the tensor product basis functions and leads to a problem that can be solved significantly faster. Let the data points be $(x_i, y_j, z_{ij}),\ i = 1, \ldots, m_x,\ j = 1, \ldots, m_y$, and let matrices Φ, Ψ, A and Z be defined by

$$\begin{aligned} (\Phi)_{ik} &= \phi_k(x_i), \quad i = 1, \ldots, m_x, \quad k = 1, \ldots, n_x, \\ (\Psi)_{j\ell} &= \psi_\ell(y_j), \quad j = 1, \ldots, m_y, \quad \ell = 1, \ldots, n_y, \end{aligned}$$

and

$$\begin{aligned} (Z)_{ij} &= z_{ij}, \quad i = 1, \ldots, m_x, \quad j = 1, \ldots, m_y, \\ (A)_{k\ell} &= a_{k\ell}, \quad k = 1, \ldots, n_x, \quad \ell = 1, \ldots, n_y. \end{aligned}$$

Then, the surface approximation problem is to solve

$$\min_A \|Z - \Phi A \Psi^{\mathrm{T}}\|^2, \tag{4}$$

the solution to which is given (formally) by

$$\left(\Phi^{\mathrm{T}}\Phi\right) A \left(\Psi^{\mathrm{T}}\Psi\right) = \Phi^{\mathrm{T}} Z \Psi. \tag{5}$$

The solution to (5) may obtained in two stages: by solving

$$\left(\Phi^{\mathrm{T}}\Phi\right) \tilde{A} = \Phi^{\mathrm{T}} Z$$

for $\tilde{A}$, followed by solving

$$A \left(\Psi^{\mathrm{T}}\Psi\right) = \tilde{A}\Psi$$

for A. These relate, respectively, to least-squares solutions of

$$\min_{\tilde{A}} \|Z - \Phi \tilde{A}\|^2, \tag{6}$$

and

$$\min_A \|\tilde{A} - A\Psi^{\mathrm{T}}\|^2. \tag{7}$$

Consequently, the *surface* approximation problem (4) is solved by considering *curve* approximation problems (6) and (7) as follows. First, for each $j = 1, \ldots, m_y$, find the least-squares best-fit curve

$$f_j(x) = \sum_{k=1}^{n_x} \tilde{a}_{kj} \phi_k(x)$$

to the data (x_i, z_{ij}), $i = 1, \ldots, m_x$. Second, for each $i = 1, \ldots, n_x$, find the least-squares best-fit curve

$$f_i(y) = \sum_{\ell=1}^{n_y} a_{i\ell} \psi_\ell(y)$$

to the data $(y_j, \tilde{a}_{ij})$, $j = 1, \ldots, m_y$. The least-squares best-fit surface is therefore obtained in $O(m_x m_y n_x^2 + m_y n_x n_y^2)$ operations compared with $O(m_x m_y n_x^2 n_y^2)$ that would apply if separability of the basis functions is ignored.

4.3 Chebyshev polynomial surfaces

We recall from section 2.1, a polynomial curve $p_n(x)$ of degree n on the interval $x \in [x_{\min}, x_{\max}]$ has the representation

$$p_n(x) = \frac{1}{2} a_0 T_0(\hat{x}) + a_1 T_1(\hat{x}) + \ldots + a_n T_n(\hat{x}) = \sum_{k=0}^{n}{}' a_k T_k(\hat{x}),$$

where $\hat{x} \in [-1, +1]$ is related to x by

$$\hat{x} = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}}$$

and $T_j(\hat{x}),\ j = 0, \ldots, n$, are the *Chebyshev polynomials* (of the first kind) [26, 31] defined by the recursion

$$T_0(\hat{x}) = 1, \quad T_1(\hat{x}) = \hat{x}, \quad T_j(\hat{x}) = 2\hat{x}T_{j-1}(\hat{x}) - T_{j-2}(\hat{x}),\ j > 1.$$

A tensor product polynomial surface $p_{n_x n_y}(x, y)$ of degree n_x in x and n_y in y on the rectangular domain $(x, y) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ is therefore represented by

$$p_{n_x n_y}(x, y) = \sum_{k=0}^{n_x}{}' \sum_{\ell=0}^{n_y}{}' a_{k\ell} T_k(\hat{x}) T_\ell(\hat{y}), \tag{8}$$

where $\hat{x}$ and $\hat{y}$ are each normalised to lie in the interval $[-1, +1]$. (We apply, here, the standard convention that coefficients in the above representation which have either k or ℓ zero are written as $a_{k\ell}/2$, and the coefficient with both k and ℓ zero is written as $a_{00}/4$.) We note that $p_{n_x n_y}(x, y) = p_{n_x n_y}(x, y, \mathbf{a}|\boldsymbol{\lambda})$ depends on the subsidiary parameters $\boldsymbol{\lambda} = (x_{\min}, x_{\max}, y_{\min}, y_{\max})^{\mathrm{T}}$.

The polynomial surface (8) has *total degree* $n_x + n_y$, the highest combined power of x and y of a basis function. Another way of representing a polynomial surface is to require that the *total degree* of the tensor product basis functions is specified as n. Such a polynomial surface has the representation

$$p_n(x, y) = \sum_{k=0, \ell=0}^{k+\ell \le n} a_{k\ell} T_k(\hat{x}) T_\ell(\hat{y}).$$

Figures 5 and 6 show a simulated data set comprising 51×51 points and representing a "feature" arranged parallel to an edge of the rectangular domain for the data. Figures 7 and 8 show a similar simulated data set but with its "feature" arranged close to a diagonal of the domain. A tensor product polynomial surface of degree $n_x = 10$ in x and $n_y = 10$ in y is used to approximate each data set. The residual deviations between the data and the surface approximations are shown in Figures 9 and 10, respectively. Each surface approximation is defined by 11×11 parameters.

4.3.1 Advantages

- Polynomial surfaces involve no subsidiary parameters that need to be chosen. (Implementations in terms of Chebyshev polynomials require $x_{\min}$, etc., to be defined, but these are assigned easily.)

- For data on regular grids, the solution algorithms are efficient and, with the use of orthogonal basis functions, numerically stable.
- Given polynomial approximation software components for one dimension (evaluation of Chebyshev basis functions, etc.) the implementation of algorithms for approximation with tensor product polynomials is straightforward, especially for data on regular grids.
- For data representing similar qualitative behaviour over the domain of interest, it is usually possible to determine good approximations.
- The order of the polynomials can be used to generate nested sequences of spaces from which to approximate the data.

4.3.2 Disadvantages

- For data representing different types of behaviour in different regions, a tensor product representation can be inefficient.
- For scattered data there is no easily tested criteria to determine *a priori* whether or not approximation with a particular order of polynomial will be well-posed.

4.4 Spline surfaces

Recalling section 2.2, a tensor product spline surface $s(x, y)$ of order n_x in x with knots $\boldsymbol{\lambda}$ and order n_y in y with knots $\boldsymbol{\mu}$ on the rectangular domain $(x, y) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ is represented by

$$s(x, y) = s(x, y|\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{k=1}^{n_x+N_x} \sum_{\ell=1}^{n_y+N_y} c_{k\ell} N_{n_x,k}(x|\boldsymbol{\lambda}) N_{n_y,\ell}(y|\boldsymbol{\mu}), \qquad (9)$$

where the knot vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ satisfy, respectively,

$$x_{\min} = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N_x-1} \leq \lambda_{N_x} < \lambda_{N_x+1} = x_{\max}$$

and

$$y_{\min} = \mu_0 < \mu_1 \leq \mu_2 \leq \ldots \leq \mu_{N_y-1} \leq \mu_{N_y} < \mu_{N_y+1} = y_{\max}.$$

The spline surface (9) is a piecewise polynomial of order n_x in x and n_y in y on $(\lambda_{i-1}, \lambda_i) \times (\mu_{j-1}, \mu_j)$, $i = 0, \ldots, N_x$, $j = 0, \ldots, N_y$. The spline is C^{n_x-k-1} along the knot-line $x = \lambda_i$ if $\text{card}(\lambda_\ell = \lambda_i, \ell \in \{1, \ldots, N_x\}) = k$ (and similarly for the knot-line $y = \mu_j$). So, for example, a spline surface

of order four in x and y for which the λ_i and μ_j are distinct is a piecewise bicubic polynomial of continuity class C^2 along the lines $x = \lambda_i$ and $y = \mu_j$.

A tensor product spline surface of order $n_x = 4$ in x and $n_y = 4$ in y, with knot-lines arranged as shown in Figure 11, is used to approximate the simulated data sets shown in Figures 5 and 7. The residual deviations between the data and the surface approximations are shown in Figures 12 and 13, respectively. The spline surface approximation to the data set of Figure 5 is considerably better than that obtained using a polynomial surface, and requires only 7×7 coefficients for its representation. However, The spline surface approximation to the data of Figure 7 is considerably worse than that obtained using a polynomial surface. Although spline surfaces (with straight knot-lines) are capable of representing a wide variety of shapes, their efficiency in doing so (measured by the number of basis functions required) depends to a large extent on a "uniformity" of the shape with x and y.

Greater flexibility of tensor product spline surfaces is achieved by allowing the knot-lines to be curved. Knot-lines are defined to follow the "features" of the surface represented by the data and, in this way, efficient spline surface approximations are possible. Define knot-lines $x = \lambda_i(y),\ i = 1, \ldots, N_x$, and $y = \mu_j(x),\ j = 1, \ldots, N_y$, to satisfy for all y

$$x_{\min} < \lambda_1(y) \leq \lambda_2(y) \leq \ldots \leq \lambda_{N_x-1}(y) \leq \lambda_{N_x}(y) < x_{\max},$$

and for all x

$$y_{\min} < \mu_1(x) \leq \mu_2(x) \leq \ldots \leq \mu_{N_y-1}(x) \leq \mu_{N_y}(x) < y_{\max}.$$

Then, a tensor product spline surface with the above knot-lines is represented by

$$s(x, y) = \sum_{k=1}^{n_x+N_x} \sum_{\ell=1}^{n_y+N_y} c_{k\ell} N_{n_x,k}(x|\boldsymbol{\lambda}(y)) N_{n_y,\ell}(y|\boldsymbol{\mu}(x)).$$

A tensor product polynomial spline surface of order $n_x = 4$ in x and $n_y = 4$ in y, with knot-lines arranged as shown in Figure 14, is used to approximate the simulated data set shown in Figure 7. The residual deviations between the data and the surface approximation are shown in Figure 15. The spline surface approximation is now as good as that to the data of Figure 5 using straight knot-lines (Figure 12).

4.4.1 Advantages

- For data on regular grids, the solution algorithms are extremely efficient and numerically stable. For scattered data, it is still possible

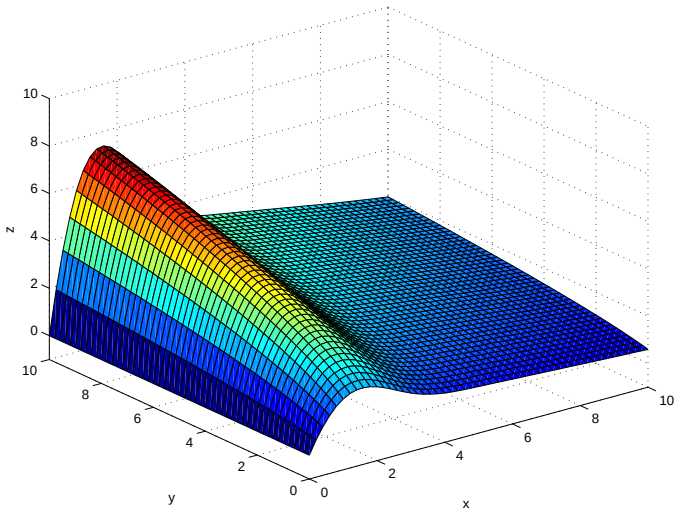

Figure 5: Surface plot of simulated data with feature parallel to the edge of the rectangular domain.

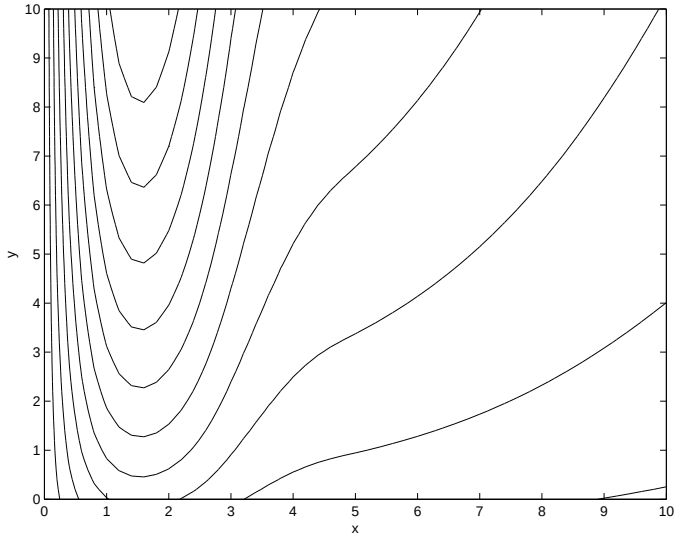

Figure 6: Contour plot for simulated data shown in Figure 5.

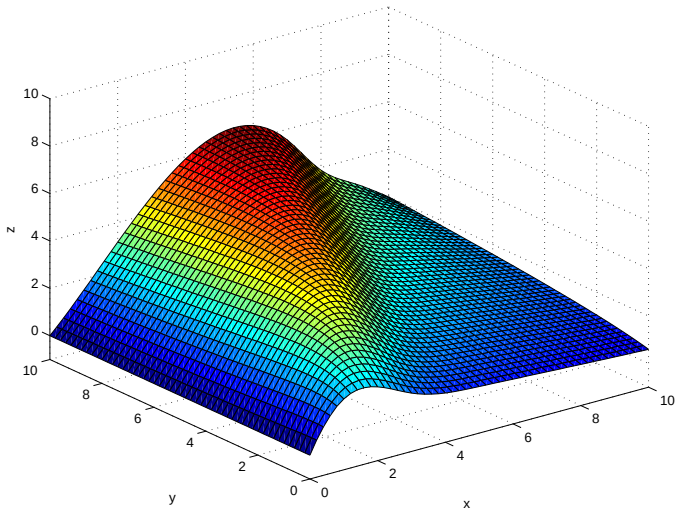

Figure 7: Surface plot of simulated data with feature along a diagonal of the rectangular domain.

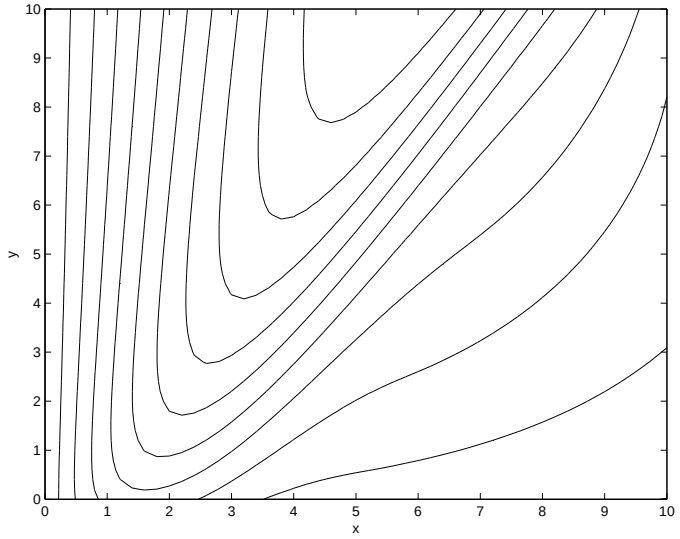

Figure 8: Contour plot for simulated data shown in Figure 7.

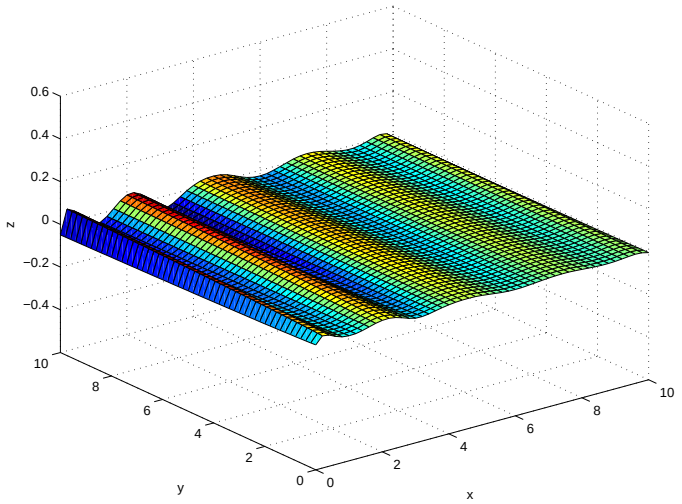

Figure 9: Residual deviations associated with the tensor product polynomial surface approximation to the data shown in Figure 5, of degrees $n_x = 10$ in x and $n_y = 10$ in y.

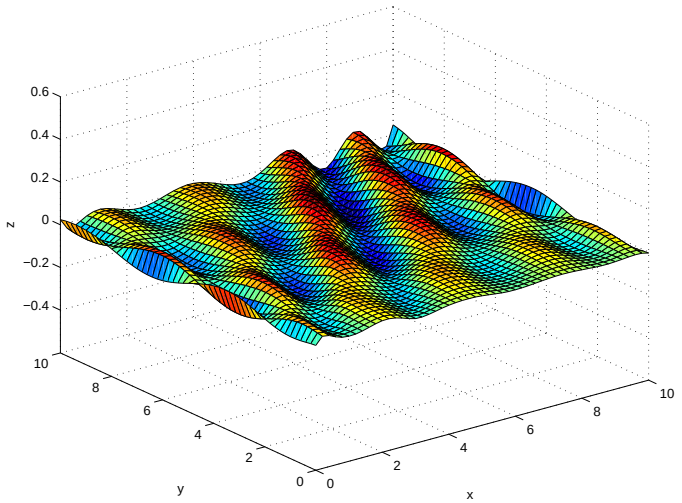

Figure 10: Residual deviations associated with the tensor product polynomial surface approximation to the data shown in Figure 7, of degrees $n_x = 10$ in x and $n_y = 10$ in y.

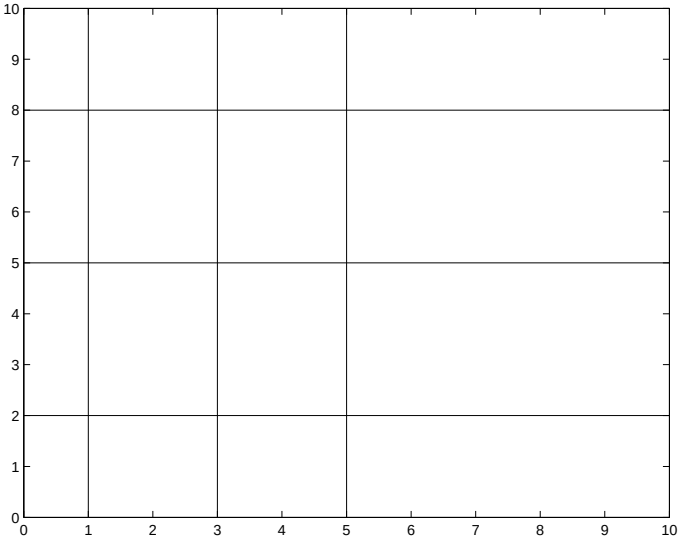

Figure 11: Knot-lines used in the representation of a tensor product spline surface approximation.

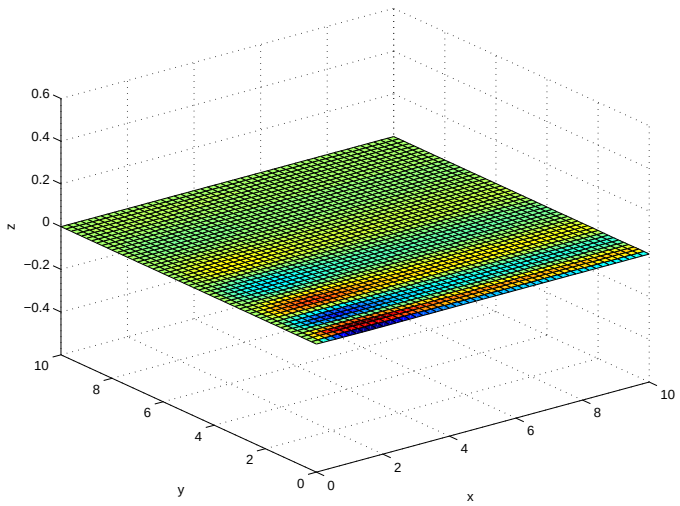

Figure 12: Residual deviations associated with the tensor product polynomial spline surface approximation to the data shown in Figure 5, of orders $n_x = 4$ in x and $n_y = 4$ in y with knot-lines as illustrated in Figure 11.

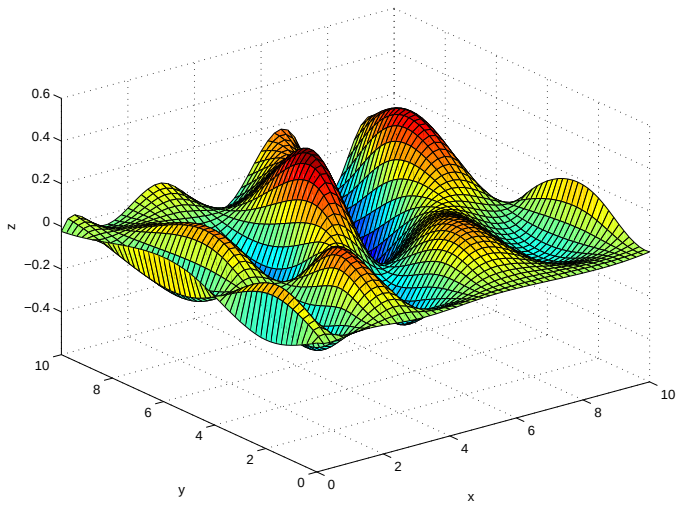

Figure 13: Residual deviations associated with the tensor product polynomial spline surface approximation to the data shown in Figure 7, of orders $n_x = 4$ in x and $n_y = 4$ in y with knot-lines as illustrated in Figure 11.

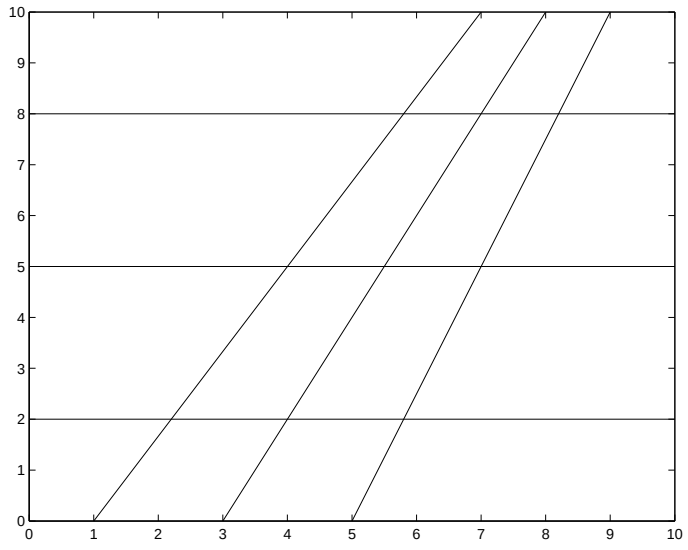

Figure 14: Knot-lines used in the representation of a tensor product spline surface approximation.

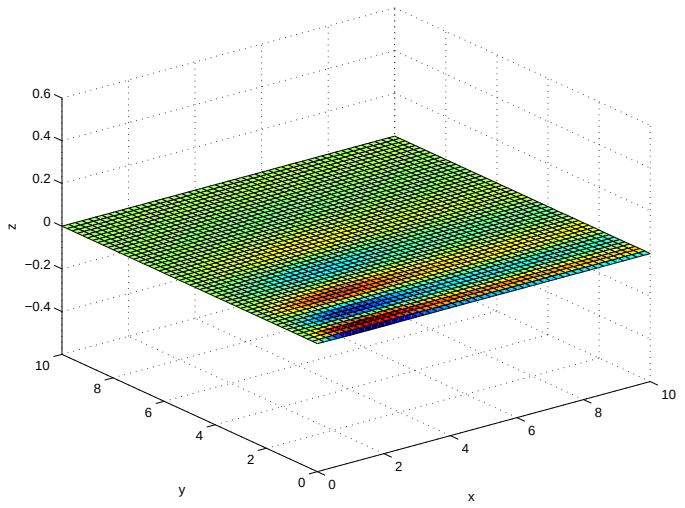

Figure 15: Residual deviations associated with the tensor product polynomial spline surface approximation to the data shown in Figure 7, of orders $n_x = 4$ in x and $n_y = 4$ in y with knot-lines as illustrated in Figure 14.

to exploit sparsity structure in the observation matrix but the gain in efficiency is much less than that for the case of one-dimension.

- Given spline approximation software components for one dimension (evaluation of B-spline basis functions, etc.) the implementation of algorithms for approximation with tensor product polynomials is straightforward for data on regular grids.

- For data representing similar qualitative behaviour over the domain of interest, it is usually possible to determine good approximations.

- The knot vectors can be chosen to generate nested sequence of spaces from which to approximate the data.

- For data on a rectangular grid, it is easy to check *a priori* whether a particular choice of knots will lead to a well-posed approximation problem.

4.4.2 Disadvantages

- Splines require the subsidiary knot vectors to be chosen. If the data or surface exhibits different behaviour in different regions, the choice of knots can affect significantly the quality of the spline representation [19].

- For data representing different types of behaviour in different regions, a tensor product representation can be inefficient.
- For scattered data, there is no easily tested criteria to determine *a priori* whether or not approximation with splines defined by a pair of knot sets will be well posed.

4.5 Heterogeneous tensor products

The examples of polynomial and spline surfaces discussed above have tensor product basis functions $\gamma_{kl} = \phi_k(x)\psi_l(y)$ where both ϕ_k and ψ_l are the same type of function. However, in many applications it is desirable to have different forms. For example, in cylindrical coordinates, we can have basis functions of the form

$$\gamma_{kl}(\theta, z) = \phi_k(z)\cos l\theta \quad \text{or} \quad \phi_k(z)\sin l\theta,$$

where $\{\phi_k\}$ are orthogonal polynomials, to represent periodic behaviour about the the cylinder axis and general behaviour parallel to the axis. For such models, if the data is located according to the appropriate regularity structure (i.e., grid-like), the separability property applies and the data approximation can be implemented very efficiently.

5 Wavelets

Wavelets are now an important tool in data analysis and a survey of their application to metrology is given in [39]. Example applications include surface metrology.

5.1 Wavelets in one dimension

In one dimension, wavelets are often associated with a multiresolution analysis (MRA). In outline, let $L^2(\mathbb{R})$ be the space of square integrable functions $f : \mathbb{R} \longrightarrow \mathbb{R}$ so that

$$\int_{-\infty}^{\infty} f^2(x)\,dx < \infty.$$

If $f, g \in L^2(\mathbb{R})$ we define

$$\langle f, g\rangle = \int_{-\infty}^{\infty} f(x)g(x)\,dx,$$

and $\|f\|^2 = \langle f, f\rangle$. This inner-product is used to define orthogonality for functions in $L^2(\mathbb{R})$.

A starting point for MRA is a function $\psi(x)$, the *mother wavelet*. From ψ we define a double sequence of functions

$$\psi_{j,k} = \frac{1}{2^{j/2}}\psi(2^{-j}x - k),$$

using translations and dilations. The mother wavelet is chosen to that $\{\psi_{j,k}\}$ forms an orthonormal basis for $L^2(\mathbb{R})$, so that any $f \in L^2(\mathbb{R})$ can be expressed as

$$f(x) = \sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} \langle f, \psi_{j,k}\rangle\psi_{j,k}(x).$$

The functions $\{\psi_{j,k}\}$, $k \in \mathbb{Z}$, form an orthonormal basis for a subspace W_j of $L^2(\mathbb{R})$ and these subspaces are used to define a nested sequence of subspaces

$$\ldots \supset V_{j-1} \supset V_j \supset V_{j+1} \supset \ldots$$

where

$$V_{j-1} = V_j \oplus W_j,$$

i.e., any function $f_{j-1} \in V_{j-1}$ can be uniquely expressed as $f_{j-1} = f_j + g_j$, with $f_j \in V_j$ and $g_j \in W_j$. We regard f_j as a smoother approximation to f_{j-1} (since $f(x) \in V_{j-1}$ if and only if $f(2x) \in V_j$) while g_j represents the difference in detail between f_{j-1} and f_j.

The orthogonality properties mean that computations using wavelets can be made very efficiently. In particular, the discrete wavelet transform is used to decompose a uniformly spaced finite set of discrete data points (j, f_j) into component functions at different frequencies (or scales). A major feature of a wavelet analysis is that (unlike Fourier analysis) it can describe different frequency behaviour at different locations.

5.2 Higher dimension wavelets

Wavelets can also be used to analyse signals in higher dimensions. From the orthonormal wavelet basis for $L^2(\mathbb{R})$,

$$\{(\psi_{j,k}(x), j, k \in \mathbb{Z}\}$$

an orthonormal basis for $L^2(\mathbb{R}^2)$ is obtained by taking the tensor products of two one-dimensional bases functions

$$\psi_{j_1,k_1,j_2,k_2}(x, y) = \psi_{j_1,k_1}(x)\psi_{j_2,k_2}(y).$$

and these functions can be used for MRA in two dimensions.

5.2.1 Advantages

- Wavelets are able to represent different types of behaviour in different regions.
- For data lying on a regular grid, algorithm implementations are efficient and numerically stable.
- Wavelets provide a nested sequence of spaces from which to approximate the data.
- Wavelets are important tools in filtering and data compression.
- Wavelets do not require the specification of subsidiary parameters (but a choice of mother wavelet is required).
- Many wavelet software packages are available.

5.2.2 Disadvantages

- Most wavelet implementations are concerned with data on a regular grid.
- The relationship between the choice of wavelet and the effectiveness of resulting analysis is not obvious.

6 Scattered data approximation

Tensor product surfaces are especially effective for approximating data where the xy-coordinates (x_i, y_i) are situated on a regular grid. If the locations of (x_i, y_i) are scattered, the tensor product approach is much less efficient. In this section we look at methods that apply to scattered data.

6.1 Polynomial surfaces

In the case of one dimension, given a set of data $X = \{(x_i, y_i)\}_{i=1}^m$, the Forsythe method generates, implicitly, a set of orthogonal polynomials $\phi_j(x)$ such that

$$\langle \phi_j, \phi_k \rangle = \sum_{i=1}^{m} \phi_j(x_i)\phi_k(x_i) = 0, \quad j \neq k.$$

Furthermore if there are at least n distinct x_i, then approximating the data with an order n (degree $n-1$) polynomial is a well-posed problem – the associated observation matrix has full rank. In two (or higher) dimensions conditions to guarantee a well conditioned approximation problem are much more complex. For example, if the data points (x_i, y_i, z_i) are such that (x_i, y_i) lie on a circle then the basis vectors corresponding to the basis functions x^2, y^2, x, y and 1 will be linearly dependent. More generally, if (x_i, y_i) lie on (or near to) an algebraic curve (i.e., one defined as the zeros of a polynomial), then the associated observation matrix will be rank deficient (or poorly conditioned).

In a report by Huhtanen and Larsen [37], an algorithm is presented for generating bivariate polynomials that are orthogonal with respect to a discrete inner product. It is straightforward to implement and includes provision for the possibility of linear independency amongst the basis vectors. The algorithm also provides a recursive scheme to evaluate the polynomial where the length of the recursion is at most $2k+1$ where k is the degree of the polynomial. We illustrate the use of this algorithm in fitting data generated on the surface

$$z = x^4 - y^4 + xy^3 - x^3y + 2. \tag{10}$$

We have generated 101 data points (x_i^*, y_i^*) uniformly distributed around the circle $x^2 + y^2 = 1$ and calculated z_i^* according to (10) so that (x_i^*, y_i^*, z_i^*) lie exactly on the surface; see Figure 16. We have then added random perturbations to generate data points (x_i, y_i, z_i):

$$x_i = x_i^* + e_i, \quad y_i = y_i^* + f_i, \quad z_i = z_i^* + g_i, \quad e_i, f_i, g_i \in N(0, \sigma^2).$$

There are 15 basis functions associated with a bivariate polynomial of total degree 4. For the data points $\{(x_i^*, y_i^*)\}$ and degree $k = 4$ the algorithm

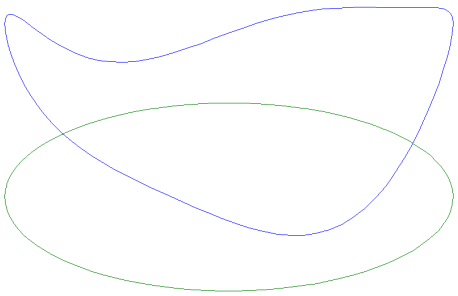

Figure 16: Curve defined by the quartic surface (10) intersected with the cylinder $x^2 + y^2 = 1$.

generates 10 orthogonal vectors out of a possible 15, the remaining five being linear combinations of the other basis vectors. The maximum computed element $|(Q^*)^{\mathrm{T}}Q^* - I|$ was 1.5543×10^{-15}. For the data points, $\{(x_i, y_i)\}$, the random perturbations are enough to ensure that the basis functions are linearly independent and the algorithm produces all 15 orthogonal vectors. The maximum computed element of $|Q^{\mathrm{T}}Q - I|$ was 5.0774×10^{-14}.

This algorithm is certainly of interest for those who wish to approximate multivariate data with polynomials and it is likely there will be further developments. Multivariate orthogonal polynomials is an area of considerable research activity (see, e.g., [22]) and the Society for Industrial and Applied Mathematics has a special interest group devoted to it.

6.1.1 Advantages

- The Huhtanen and Larsen (HL) algorithm provides a method of approximating scattered data by polynomials.
- The algorithm is efficient compared to a full matrix approach and has favourable numerical properties.
- The algorithm copes with possible rank deficiency in the basis functions.
- The HL algorithm is reasonably straightforward to implement.
- The same approach can be applied in higher dimensions.

- The total order of the polynomial can be chosen to generate a nested sequence of spaces from which to chose an approximant.

6.1.2 Disadvantages

- Standard numerical tools for its implementation are not yet widely available.

6.2 RBFs: radial basis functions

In section 6.1, we indicated that some recently developed algorithms concern the approximation of scattered data using polynomials. Recent years have seen much interest in a different approach to scattered data approximation.

Let $\Lambda = \{\boldsymbol{\lambda}_j\}$, $j = 1, \ldots, n$, be a set of points in $\mathbb{R}^p$, and $\rho : \mathbb{R} \longrightarrow [0, \infty)$ a fixed function. A radial basis function (RBF) with centres Λ has the form

$$\phi(\mathbf{x}, \mathbf{a}) = \phi(\mathbf{x}, \mathbf{a}|\Lambda) = \sum_{j=1}^{m} a_j \rho(\|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

where $\|\mathbf{x}\| = (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{1/2}$ is the Euclidean norm of a vector. Defining

$$\phi_j(\mathbf{x}) = \rho(\|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

then ϕ is seen to be a linear combination of basis functions. Therefore, approximation with RBFs follows the same general approach as with other empirical models defined in terms of basis functions. Given a set of data points $X = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}\}$, $i = 1, \ldots, m$, the associated observation matrix has

$$B_{ij} = \rho(\|\mathbf{x}_i - \boldsymbol{\lambda}_j\|).$$

In least squares approximation, estimates of the parameters $\mathbf{a}$ are found by solving

$$\min_{\mathbf{a}} \|\mathbf{y} - B\mathbf{a}\|^2.$$

Common choices for the function ρ are i) $\rho(r) = r^3$, *cubic*, ii) $\rho(r) = e^{-r^2}$, *Gaussian*, iii) $\rho(r) = r^2 \log r$, *thin plate spline*, iv) $\rho(r) = (r^2 + \lambda^2)^{1/2}$, *multiquadric*, and v) $\rho(r) = (r^2 + \lambda^2)^{-1/2}$, *inverse multiquadric*. In practice, a scaling parameter μ_0 is required so that the RBF has the form

$$\phi(\mathbf{x}, \mathbf{a}|\mu_0, \Lambda) = \sum_{j=1}^{m} a_j \rho(\mu_0 \|\mathbf{x} - \boldsymbol{\lambda}_j\|).$$

The basic idea of an RBF can be generalised in many ways including the introduction of scaling parameters associated with each centre,

$$\phi(\mathbf{x}, \mathbf{a}|\boldsymbol{\mu}, \Lambda) = \sum_{j=1}^{m} a_j \rho(\mu_j \|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

or even symmetric positive definite matrices M_j:

$$\phi(\mathbf{x}, \mathbf{a}|\{M_j\}, \Lambda) = \sum_{j=1}^{m} a_j \rho((\mathbf{x} - \boldsymbol{\lambda}_j)^{\mathrm{T}} M_j (\mathbf{x} - \boldsymbol{\lambda}_j)).$$

However, with the introduction of more flexibility comes the problem of how to use this flexibility effectively.

6.2.1 Advantages

- RBFs apply to scattered data.
- RBFs apply to multivariate data in any dimension. The computational cost is $O(mn(n + p)$, where m is the number of data points, n is the number of centres and p is the dimension.
- RBFs can be used to represent different types of behaviour in different regions.
- It is generally possible to chose centres so that the data approximation problem is well-posed, i.e., there is no rank deficiency.
- RBF algorithms are easy to implement involving only elementary operations and standard numerical linear algebra.
- By choosing the set of centres Λ appropriately, it is possible to generate a nested sequence of spaces from which to choose an approximant.

6.2.2 Disadvantages

- RBF basis functions have no natural orthogonality and can often be poorly conditioned.
- RBFs give rise to full observation matrices with no obvious way of increasing computational efficiency.
- RBFs require the choice of subsidiary parameters, i.e., the centres and scaling parameter(s).

This disadvantages are discussed further in section 10.

7 NURBS: nonuniform rational B-splines

7.1 NURBS curves

Nonuniform rational B-splines (NURBS) are used for computer graphics and extensively in computer-aid design for defining complex curves and surfaces and are therefore important in co-ordinate metrology. A nonuniform rational B-splines curve of order k is defined as a parametric curve $\boldsymbol{C} : \mathbb{R} \longrightarrow \mathbb{R}^2$ with

$$\boldsymbol{C}(u) = \frac{\sum_{j=0}^{n} N_{k,j}(u|\boldsymbol{\lambda}) w_j \mathbf{P}_j}{\sum_{j=0}^{n} N_{k,j}(u) w_j}$$

where $\boldsymbol{P_j} \in \mathbb{R}^2$ are the control points, w_j weights and $N_{k,j}(u|\boldsymbol{\lambda})$ B-spline basis functions defined on a knot set $\boldsymbol{\lambda}$.

7.2 NURBS surfaces

NURBS surfaces $\boldsymbol{S} : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ are generated using tensor products of B-spline basis functions:

$$\boldsymbol{S}(u,v) = \frac{\sum_{j=0}^{n} \sum_{q=0}^{m} N_{k,j}(u|\boldsymbol{\lambda}) N_{l,q}(v|\boldsymbol{\mu}) w_{jq} \mathbf{P}_{jq}}{\sum_{j=0}^{n} \sum_{q=0}^{m} N_{k,j}(u|\boldsymbol{\lambda}) N_{l,q}(v|\boldsymbol{\mu}) w_{jq}},$$

where $N_{k,j}(u|\boldsymbol{\lambda})$ and $N_{l,q}(v|\boldsymbol{\mu})$ are the B-spline basis functions, $\mathbf{P}_{ij} \in \mathbb{R}^3$ are control points, and w_{jq} weights.

7.2.1 Advantages

- NURBS can be used to model and modify highly complex curves and surfaces.
- The shape of the curve or surface is easily determined and modified by the location of the control points. NURBS provide local control, so that shifting one control point only affects the surface shape near that control point.
- NURBS are invariant under scaling, translation, shear, and rotation,
- NURBS can be used to define quadric surfaces, such as spheres and ellipsoids, commonly used in CAD exactly. Parametric B-spline surfaces can only approximate such surfaces and in doing so require many more control points.

7.2.2 Disadvantages

- Although NURBS are in principle straightforward to implement, efficient and numerically stable approaches require appropriate use of the recurrence formulae associated with B-splines.
- Data approximation with NURBS (fitting a cloud of points with a NURBS curve or surface) is likely to give rise to rank deficient or poorly conditioned problems. However there are a number of ways of approaching approximation with parametric curves and surfaces, some of which give rise to well conditioned problems (see, e.g., [8, 24]).

8 Neural networks and support vector machines

Neural networks (NNs), see, e.g., [5, 6, 35], represent a broad class of empirical multivariate models.

8.1 Multilayer perceptron

In a multilayer perceptron (MLP) [35, 40], a vector of inputs $\mathbf{x}$ is transformed to a vector of outputs $\mathbf{z}$ through a sequence of matrix-vector operations combined with the application of nonlinear *activation functions*. Often a network has three layers of nodes – input, hidden and output – and two transformations $\mathbb{R}^m \longrightarrow \mathbb{R}^l \longrightarrow \mathbb{R}^n$, $\mathbf{x} \longrightarrow \mathbf{y} \longrightarrow \mathbf{z}$ with

$$y_j = \psi(\mathbf{a}_j^{\mathrm{T}}\mathbf{x} + b_j), \quad z_k = \phi(\mathbf{c}_k^{\mathrm{T}}\mathbf{y} + d_k),$$

or, in matrix terms,

$$\mathbf{y} = \boldsymbol{\psi}(A\mathbf{x} + \mathbf{b}), \quad \mathbf{z} = \boldsymbol{\phi}(C\mathbf{y} + \mathbf{d}) = M(\mathbf{x}, A, \mathbf{b}, C, \mathbf{d}),$$

where A is an $l \times m$ matrix, C an $n \times l$ matrix, and $\mathbf{b}$ and $\mathbf{d}$ are l- and n-vectors, respectively. The activation function is often chosen to be the logistic sigmoid function $1/(1 + e^{-x})$ or a hyperbolic tangent function $\tanh(x)$. These functions have unit gradient at zero and approach 1 as $x \to \infty$ and 0 or -1 as $x \to -\infty$. For classification problems, the network is designed to work as follows. The value of y_j indicates whether a feature specified by $\mathbf{a}_j$ is present ($y_j \approx 1$) or absent ($y_j \approx 0$ or -1) in the input $\mathbf{x}$. The output $\mathbf{z}$ completes the classification of the input according to the features identified in the hidden layer $\mathbf{y}$: the input is assigned to the qth class if $z_q \approx 1$ and $z_r \approx 0$ or -1, $r \neq q$. For empirical modelling, the second activation function is usually chosen to be the identity function $\phi(x) = x$, so that all values of output are possible, and

$$\mathbf{z} = M(\mathbf{x}, A, \mathbf{b}, C, \mathbf{d}) = C\boldsymbol{\psi}(A\mathbf{x} + \mathbf{b}) + \mathbf{d} \tag{11}$$

a flexible multivariate function $M : \mathbb{R}^m \longrightarrow \mathbb{R}^n$.

Given training data comprising sets of inputs and required outputs $\{(\mathbf{x}_q, \mathbf{z}_q)\}$, an iterative optimisation process – the back-propagation algorithm – can be used to adjust the weighting matrices A and C and bias vectors $\mathbf{b}$ and $\mathbf{d}$ so that $M(\mathbf{x}_q, A, \mathbf{b}, C, \mathbf{d}) \approx \mathbf{z}_q$. Alternatively, standard large-scale optimisation techniques [17, 28, 30, 54] such as conjugate gradient methods can be employed. However, the optimisation problems are likely to be poorly conditioned or rank deficient and the optimisation algorithms need to cope with this possibility. Many algorithms therefore employ large-scale techniques combined with regularisation techniques [32, 33, 48].

MLP models are extremely flexible. Many of the problems associated with implementing them for a particular application are in deciding how to reduce the flexibility in order to produce a compact model while at the same time retaining enough flexibility in order to represent adequately the system being modelled.

If $\mathcal{M}_{mln}$ is the space of all MLP models with m, l and n nodes, then $\mathcal{M}_{mln} \subset \mathcal{M}_{mqn}$, $l < q$.

8.2 RBF networks

Radial basis function (RBF) networks [7, 43, 44] have a similar design to multilayer perceptrons (MLPs) but the activation function is a radial basis function. Typically, we have

$$y_j = \rho_j(\|\mathbf{x} - \boldsymbol{\lambda}_j\|), \qquad \mathbf{z} = C\mathbf{y} + \mathbf{d},$$

where ρ_j is a Gaussian function $\rho_j(x) = \exp\{-x^2/(2\sigma_j^2)\}$, for example. More generally, we can have

$$y_j = \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\lambda})^{\mathrm{T}} M_j (\mathbf{x} - \boldsymbol{\lambda})\right\},$$

where M_j is a symmetric, semi-positive definite matrix.

8.2.1 Advantages

- NNs can be used to approximate any continuous function $\mathbf{f} : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ [27, 36].
- NNs can be used to perform nonlinear classification, in which data points belonging to different classes are separated by nonlinear hypersurfaces.
- NN models are straightforward to evaluate and back-propagation algorithms, for example, are easy to implement.
- The dimension of the hidden layer(s) can be used to generate a nested sequence of spaces from which to choose a model.

8.2.2 Disadvantages

- The determination of optimal weights and biases is a nonlinear optimisation problem.

- The back-propagation algorithm can converge slowly to one of possibly many local minima.
- The behaviour of the model on training data can be a poor guide to its behaviour on similar data.
- The evaluation of the uncertainty associated with the fitted parameters is difficult.
- The effectiveness of the network can depend critically on its design (number and size of hidden layers).

8.3 Support vector machines

Support vector machines (SVMs) are another tool used in classification problems [45, 46, 49]. Consider first the problem of finding a simple rule which classifies a set of data points $\{\mathbf{x}_i \in \mathbb{R}^n : i = 1, \ldots, m\}$ into two classes specified by associated indices $y_i = \pm 1$. The simplest approach is to try to find a hyperplane

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0,$$

specified by coefficients $\mathbf{w} \in \mathbb{R}^n$ and constant b which separates the two sets of points so that $\mathbf{w}$ and b satisfy

$$\operatorname{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) = y_i, \quad i = 1, \ldots, m,$$

or, equivalently,

$$(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)y_i \geq 1, \quad i = 1, \ldots, m.$$

If such a hyperplane exists we say the classes are linearly separable. However, in many practical applications the separating hyperplane does not exist and, therefore, we introduce slack variables

$$\xi_i \geq 0, \quad i = 1, \ldots, m, \tag{12}$$

to give modified conditions

$$(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)y_i \geq 1 - \xi_i, \quad i = 1, 2, \ldots, m. \tag{13}$$

The role of the slack variables is to allow a model to incorrectly classify data points that lie close to the hyperplane.

With γ pre-specified, the SVM approach to determining $\mathbf{w}$ and b seeks to minimise

$$\gamma \sum_{i=1}^{m} \xi_i + \mathbf{w}^{\mathrm{T}}\mathbf{w},$$

subject to the constraints (12) and (13). Introducing Lagrange multipliers, this constrained optimisation problem can be reformulated as maximising

$$\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$$

subject to

$$0 \le \alpha_i \le \gamma, \quad i = 1, \ldots, m, \text{ and } \sum_{i=1}^{m}\alpha_i y_i = 0.$$

This problem involves maximising a quadratic function subject to linear inequality constraints. From its solution the coefficients $\mathbf{w}$ are found from

$$\mathbf{w} = \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i.$$

The Karush-Kuhn-Tucker complimentary conditions require

$$\alpha_i \left[y_i(\mathbf{x}_i^{\mathrm{T}}\mathbf{w} + b) - 1 \right] = 0, \quad i = 1, 2, \ldots, m,$$

where $0 \le \alpha_i \le \gamma$. Choosing i such that $\alpha_i \in (0, \gamma)$, i.e, $\alpha_i \neq 0$ and $\alpha_i \neq \gamma$, provides an equation for b. For this $\mathbf{w}$ and b, we obtain the classifier

$$f(x) = \text{sign}\left(\sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i^{\mathrm{T}}\mathbf{x} + b \right). \tag{14}$$

A non-zero value for α_i in (14) signifies that the ith data point is a support vector in the model. The parameter γ is used to balance the success in classification against the number of support vectors required to define the separating surface.

The support vector machine classifier (14) computes a linear decision surface between classes. In order to allow for more general decision surfaces, we look for mappings $\Phi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ such that the inner product $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y})$ for some *kernel function* $k(\mathbf{x}, \mathbf{y})$. In other words, the inner-product in $m-$space can be computed in terms of n-vectors through k. The idea in using such a Φ is to map $\{\mathbf{x}_i\}$ nonlinearly into a higher dimensional space in which there is a separating hyperplane without incurring any serious computational penalty. An example mapping is

$$\Phi : \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{bmatrix}.$$

We note that

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 = \left(\mathbf{x}^{\mathrm{T}}\mathbf{y}\right)^2,$$

so that the corresponding kernel function is $k(\mathbf{x}, \mathbf{y}) = \left(\mathbf{x}^{\mathrm{T}}\mathbf{y}\right)^2$. Mercer's Theorem (see, e.g., [45]) gives reasonably straightforward criteria on functions k such that $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ for some transformation Φ. Examples of such kernel functions are:

- Polynomial kernel given by,

$$k(\mathbf{x}, \mathbf{y}) = \left(\mathbf{x}^{\mathrm{T}}\mathbf{y}\right)^d, \quad d > 0.$$

- Gaussian kernel given by,

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-||\mathbf{x} - \mathbf{y}||_2^2\right).$$

- Hyperbolic tangent kernel given by,

$$k(\mathbf{x}, \mathbf{y}) = \tanh\left(\alpha + \beta\mathbf{x}^{\mathrm{T}}\mathbf{y}\right), \quad \alpha, \beta \geq 0.$$

The use of kernels allows us to apply the same algorithmic approaches to classification as for linear separation but with classification functions of the form

$$\mathrm{sign}\left(\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b\right).$$

Support vector machines have developed from research in the pattern recognition and machine learning community [49, 50, 51]. The use of kernel functions has a more general application [2, 52, 53] and provides SVMs with effective algorithmic tools. SVMs have also an interesting foundation in statistical inference in which functional relationships between variables is elicited from observed regularity in observational data. Although pattern recognition is somewhat removed from metrological data analysis, SVMs and their underlying technology are of potential value to metrology, particularly in situations in which the system under study is imperfectly understood, for example, in biotechnology or soft metrology.

9 Example: polynomial, spline and RBF fits to interferometric data

We illustrate the use of polynomials, splines and RBFs in approximating data arising from the interferometric measurement of an optical surface. Figure 17 shows an example set of measurements of heights z_i in nanometres over a regular 146×146 grid.

We have fitted this data using the following models and algorithms:

A Gaussian RBF with centres on a regular 10×10 grid,

B thin plate spline RBF with the same centres as A,

C 105 orthogonal polynomials of up to total degree 13 generated using the algorithm of Huhtanen and Larsen [37],

D tensor product polynomial with basis functions $\gamma_{kl} = T_k(x)T_l(y)$, $0 \leq k, l \leq 9$, where T_k is a Chebyshev basis functions of degree k, (i.e., 100 parameters in all),

E as D but with $0 \leq k, l \leq 39$ (i.e., 1600 parameters),

F a tensor product bicubic spline with 6 interior knots along each axis (i.e., 100 parameters in all), and

G as F with with 36 interior knots on each axis (i.e., 1600 parameters).

Figure 18 plots the fitted surface determined using algorithm A to the data in Figure 17 while Figure 19 plots the associated residuals. Figures 20–25 graph the residuals associated with algorithms B – G. Figures 26 and 27 shows the fitted surfaces associated with algorithms E and G.

9.1 Remarks

9.1.1 Quality of fit

All methods generally produce a good fit. Visually the fitted surfaces seem to model the data very well. Only the fit associated with the orthogonal polynomial generated using the HL algorithm shows unwanted edge effects in the residuals (Figure 21). The RMS residual error for all fits ranges from approximately 1 nm for algorithms E and G involving 1600 parameters to 3 nm for algorithm B.

9.1.2 Computational efficiency

These experiments involve over 21,000 data points and approximately 100 to 1600 parameters and represent computational problems quite large by comparison with many approximation problems in metrology. The tensor product approaches for a regular grid data are approximately n times faster than the full matrix approach associated with RBF approximation where n is the number of parameters.

To give an idea of the computational requirements of the various algorithms, using Matlab 6.5 and a Dell Optiplex GX240, Intel Pentium 4, 1.7 GHz PC, the time taken for algorithm A was i) 8.5 s to determine the matrix D of distances $d_{ij} = \|\mathbf{x}_i - \boldsymbol{\lambda}_j\|$, ii) 0.9 s to apply ρ to D, iii) 4.1 s to solve the linear least squares system, making a total of 13.5 s. By comparison algorithm D took 0.02 s. For algorithm C, the time taken to generate the orthogonal basis was 9.8 s. The time taken to calculate the matrix D is comparatively slow since it involves two iterative loops. In Fortran, for example, this step would be relatively much quicker. Similar remarks apply to the implementation of the HL algorithm in Matlab.

9.1.3 Condition of the observation matrices

For the tensor polynomial and splines approaches, the condition numbers of the matrices generated were all less than 10. For the Gaussian RBF (algorithm A) the condition number was 4.2×10^6, while that for the thin plate spline (algorithm B) was 1.1×10^4. For the HL algorithm (C), the maximum absolute value of the off-diagonal elements of the computed matrix $Q^{\mathrm{T}}Q$ was 4.0×10^{-14}.

The data set in Figure 17 lay on a regular grid and we were able to exploit this in fitting tensor product polynomial and spline surfaces. The RBF approach applies equally to scattered data. We have taken a random subset of the data in Figure 17 and fitted Gaussian and thin plate spline RBFs to the data. Figure 28 plots the $xy-$coordinates of the data along with the RBF centres; Figure 29 plots the coordinate data. Figure 30 plots the residual errors associated with the fits of a Gaussian and thin plate spline RBF and bivariate polynomial of total degree 13 generated using the HL algorithm.

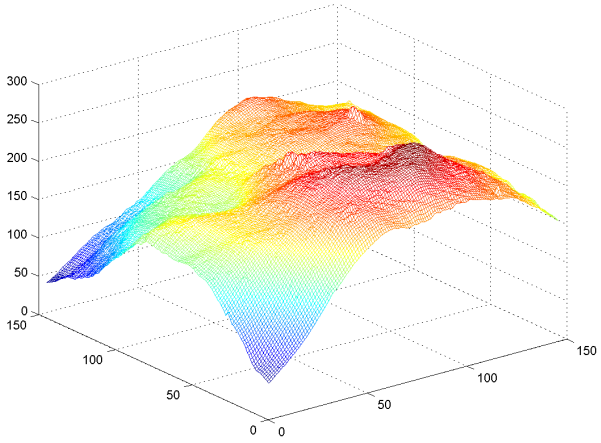

Figure 17: Measurements of an optical surface using interferometry. The units associated with the vertical axis are nanometres.

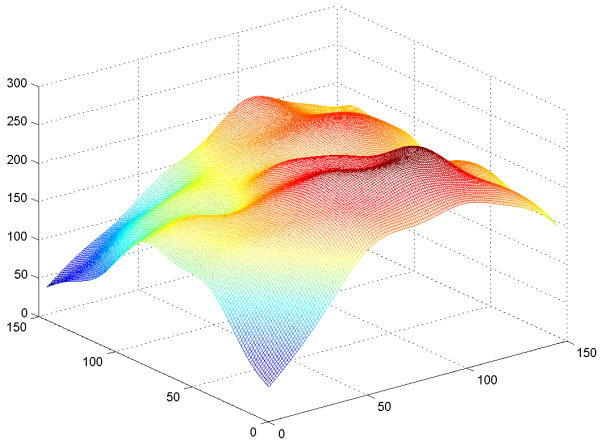

Figure 18: Gaussian RBF fitted to interferometric data (Figure 17).

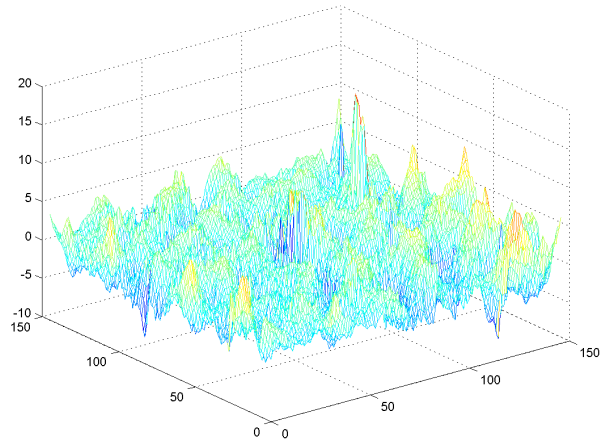

Figure 19: Residuals associated with the fit of a Gaussian RBF (Figure 18) to interferometric data (Figure 17).

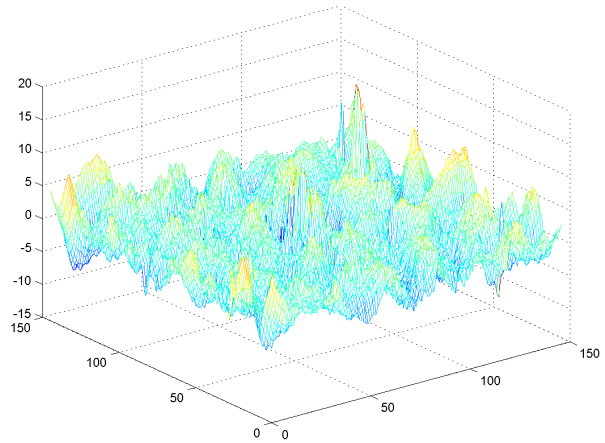

Figure 20: Residuals associated with the fit of a thin plate spline RBF to interferometric data (Figure 17).

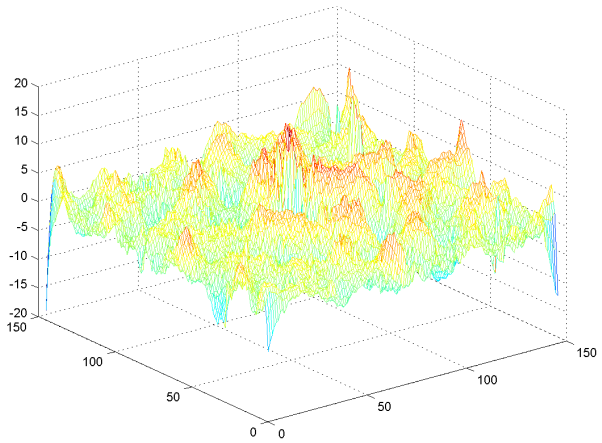

Figure 21: Residuals associated with the fit of a discrete orthogonal bivariate polynomial of total degree 13 to interferometric data (Figure 17).

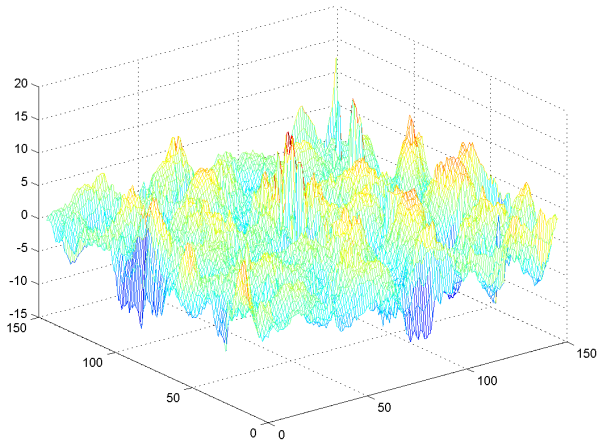

Figure 22: Residuals associated with the fit of an order 10 tensor product Chebyshev bivariate polynomial to interferometric data (Figure 17).

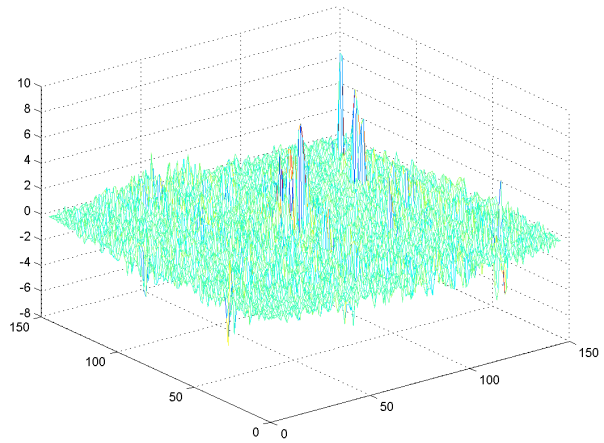

Figure 23: Residuals associated with the fit of an order 40 tensor product Chebyshev bivariate polynomial to interferometric data (Figure 17).

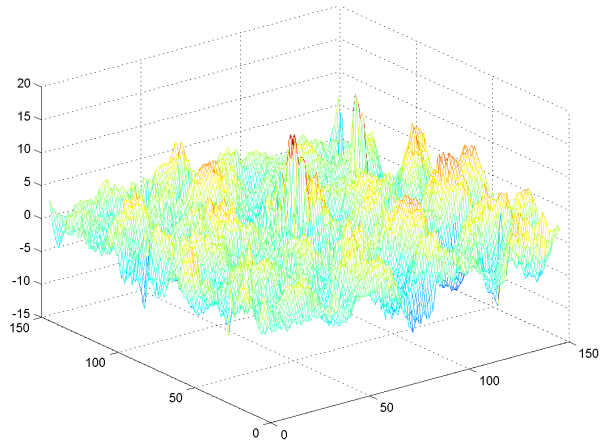

Figure 24: Residuals associated with the fit of a tensor product bicubic spline with 6 interior knots along each axis to interferometric data (Figure 17).

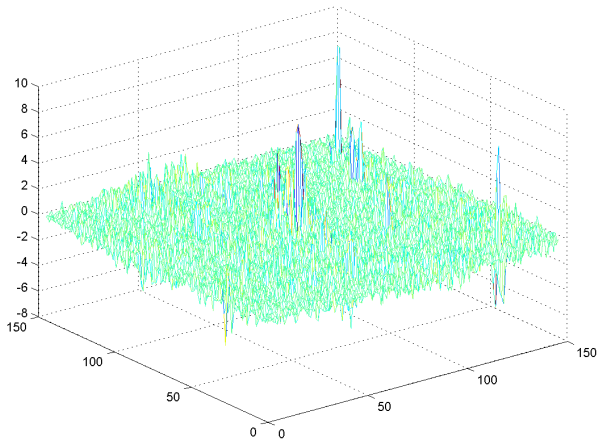

Figure 25: Residuals associated with the fit of a tensor product bicubic spline with 36 interior knots along each axis to interferometric data (Figure 17).

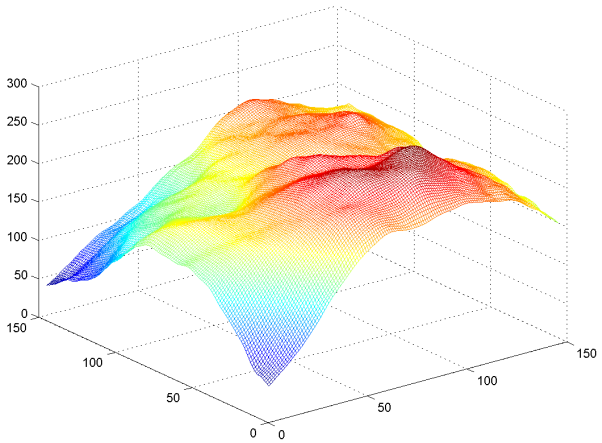

Figure 26: An order 40 tensor product Chebyshev bivariate polynomial fitted to interferometric data (Figure 17).

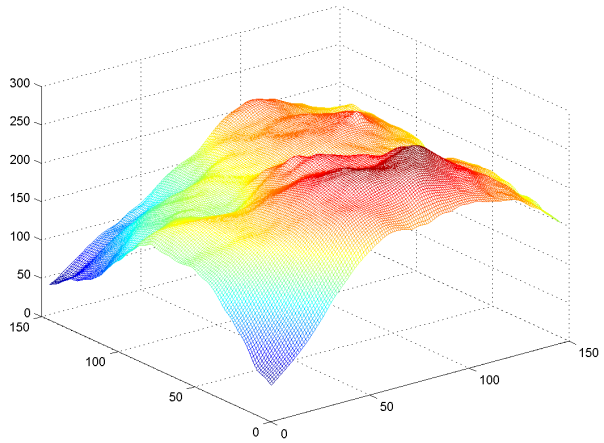

Figure 27: A fit of a tensor product bicubic spline with 36 interior knots along each axis to interferometric data (Figure 17).

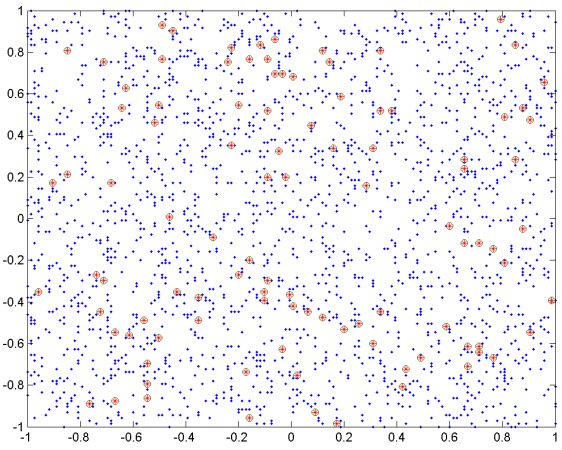

Figure 28: $xy-$coordinates of a subset of the interferometric data (Figure 17). The RBF centres are marked with an 'o'.

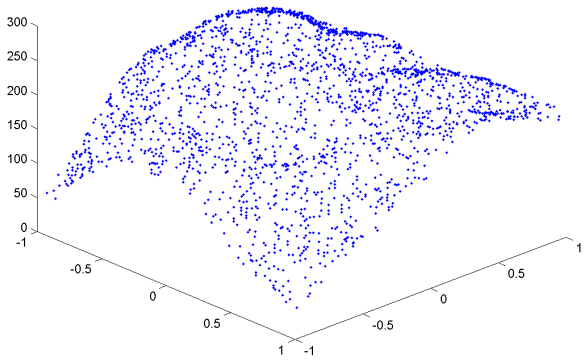

Figure 29: A randomly selected subset of the interferometric data (Figure 17).

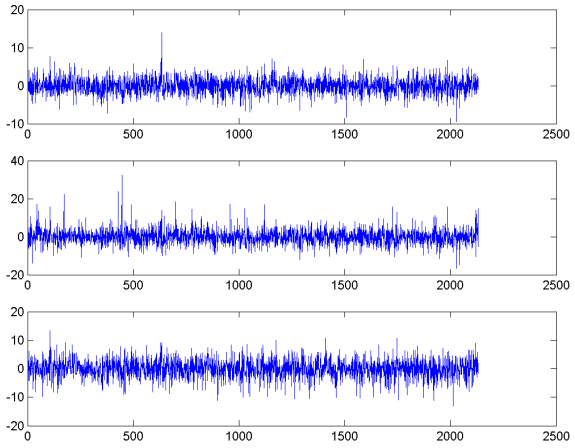

Figure 30: Residuals associated with the fit of a Gaussian RBF (top), thin plate spline RBF (middle) and bivariate polynomial of total degree 13 (bottom) to interferometric data (Figure 29).

10 Summary and concluding remarks

In this report we have been concerned with empirical models for multivariate data and systems. Polynomials and splines have been used very successfully for modelling functions of one variable. What are the options for multivariate data? We have found:

1. For data on a regular grid, approaches based on tensor product polynomials are generally very satisfactory, combining good approximation with very efficient approximation algorithms. Wavelets are important for multiresolution data, i.e., describing local behaviour at different scales or frequencies. Some classes of problems can be transformed so that tensor product methods can be applied.

2. For scattered data approximation, radial basis functions (RBF) have many attractive features: easy to implement, good approximations, available for higher dimensions. However, to become a standard tool for metrologists, further work would be beneficial to improve numerical stability and computational efficiency.

3. Nonuniform rational B-slines are important tools in computation geometry and coordinate metrology.

4. To the extent that nonlinear classification problems are required for metrology, neural networks and support vector machines provide an important capability, with considerable research being undertaken on statistical formulations, algorithms and applications.

In view of the potential of RBFs to fulfil an important role in modelling and representing multivariate data, considerable research is being undertaken to improve computation times (see, e.g., [3, 4]) and conditioning (e.g., [23, 38, 42]) as well as on the choice of centres (e.g., [41]). As RBF algorithms improve, they are likely to become a standard tool for multivariate modelling and data approximation in metrology.

Much of the underlying technology of support vector machines – statistical information theory, kernel methods, etc., – are of potential value to metrology, particular in situations in which the system under study is imperfectly understood, for example, in biotechnology.

Acknowledgment. This work has been supported by the National Measurement System Directorate of the UK Department of Trade and Industry as part of its NMS Software Support for Metrology programme.

References

[1] I. A. Anderson, M. G. Cox, and J. C. Mason. Tensor-product spline interpolation to data on or near a family of lines. *Numerical Algorithms*, 5:193–204, 1993.

[2] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[3] R. K. Beatson and W. A. Light. Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines. *IMA J. Num. Anal.*, 17:343–372, 1997.

[4] R. K. Beatson, W. A. Light, and S. Billings. Fast solution of radial basis function interpolation equation: domain decomposition methods. *SIAM J. Sci. Comp*, 22(5):1717–1740, 2001.

[5] C. M. Bishop. *Neural networks and pattern recognition.* Oxford Univeristy Press, 1995.

[6] C. M. Bishop, editor. *Neural networks and Machine Learning.* Springer, 1998. 1997 NATO Advanced Study Institute.

[7] D. S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[8] B. P. Butler, M. G. Cox, and A. B. Forbes. The reconstruction of workpiece surfaces from probe coordinate data. In R. B. Fisher, editor, *Design and Application of Curves and Surfaces*, pages 99–116. Oxford University Press, 1994. IMA Conference Series.

[9] B. P. Butler, M. G. Cox, A. B. Forbes, P. M. Harris, and G. J. Lord. Model validation in the context of metrology: a survey. Technical Report CISE 19/99, National Physical Laboratory, Teddington, 1999.

[10] M. G. Cox. The numerical evaluation of B-splines. *Journal of the Institute of Mathematics and its Applications*, 8:36–52, 1972.

[11] M. G. Cox. The numerical evaluation of B-splines. *J. Inst. Math. Appl.*, 10:134–149, 1972.

[12] M. G. Cox. The numerical evaluation of a spline from its B-spline representation. *Journal of the Institute of Mathematics and its Applications*, 21:135–143, 1978.

[13] M. G. Cox. The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA J. Numer. Anal.*, 1:3 – 22, 1981.

[14] M. G. Cox. Practical spline approximation. In P. R. Turner, editor, *Lecture Notes in Mathematics 965: Topics in Numerical Analysis*, pages 79–112, Berlin, 1982. Springer-Verlag.

[15] M. G. Cox. Practical spline approximation. In P. R. Turner, editor, *Notes in Mathematics 965: Topics in Numerical Analysis*, pages 79–112, Berlin, 1982. Springer-Verlag.

[16] M. G. Cox, M. P. Dainton, and P. M. Harris. Software Support for Metrology Best Practice Guide No. 6: Uncertainty and Statistical Modelling. Technical report, National Physical Laboratory, Teddington, 2001.

[17] M. G. Cox, A. B. Forbes, P. M. Fossati, P. M. Harris, and I. M. Smith. Techniques for the efficient solution of large scale calibration problems. Technical Report CMSC 25/03, National Physical Laboratory, Teddington, May 2003.

[18] M. G. Cox, A. B. Forbes, and P. M. Harris. Software Support for Metrology Best Practice Guide 4: Modelling Discrete Data. Technical report, National Physical Laboratory, Teddington, 2000.

[19] M. G. Cox, P. M. Harris, and P. D. Kenward. Fixed- and free-knot least-squares univariate data approximation by polynomial splines. NPL report CMSC 13/02, National Physical Laboratory, Teddington, 2002.

[20] M.G. Cox, A. B. Forbes, P. M. Harris, and G. N. Peggs. Determining cmm behaviour from measurements of standard artefacts. Technical Report CISE 15/98, National Physical Laboratory, Teddington, UK, 1998.

[21] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972.

[22] C. F. Dunkl and Y. Xu. *Orthogonal polynomials of several variables.* Cambride University Press, 2001.

[23] N. Dyn, D. Levin, and S. Rippa. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM J. Sci. Comp.*, 7:639–659, 1986.

[24] A. B Forbes. Model parametrization. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools for Metrology*, pages 29–47, Singapore, 1996. World Scientific.

[25] G. E. Forsythe. Generation and use of orthogonal polynomials for data fitting with a digital computer. *SIAM Journal*, 5:74–88, 1957.

[26] L. Fox and I. B. Parker. *Chebyshev polynomials in numerical analysis*. Oxford University press, 1968.

[27] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):845–848, 1989.

[28] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[29] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, third edition, 1996.

[30] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, Philadelphia, 1997.

[31] D. C. Handscombe and J. C. Mason. *Chebyshev Polnomials*. Chapman&Hall/CRC Press, London, 2003.

[32] P. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM J. Sci. Stat. Comp.*, 34(4):561–580, 1992.

[33] P. Hansen. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Num. Alg.*, 6:1–35, 1994.

[34] P. M. Harris. The use of splines in the modelling of a photodiode response. Technical Report DITC 88/87, National Physical Laboratory, Teddington, UK, 1987.

[35] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, second edition, 1999.

[36] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[37] M. Huhtanen and R. M Larsen. On generating discrete orthogonal bivariate polynomials. *BIT*, 42:393–407, 2002.

[38] W. Light. Computing with radial basis functions the Beatson-Light way! In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 220–235. University of Huddersfield, 2002.

[39] G. L. Lord, E. Pardo-Igúzquiza, and I. M. Smith. A practical guide to wavelets for metrology. Technical Report NPL Report CMSC 02/00, National Physical Laboratory, Teddington, June 2000.

[40] M. L. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[41] R. Morandi and A. Sestini. Geometric knot selection for radial scattered data approximation. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 244–251. University of Huddersfield, 2002.

[42] C. T. Mouat and R. K. Beatson. On the boundard over distance preconditioner for radial basis function interpolation. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 252–259. University of Huddersfield, 2002.

[43] M. J. L. Orr. Introduction to radial basis function networks. Technical report, Centre for Cognitive Science, University of Edinburgh, April 1996.

[44] M. J. L. Orr. Recent advances in radial basis function networks. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, June 1999.

[45] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-30, Institute for Computer Architecture and Software Technology (FIRST), Berlin, October 1998. Neural and Computational Learning (NeuroCOLT2) Technical Report Series.

[46] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandevalle. *Least squares support vector machines*. World Scientific, Singapore, 2002.

[47] G. Szego. *Orthogonal Polynomials*. American Mathematical Society, New York, 1959.

[48] A. N. Tikhonov and V. Y. Arsenin. *Solutions to Ill–Posed Problems*. Winston and Sons, Washington D. C., 1977.

[49] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[50] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[51] V. Vapnik and A. Chervonenkis. *Theory of pattern recognition*. Nauka, Moscow, 1974. In Russian. German Tranlation: *Theorie der Zeichenerkennung*, Akademie-Verklag, Berlin, 1979.

[52] G. Wahba. *Spline models for observational data*. SIAM, Philadelphia, 1990.

[53] H. L. Weinert. *Reproducing kernel Hilbert spaces*. Hutchinson Ross, Stroudsburg, PA, 1982.

[54] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *Trans. Math. Soft*, 23(4), 1997.