Computing 32, 319–342 (1984)

Computing

Algorithms for Smoothing Data on the Sphere with Tensor Product Splines

P. Dierckx, Heverlee

Received July 11, 1983

Abstract — Zusammenfassung

Algorithms for Smoothing Data on the Sphere with Tensor Product Splines. Algorithms are presented for fitting data on the sphere by using tensor product splines which satisfy certain boundary constraints. First we consider the least-squares problem when the knots are given. Then we discuss the construction of smoothing splines on the sphere. Here the knots are located automatically. A Fortran IV implementation of these two algorithms is described.

AMS Subject Classification (1980): 65D07.

Key words: Smoothing on the sphere, tensor product splines, B-splines, spherical harmonics.

Algorithmen für den Ausgleich über die Sphäre durch Tensorprodukt-Splines. Algorithmen werden vorgestellt für den Ausgleich über die Sphäre mit Hilfe von Tensorprodukt-Splines, die gewissen Randbedingungen genügen müssen. Erst untersuchen wir das Problem der kleinsten Quadrate, wenn die Knoten gegeben sind. Dann besprechen wir die Konstruktion von Ausgleichssplines über die Sphäre. Die Knoten werden hier automatisch lokalisiert. Eine Fortran-IV-Version dieser zwei Algorithmen wird beschrieben.

1. Introduction

In this paper we consider the problem of smoothing data on the sphere. It has important applications, especially in meteorology. Suppose e.g. that some geopotential height (the height above sea level at which the pressure has a certain value) is measured (with noise) at a number of weather stations around the world, then it is useful to find a smooth function $r(\theta, \phi)$, defined on the surface of the earth ($\theta =$ latitude, $\phi =$ longitude) by which an estimate of this quantity can be computed at any position (θ, ϕ).

Other applications can be found in medicine and computer aided design where one is interested in obtaining a mathematical description of a closed surface in the form

$$\begin{aligned} x &= r(\theta, \phi)\sin\theta\cos\phi && \\ y &= r(\theta, \phi)\sin\theta\sin\phi && 0 \le \theta \le \pi \\ z &= r(\theta, \phi)\cos\theta && 0 \le \phi \le 2\pi \end{aligned} \tag{1.1}$$

In order to have a satisfactory solution to these problems, the function $r(\theta, \phi)$ must fulfill a number of specific conditions. First of all, $r(\theta, \phi)$ must be periodic in ϕ. Besides there is the "pole problem", For $\theta=0$ and $\theta=\pi$, the value of $r(\theta, \phi)$ must be independent of ϕ. Moreover, if we wish that the function $z=f(x, y)$ which is implicitly defined by (1.1), is sufficiently smooth at the poles, then additionally conditions on the derivatives of r must be imposed, e.g.

$$\frac{\partial r(0, \phi)}{\partial \theta}=\cos \phi \frac{\partial r(0,0)}{\partial \theta}+\sin \phi \frac{\partial r(0, \pi/2)}{\partial \theta} \tag{1.2}$$

and $$0 \leq \phi \leq 2\pi$$

$$\frac{\partial r(\pi, \phi)}{\partial \theta}=\cos \phi \frac{\partial r(\pi,0)}{\partial \theta}+\sin \phi \frac{\partial r(\pi, \pi/2)}{\partial \theta} \tag{1.3}$$

in order to have C^1 continuity.

Spherical harmonics [1] provide these properties by combining the periodic Fourier functions and the associated Legendre functions. Given a set of data points (θ_q, ϕ_q), $q=1,2,\ldots,m$, the most commonly used approximation method is then to fit in the least-squares sense (with N fixed) a function of the form

$$R(\theta, \phi)=\sum_{n=0}^{N} \sum_{l=0}^{n} \bar{P}_n^l(\cos \theta)\left(a_{n,l} \cos(l\phi)+b_{n,l} \sin(l\phi)\right) \tag{1.4}$$

where the $\bar{P}_n^l(x)$ denote the normalized associated Legendre functions, i.e.

$$\bar{P}_n^l(x)=\left[\frac{(2n+1)}{2} \frac{(n-l)!}{(n+l)!}\right]^{1/2} P_n^l(x) \tag{1.5}$$

with

$$P_n^l(x)=(1-x^2)^{l/2}\left(\frac{d^l}{dx^l}\right) P_n(x) \tag{1.6}$$

and $P_n(x)$ the Legendre polynomial of degree n.

Recently Wahba [15] has described another approximation method in which the notion of periodic polynomial splines on the circle and thin plate splines on Euclidean d-space is generalized to the sphere. The approximating functions appear to have interesting statistical properties. A disadvantage however is that the number of coefficients in the representation of these splines, equals the number of data points. Therefore, if m is large, the method seems less appropriate, also due to the lack of sparsity of the system from which the coefficients must be calculated.

Boydstun, Armstrong and Bookstein [3] have considered tensor product splines in their maximum reach estimation techniques. However no attempt was made to deal with the "pole problem" nor to take full advantage from the use of B-splines.

In this paper we will impose conditions on these tensor product splines which will result in a sufficiently smooth approximation at the poles. Also we will demonstrate how the least-squares problem with these constrained splines can be solved efficiently. Finally we will generalize the smoothing criterion discribed in [6]. This will allow us to automate the determination of knots of such splines.

2. Tensor Product Splines on the Sphere

In our approximation problem a closed rectangular domain $D=[0,\pi]\times[0,2\pi]$ is concerned (Fig. 1).

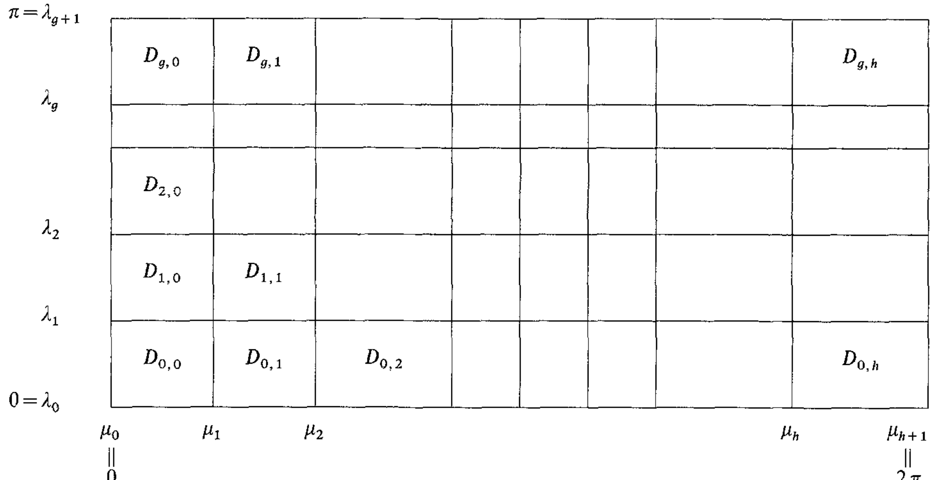

Fig. 1. The rectangle D, divided by knots λ_i, μ_j into panels $D_{i,j}$

Consider the strictly increasing sequence of real numbers

$$0=\lambda_0<\lambda_1<\ldots<\lambda_g<\lambda_{g+1}=\pi \tag{2.1}$$

$$0=\mu_0<\mu_1<\ldots<\mu_h<\mu_{h+1}=2\pi \tag{2.2}$$

then the function $s(\theta,\phi)$ is called a spline of degree k in θ and l in ϕ, with knots λ_i, $i=1,2,\ldots,g$ in the θ-direction and $\mu_j, j=1,2,\ldots,h$ in the ϕ-direction if the following conditions are satisfied.

(i) On any subrectangle $D_{i,j}=[\lambda_i,\lambda_{i+1}]\times[\mu_j,\mu_{j+1}]$, $i=0,1,\ldots,g$; $j=0,1,\ldots,h$, $s(\theta,\phi)$ is given by a polynomial of degree k in θ and l in ϕ.

(ii) All derivatives $\partial^{i+j}s(\theta,\phi)/\partial\theta^i\,\partial\phi^j$ for $0\le i\le k-1$ and $0\le j\le l-1$ are continuous in D.

If we introduce a number of additional knots satisfying

$$\begin{aligned}&\lambda_{-k}\le\lambda_{-k+1}\le\ldots\le\lambda_{-1}\le 0\\&\pi\le\lambda_{g+2}\le\ldots\le\lambda_{g+k}\le\lambda_{g+k+1}\end{aligned} \tag{2.3}$$

$$\begin{aligned}&\mu_{-l}\le\mu_{-l+1}\le\ldots\le\mu_{-1}\le 0\\&2\pi\le\mu_{h+2}\le\ldots\le\mu_{h+1}\le\mu_{h+l+1}\end{aligned} \tag{2.4}$$

but which are further arbitrary, every such spline on D can uniquely be expressed as

$$s(\theta,\phi)=\sum_{i=-k}^{g}\sum_{j=-l}^{h}c_{i,j}M_{i,k+1}(\theta)N_{j,l+1}(\phi) \tag{2.5}$$

where $M_{i,k+1}(\theta)$ and $N_{j,l+1}(\phi)$ are normalized B-splines, i.e.

$$M_{i,k+1}(\theta)=(\lambda_{i+k+1}-\lambda_i)\,\Delta_t^{k+1}(\lambda_i,\lambda_{i+1},\ldots,\lambda_{i+k+1})\,g_k(t;\theta) \tag{2.6}$$

$$N_{j,l+1}(\phi)=(\mu_{j+l+1}-\mu_j)\,\Delta_t^{l+1}(\mu_j,\mu_{j+1},\ldots,\mu_{j+l+1})\,g_l(t;\phi) \tag{2.7}$$

with

$$\begin{aligned} g_m(t;x)&=(t-x)_+^m=(t-x)^m && \text{if } t\geq x \\ &=0 && \text{if } t<x \end{aligned} \tag{2.8}$$

and where $\Delta_t^m(z_i,z_{i+1},\ldots,z_{i+m})\,f(t)$ stands for the m-th divided difference of the function $f(t)$ on the points $z_i,z_{i+1},\ldots,z_{i+m}$.

Normalized B-splines enjoy some interesting properties such as

$$M_{i,k+1}(\theta)=0 \quad \text{if } \theta<\lambda_i \text{ or } \theta>\lambda_{i+k+1} \tag{2.9}$$

$$N_{j,l+1}(\phi)=0 \quad \text{if } \phi<\mu_j \text{ or } \phi>\mu_{j+l+1} \tag{2.10}$$

$$M'_{i,k+1}(\theta)=k\left[\frac{M_{i,k}(\theta)}{\lambda_{i+k}-\lambda_i}-\frac{M_{i+1,k}(\theta)}{\lambda_{i+k+1}-\lambda_{i+1}}\right] \tag{2.11}$$

$$N'_{j,l+1}(\phi)=l\left[\frac{N_{j,l}(\phi)}{\mu_{j+l}-\mu_j}-\frac{N_{j+1,l}(\phi)}{\mu_{j+l+1}-\mu_{j+1}}\right] \tag{2.12}$$

$$\sum_i M_{i,k+1}(\theta)\equiv\sum_j N_{j,l+1}(\phi)\equiv 1 \tag{2.13}$$

and they can be evaluated in a very stable way using the recurrence scheme of de Boor [2] and Cox [4].

In our application we are interested in a spline function $s(\theta,\phi)$ which is periodic in ϕ. If we choose the boundary knots (2.4) in the following way

$$\begin{aligned} &\mu_{-j}=\mu_{h+1-j}-2\pi \\ \text{and}\quad &\mu_{j+h+1}=\mu_j+2\pi \qquad j=1,2,\ldots,l \end{aligned} \tag{2.14}$$

then from (2.7) and the properties of divided differences it follows that (see e.g. [5] and [7])

$$N_{-j,l+1}(\phi)=N_{-j+h+1,l+1}(\phi+2\pi) \qquad j=1,2,\ldots,l \tag{2.15}$$

Therefore, by taking account of property (2.10) and (2.12), we find that

$$\frac{\partial^j s(\theta,0)}{\partial\phi^j}=\frac{\partial^j s(\theta,2\pi)}{\partial\phi^j} \qquad \begin{aligned} &j=0,1,\ldots,l-1 \\ &0\leq\theta\leq\pi \end{aligned} \tag{2.16}$$

if we impose the conditions

$$c_{i,h+1-j}=c_{i,-j} \qquad \begin{aligned} &i=-k,-k+1,\ldots,h \\ &j=1,2,\ldots,l \end{aligned} \tag{2.17}$$

The boundary knots (2.3) are chosen as follows

$$\begin{aligned} &\lambda_{-i}=0 \\ &\lambda_{g+1+i}=\pi \end{aligned} \qquad i=1,2,\ldots,k. \tag{2.18}$$

Then from (2.6) and the definition of divided differences on coincident points, it follows that

$$M_{i,k+1}(0)=\delta_{i,-k} \tag{2.19}$$

and

$$M_{i,k+1}(\pi)=\delta_{i,g} \tag{2.20}$$

with $\delta_{i,j}$ the Kronecker delta. Consequently, if we impose that

$$c_{-k,j}=c_{-k,-l}(=\alpha) \qquad j=-l+1,-l+2,\ldots,h \tag{2.21}$$

$$c_{g,j}=c_{g,-l}(=\beta) \tag{2.22}$$

it follows from (2.5) and (2.13) that the value of $s(\theta,\phi)$ is independent of ϕ at the poles, i.e.

$$s(0,\phi)=\alpha \qquad 0\le\phi\le 2\pi \tag{2.23}$$

$$s(\pi,\phi)=\beta \tag{2.24}$$

The conditions (1.2) and (1.3) needed for having C^1 continuity at the poles, cannot be fulfilled exactly if we use tensor product splines. However they can be satisfied approximately if we replace $\cos\phi$ and $\sin\phi$ by their periodic spline interpolants $C(\phi)$ and $S(\phi)$, i.e.

$$C(\phi)=\sum_{j=-l}^{h} d_j N_{j,l+1}(\phi) \tag{2.25}$$

with

$$C(\mu_j)=\cos(\mu_j), \qquad j=0,1,\ldots,h \tag{2.26}$$

$$d_{-j}=d_{h+1-j}, \qquad j=1,2,\ldots,l \tag{2.27}$$

and

$$S(\phi)=\sum_{j=-l}^{h} e_j N_{j,l+1}(\phi) \tag{2.28}$$

with

$$S(\mu_j)=\sin(\mu_j), \qquad j=0,1,\ldots,h \tag{2.29}$$

$$e_{-j}=e_{h+1-j}, \qquad j=1,2,\ldots,l. \tag{2.30}$$

From (2.5), (2.11) and (2.9) it follows that

$$\frac{\partial s(\theta,\phi)}{\partial\theta}=k\sum_{i=-k+1}^{g}\sum_{j=-l}^{h}\frac{c_{i,j}-c_{i-1,j}}{\lambda_{i+k}-\lambda_i}M_{i,k}(\theta)N_{j,l+1}(\phi) \tag{2.31}$$

and consequently from (2.19), (2.18) and (2.21) that

$$\frac{\partial s(0,\phi)}{\partial\theta}=\frac{k}{\lambda_1}\sum_{j=-l}^{h}(c_{-k+1,j}-\alpha)N_{j,l+1}(\phi). \tag{2.32}$$

Therefore, by replacing r by s and $\cos\phi$ and $\sin\phi$ by $C(\phi)$ and $S(\phi)$, condition (1.2) finally becomes

$$c_{-k+1,j}=\alpha+\gamma_1 d_j+\gamma_2 e_j \qquad j=-l,\ldots,h \tag{2.33}$$

where

$$\gamma_1 = \frac{\lambda_1}{k} \frac{\partial s(0,0)}{\partial \theta} \tag{2.34}$$

and

$$\gamma_2 = \frac{\lambda_1}{k} \frac{s(0,\pi/2)}{\partial \theta}. \tag{2.35}$$

Similarly, condition (1.3) becomes

$$c_{g-1,j} = \beta + \delta_1 d_j + \delta_2 e_j \qquad j = -l, \ldots, h \tag{2.36}$$

with

$$\delta_1 = \frac{(\lambda_g - \pi)}{k} \frac{\partial s(\pi,0)}{\partial \theta} \tag{2.37}$$

and

$$\delta_2 = \frac{(\lambda_g - \pi)}{k} \frac{s(\pi,\pi/2)}{\partial \theta}. \tag{2.38}$$

A bivariate spline function (2.5) which satisfies the conditions (2.17), (2.21), (2.22), (2.33) and (2.36) will be called a tensor product spline on the sphere and denoted by $\hat{s}(\theta,\phi)$. Substituting all these conditions into (2.5) yields an explicit representation

$$\begin{aligned} \hat{s}(\theta,\phi) = {} & \alpha\big(M_{-k,k+1}(\theta) + M_{-k+1,k+1}(\theta)\big) + \big(\gamma_1 C(\phi) + \gamma_2 S(\phi)\big) M_{-k+1,k+1}(\theta) \\ & + \sum_{i=-k+2}^{g-2} \sum_{j=-l}^{h-l} c_{i,j} M_{i,k+1}(\theta) \hat{N}_{j,l+1}(\phi) \\ & + \big(\delta_1 C(\phi) + \delta_2 S(\phi)\big) M_{g-1,k+1}(\theta) + \beta\big(M_{g,k+1}(\theta) + M_{g-1,k+1}(\theta)\big) \end{aligned} \tag{2.39}$$

where [7]

$$\hat{N}_{j,l+1}(\phi) = \sum_{u=0}^{v} N_{j+u(h+1),l+1}(\phi) \qquad j = -l, \ldots, h-l \tag{2.40}$$

with

$$v = \left\lceil \frac{h-j}{h+1} \right\rceil. \tag{2.41}$$

($\lceil x \rceil$ stands for the greatest integer less than or equal to x); or less generally (see (2.15)), if the number of knots h is greater than $l-2$

$$\begin{aligned} \hat{N}_{j,l+1}(\phi) &= N_{j,l+1}(\phi) + N_{j+h+1,l+1}(\phi) \qquad & j = -l, \ldots, -1 \\ &= N_{j,l+1}(\phi) & j = 0, 1, \ldots, h-l. \end{aligned} \tag{2.42}$$

From (2.39) it follows that the dimension $\hat{d}$ of the vector space of tensor splines on the sphere with knots λ_i, $i = 1, 2, \ldots, g$; μ_j, $j = 1, 2, \ldots, h$ equals

$$\hat{d} = 6 + (g+k-3)(h+1) \tag{2.43}$$

if we suppose that $g + k \geq 3$. This is quite less than the dimension

$$d = (g+k+1)(h+l+1) \tag{2.44}$$

for the vector space of general splines with these knots, as can easily be deduced from (2.5).

3. Least-Squares Tensor Product Splines

Given the function values r_q at some points (θ_q, ϕ_q), $q=1,2,\dots,m$ scattered arbitrarily in D, together with a set of positive weights w_q, $q=1,2,\dots,m$, we wish to calculate in an efficient way, the weighted least-squares spline $\hat{s}(\theta,\phi)$ of given degrees (k,l) with given knots λ_i, μ_j.

3.1. General Least-Squares Splines [9]

Let us first recall how the general least-squares spline, i.e. the spline $s(\theta,\phi)$ for which

$$\sigma = \sum_{q=1}^{m} w_q \left(r_q - s(\theta_q, \phi_q)\right)^2 \tag{3.1}$$

is minimal, can be determined efficiently.

Using the representation (2.5), the least-squares problem simply results in the computation of the coefficients $c_{i,j}$ as the least-squares solution of the overdetermined system

$$\sqrt{w_q} \sum_{i=-k}^{g} \sum_{j=-l}^{h} c_{i,j} M_{i,k+1}(\theta_q) N_{j,l+1}(\phi_q) = \sqrt{w_q}\, r_q \qquad q=1,2,\dots,m. \tag{3.2}$$

This may be written in matrix notation as

$$A\bar{c} = \bar{e} \tag{3.3}$$

where $\bar{c}$ denotes the vector containing the coefficients $c_{i,j}$ in the order $j=-l, -l+1, \dots, h$ for fixed i, in which $i=-k,\dots,g$ in turn. From (2.9) and (2.10) it follows that the observation matrix A contains many zeros. Especially, if we arrange the data points (θ_q, ϕ_q) according to the panel $D_{u,v}$ containing them, in the order $v=0,1,\dots,h$ for fixed u, in which $u=0,1,\dots g$ in turn, the matrix A takes the following band form

$$A = \begin{vmatrix} A_{0,0} & A_{0,1} & A_{0,2} & \cdots & A_{0,k} & & \\ & A_{1,1} & A_{1,2} & \cdots & A_{1,k} & A_{1,k+1} & \\ & & A_{2,2} & \cdots & A_{2,k} & A_{2,k+1} & A_{2,k+2} \\ & & \cdots & \cdots & \cdots & \cdots & \\ & & & & & A_{g,g} \cdots & A_{g,k+g} \end{vmatrix}. \tag{3.4}$$

Here, in turn, each matrix $A_{u,v}$ is a rectangular bandmatrix with $h+l+1$ columns and a number of rows m_u equal to the total number of data points in the panels $D_{u,0}, D_{u,1}, \dots, D_{u,h}$. In fact, this matrix has the values

$$\sqrt{w_q} M_{v-k,k+1}(\theta_q) N_{j,l+1}(\phi_q), \; j=-l, \; -l+1, \dots, h$$

for these latter data points (θ_q, ϕ_q) as its elements. So, each matrix $A_{u,v}$ has the typical structure

$$A_{u,v}=\left|\begin{array}{ccccccccc}
\multicolumn{4}{c}{\overbrace{x\quad x\quad x\quad x}^{l+1}} & & & & & \\
x & x & x & x & & & & & \\
 & x & x & x & x & & & & \\
 & x & x & x & x & & & & \\
 & x & x & x & x & & & & \\
 & & x & x & x & x & & & \\
 & & & - & - & - & - & & \\
 & & & & & x & x & x & x \\
 & & & & & x & x & x & x
\end{array}\right|
\begin{array}{l}
(\theta_q,\phi_q)\in D_{u,0} \\
\\
\\
(\theta_q,\phi_q)\in D_{u,1} \\
\\
\\
\\
\\
(\theta_q,\phi_q)\in D_{u,h}
\end{array} \tag{3.5}$$

if we denote the non-zero elements by x.

So, from (3.5) we can easily find that the bandwidth b of matrix A is

$$b=k(h+l+1)+l+1 \tag{3.6}$$

and not $(k+1)\,(h+l+1)$ as might be expected from (3.4).

We can use then an orthogonalization method with Givens rotations without square roots [8] to solve (3.3). Indeed, with this method, the zeros already present in A and the special band structure are readily exploited to reduce computation time and memory requirements [5]. Also, eventually the independent variables θ and ϕ (and the corresponding parameters) will be switched if this can reduce the bandwidth b, i.e. if $l\cdot g<k\cdot h$.

Finally we mention that the matrix A possibly is (numerically) rank deficient. In that case the minimal-length solution of (3.3) can be computed (see for example [12]).

3.2. Least-Squares Splines on the Sphere

We will show now, how the schemes described above must be adjusted in case a least-squares spline on the sphere must be computed, i.e. a spline $\hat{s}(\theta,\phi)$ for which

$$\hat{\sigma}=\sum_{q=1}^{m} w_q\big(r_q-\hat{s}(\theta_q,\phi_q)\big)^2 \tag{3.7}$$

is minimal. Using the representation (2.39) this problem then results in the solution of a system

$$\hat{A}\,\bar{C}=\bar{e} \tag{3.8}$$

with $\bar{C}$ the vector containing the coefficients of $\hat{s}(\theta,\phi)$ in the order

$$\alpha,\gamma_1,\gamma_2,\,c_{-k+2,\,-l},\,c_{-k+2,\,-l+1},\ldots,\,c_{-k+2,h-l},\,c_{-k+3,\,-l},\ldots,$$
$$c_{-k+3,h-l},\ldots,\,c_{g-2,\,-l},\ldots,c_{g-2,h-l},\,\delta_1,\,\delta_2,\,\beta.$$

If we consider the data points (θ_q,ϕ_q) in the same order as described in section 3.1, the observation matrix $\hat{A}$ now takes the form

$$\hat{A}=\left|\begin{array}{llllllll} B_0 & \hat{A}_{0,2} \dots \hat{A}_{0,k} & & & & & & \\ B_1 & \hat{A}_{1,2} \dots \hat{A}_{1,k} & \hat{A}_{1,k+1} & & & & & \\ & \hat{A}_{2,2} \dots \hat{A}_{2,k} & \hat{A}_{2,k+1} & \hat{A}_{2,k+2} & & & & \\ & & \dots\dots\dots\dots & & & & & \\ & & & \hat{A}_{g-1,g-1} & \hat{A}_{g-1,g} & \dots & \hat{A}_{g-1,g+k-2} & B_{g-1} \\ & & & & \hat{A}_{g,g} & \dots & \hat{A}_{g,g+k-2} & B_g \end{array}\right| \tag{3.9}$$

Each matrix $\hat{A}_{u,v}$ has still m_u rows but the number of columns now only is $h+1$. $\hat{A}_{u,v}$ contains the values

$$\sqrt{w_q}\, M_{v-k,k+1}(\theta_q)\, \hat{N}_{j,l+1}(\phi_q),\ j=-l,-l+1,\dots,h-l$$

for the data points (θ_q, ϕ_q) belonging to the panels $D_{u,0}, D_{u,1}, \dots, D_{u,h}$. So, each $\hat{A}_{u,v}$ still contains many zeros but from (2.40)–(2.42) it follows that its band structure is cyclic, particularly of the form

$$\hat{A}_{u,v}=\left|\begin{array}{cccccccc} x & x & x & x & & & & \\ x & x & x & x & & & & \\ & x & x & x & x & & & \\ & x & x & x & x & & & \\ & x & x & x & x & & & \\ & & - & - & - & - & & \\ & & & & x & x & x & x \\ & & & & x & x & x & x \\ x & & & & & x & x & x \\ x & & & & & x & x & x \\ x & x & & & & & x & x \\ x & x & & & & & x & x \\ x & x & x & & & & & x \\ x & x & x & & & & & x \end{array}\right| \begin{array}{l} (\theta_q,\phi_q)\in D_{u,0} \\ \\ \\ (\theta_q,\phi_q)\in D_{u,1} \\ \\ - - - - \\ \\ (\theta_q,\phi_q)\in D_{u,h-l} \\ \\ (\theta_q,\phi_q)\in D_{u,h-l+1} \\ \\ - - - - \\ \\ (\theta_q,\phi_q)\in D_{u,h} \end{array} \tag{3.10}$$

So, here the zeros of $\hat{A}_{u,v}$ do not further reduce the total bandwidth $\hat{b}$ of $\hat{A}$. Since the matrices B_u are full matrices with m_u rows and 3 columns it follows from (3.9) that

$$\begin{aligned} \hat{b} &= (k+1)(h+1) && \text{if} \quad g>3 \\ &= 3+k(h+1) && \text{if} \quad g=3 \\ &= \hat{d} && \text{if} \quad g<3. \end{aligned} \tag{3.11}$$

4. Smoothing Tensor Product Splines

4.1. General Smoothing Splines

In [6] we have described an algorithm for fitting a bivariate smoothing spline to data given at arbitrary points. The idea is to determine a spline approximation $s(\theta, \phi)$ trying to find a compromise between the following objectives.

(i) The prescribed value r_q should be fitted closely enough.

(ii) The approximating spline should be smooth enough in the sense that the discontinuities in its derivatives are as small as possible.

In order to formulate this criterion mathematically some measure of smoothness and some measure of closeness of fit must be introduced. For the latter the weighted sum of squared residuals σ (3.1) can be taken.

As a suitable smoothing norm, i.e. a measure of the lack of smoothness

$$\eta=\sum_{q=1}^{g}\sum_{j=-l}^{h}\left(\sum_{i=-k}^{g} a_{i,q}c_{i,j}\right)^2+\sum_{r=1}^{h}\sum_{i=-k}^{g}\left(\sum_{j=-l}^{h} b_{j,r}c_{i,j}\right)^2 \tag{4.1}$$

is proposed where $a_{i,q}$ and $b_{j,r}$ denote the discontinuity jumps of the derivatives of the B-splines at the knots, i.e.

$$a_{i,q}=M_{i,k+1}^{(k)}(\lambda_q+0)-M_{i,k+1}^{(k)}(\lambda_q-0) \tag{4.2}$$

and

$$b_{j,r}=N_{j,l+1}^{(l)}(\mu_r-0)-N_{j,l+1}^{(l)}(\mu_r-0). \tag{4.3}$$

Then the approximation criterion is formulated mathematically as follows

$$\text{Minimize } \eta(\bar{c}) \tag{4.4}$$

$$\text{Subject to the constraint } \sigma(\bar{c})\leq S \tag{4.5}$$

where S is a nonnegative constant which must be supplied by the user to control the extent of smoothing and therefore is called the smoothing factor.

Consider the following system of equations

$$\begin{aligned}
&\sqrt{w_q}\sum_{i=-k}^{g}\sum_{j=-l}^{h} c_{i,j}M_{i,k+1}(\theta_q)N_{j,l+1}(\phi_q)=\sqrt{w_q}\,r_q,\ q=1,2,\dots,m\\
&\frac{1}{\sqrt{p}}\sum_{i=-k}^{g} a_{i,q}c_{i,j}=0\quad q=1,2,\dots,g;\ j=-l,\dots,h\\
&\frac{1}{\sqrt{p}}\sum_{j=-l}^{h} b_{j,r}c_{i,j}=0\quad r=1,2,\dots,h;\ i=-k,\dots,g
\end{aligned} \tag{4.6}$$

which depends on the parameter p in the sense that, being overdetermined, it is supposed to be solved in the least-squares sense, i.e. such that $\sigma(\bar{c})+p^{-1}\eta(\bar{c})$ is minimal with σ and η given by (3.1) and (4.1).

Let $s_p(\theta,\phi)$ denote the corresponding spline. It is then easily verified that, using the method of Lagrange, problem (4.4)–(4.5) simply results in the computation of the B-spline coefficients $\bar{c}$ from (4.6) when p is given the value of the positive root of the equation $F(p)=S$ with

$$F(p)=\sum_{q=1}^{m} w_q\left(r_q-s_p(\theta_q,\phi_q)\right)^2. \tag{4.7}$$

The smoothing spline $s_p(\theta, \phi)$ has the following properties [6]:

(i) To each positive p there corresponds a single spline $s_p(\theta, \phi)$, the B-spline coefficients of which are the (minimal-length) least-squares solution of (4.6).

(ii) If p tends to infinity, (4.6) results in (3.2) and $s_p(\theta, \phi)$ simply becomes the least-squares spline $S_{g,h}(\theta, \phi)$ with fixed knots. Let

$$F_{g,h}(\infty) = \sum_{q=1}^{m} w_q \left(r_q - S_{g,h}(\theta_q, \phi_q)\right)^2. \tag{4.8}$$

(iii) If p tends to zero, the second and third set of equations in (4.6) have much more weight than the first set, so that, if possibly they will have to be satisfied exactly. Now these equations simply express that the spline has no discontinuities in its derivatives as can be deduced from (2.5), (4.2) and (4.3). Therefore we may conclude that if p tends to zero, $s_p(\theta, \phi)$ will tend to the least-squares polynomial $P_{k,l}(\theta, \phi)$ of degree k in θ and l in ϕ. Let

$$F(0) = \sum_{q=1}^{m} w_q \left(r_q - P_{k,l}(\theta_q, \theta_q)\right)^2. \tag{4.9}$$

(iv) $F(p)$ is a continuous, strictly decreasing and convex function for $p > 0$.

Therefore we know that, once a set of knots is found such that

$$F_{g,h}(\infty) \leq S < F(0) \tag{4.10}$$

there exists a single spline $s_p(\theta, \phi)$ with these knots for which $F(p) = S$. This value of p can then be determined iteratively by means of a rational interpolation scheme [6].

For finding a set of knots which satisfies (4.10) we proceed in the following manner. First we determine the least-squares polynomial $P_{k,l}(\theta, \phi)$ which simply is the least-squares spline $S_{0,0}(\theta, \phi)$. If $F_{0,0}(\infty) \leq S$ this polynomial is a solution of our problem. However, usually we will find that $F_{0,0}(\infty) > S$. In that case we determine successive least-squares splines $S_{g,h}(\theta, \phi)$ with an increasing number of knots. At each iteration we locate one additional knot there where the fit $S_{g,h}(\theta, \phi)$ is particularly poor (for more details, see [6]). So the strategy for locating knots is adaptive in the sense that there will be more knots if S is small and less if it is large and also in the sense that the spline will have more knots in those regions where the function underlying the data is difficult to approximate then where it has a smooth behaviour.

4.2. Smoothing Splines on the Sphere

To obtain a smoothing spline on the sphere we will simply solve the problem (4.4)–(4.5) and the resulting system (4.6) subject to the additional constraints (2.17), (2.21), (2.22), (2.33) and (2.36). After eliminating these constraint equations, a system is obtained in the coefficients $\alpha, \gamma_1, \gamma_2, c_{i,j}, i = -k+2, \ldots g-2; j = -l, \ldots, h-l, \delta_1, \delta_2$ and β of a smoothing spline $\hat{s}_p(\theta, \phi)$.

E. g. from (2.21) it follows that the equations with $i = -k$ in the third set of (4.6) can be written as

$$\frac{1}{\sqrt{p}} \alpha \sum_{j=-l}^{h} b_{j,r} = 0 \quad r = 1, 2, \ldots, h \tag{4.11}$$

Regardless the value of α, these equations will always be satisfied exactly as follows from (2.13) and (4.3). Similarly, we can also drop the equations with $i = g$ as follows from (2.13), (2.22) and (4.3). As concerned the middle equations in (4.6) we can easily see from (2.17) that those for $j = -l, -l+1, \ldots, -1$ are now identical to those for $j = -l+h+1, -l+h+2, \ldots, h$. Therefore, they will be considered only once (so our smoothing norm η is slightly adapted).

For the determination of the coefficients of $\hat{s}_p(\theta, \phi)$ from the remaining and adapted equations (4.6) we can proceed then in a similar way as described in section 3. It can be verified that the middle equations, particularly the one with $q = 2$ and $j = h - l$, will slightly increase the bandwidth $\hat{b}$ (3.11) of the matrix of our system to become

$$\begin{aligned} \hat{b} &= (k+1)(h+1)+3 \quad &\text{if} \quad g > 3 \\ &= \hat{d} &\text{if} \quad g \leq 3 \end{aligned} \tag{4.12}$$

since $a_{i,q} = 0$ only if $i < q - k - 1$ or $i > q$.

The iterative determination of the root of $F(p) = S$, now with $\hat{s}_p$ instead of s_p in the definition (4.7) of F, can be carried out in the same way as for the general smoothing spline, since $\hat{s}_p$ has similar properties as s_p. Of course, if p tends to infinity, $\hat{s}_p(\theta, \phi)$ now becomes a least-squares tensor spline $\hat{S}_{g,h}(\theta, \phi)$ on the sphere. And if p tends to zero, $\hat{s}_p(\theta, \phi)$ becomes a least-squares polynomial $\hat{P}_{k,0}(\theta, \phi)$ still of degree k in θ but now of degree 0 in the variable ϕ as follows from the periodicity property (2.16). So, also taking account of the C^1 conditions (1.2) and (1.3) which will be satisfied exactly by this polynomial, we may just as well write that

$$\hat{P}_{k,0}(\theta, \phi) \equiv p_k(\theta) \qquad 0 \leq \theta \leq \pi, \ 0 \leq \phi \leq 2\pi \tag{4.13}$$

with $p_k(\theta)$ a polynomial of degree k, satisfying

$$p_k'(0) = p_k'(\pi) = 0. \tag{4.14}$$

Also, the strategy for finding a suitable set of knots can readily be adopted. If $F(0)$, now corresponding to the least-squares polynomial $\hat{P}_{k,0}(\theta, \phi)$, is greater than S we determine successive least-squares splines on the sphere until again (4.10) is satisfied. We start this iteration process with the spline $\hat{S}_{1,3}(\theta, \phi)$ according to the knots $\lambda_1 = \pi/2$, $\mu_1 = \pi/2$, $\mu_2 = \pi$ and $\mu_3 = 2\pi/3$. The corresponding spline interpolants $C(\phi)$ and $S(\phi)$ will then approximate $\cos(\phi)$ and $\sin(\phi)$ yet reasonably well. As opposed to the policy for the general smoothing spline, we also add two knots instead of one if the ϕ-direction is chosen, such that

$$h = 2n - 1, \ n = 2, 3, 4, \ldots \tag{4.15}$$

$$\mu_{j+n} = \mu_j + \pi \qquad j = 0, 1, \ldots, n. \tag{4.16}$$

The meaning of this is the following. From (2.7), (4.16) and the properties of divided differences, we find that

$$N_{j+n,l+1}(\phi + \pi) \equiv N_{j,l+1}(\phi), \qquad j = -l, \ldots, n-1. \tag{4.17}$$

Consequently, the B-spline coefficients of the spline interpolants $C(\phi)$ and $S(\phi)$ will also be such that

$$\begin{aligned} d_{j+n} &= -d_j \\ e_{j+n} &= -e_j \end{aligned} \qquad j=-l, \ldots, n-1 \tag{4.18}$$

considering the properties $\cos(\phi+\pi)=-\cos(\phi)$ and $\sin(\phi+\pi)=-\sin\phi$. So, from (2.32), (2.33), (4.17) and (4.18) we can easily derive that

$$\frac{\partial s(0,\phi+\pi)}{\partial\theta} \equiv -\frac{\partial s(0,\phi)}{\partial\theta} \qquad 0 \le \phi \le \pi \tag{4.19}$$

and analogously

$$\frac{\partial s(\pi,\phi+\pi)}{\partial\theta} \equiv -\frac{\partial s(\pi,\phi)}{\partial\theta} \qquad 0 \le \phi \le \pi. \tag{4.20}$$

Therefore, we may conclude that although the surface (1.1) with $\hat{s}_p$ instead of r, has only approximately C^1 continuity at the poles, at least this property is guaranteed for the intersection of the surface with any vertical plane $\phi=$constant.

5. A Fortran Subroutine Package for Smoothing Data on the Sphere with Tensor Product Splines

The algorithms described in sections 3.2 and 4.2 have been implemented in a Fortran subroutine package. This package consists of eleven subroutines, i.e. SMOSPH, ORDER, BSPLIN, BACK, COSSIN, ROTATE, RANDEF, DISCO, RATION, STAREP and BISP. Two routines, SMOSPH and BISP are the master subroutines of the package and each interfaces with the user. The remaining routines are supporting routines called by SMOSPH (ORDER, BSPLIN, BACK, COSSIN, ROTATE, RANDEF, DISCO, RATION, STAREP) and BISP (BSPLIN).

All programs are written in Standard Fortran and have been checked with the PFORT verifier of Bell Labs. The package has been succesfully implemented on an IBM 3033. A magnetic tape copy of the package, together with an example program can be obtained from the author.

5.1. Subroutine SMOSPH (TETA, PHI, R, W, M, IOPT, S, NT, TT, NP, TP, SUP, C, FP, IER)

Given the function values $R(I)$ at some points $(\text{TETA}(I), \text{PHI}(I))$ scattered arbitrarily in $D=[0,\pi]\times[0,2\pi]$, together with a set of weights $W(I)$, $I=1,2,\ldots,M$, subroutine SMOSPH determines a bicubic ($k=l=3$) spline approximation $\hat{s}(\theta,\phi)$ on the sphere. The program provides different modes of computation. If IOPT<0 it calculates a weighted least-squares spline on the sphere. In that case the user must provide the position of the interior knots in each direction and their number $(\text{TT}(j+4)=\lambda_j,\ j=1,2,\ldots,g=\text{NT}-8;\ \ \text{TP}(j+4)=\mu_j,\ j=1,2,\ldots,h=\text{NP}-8)$. If IOPT$\ge 0$ these knots are determined automatically by the routine. In that case, the user must provide a nonnegative number S. SMOSPH then determines a smoothing spline such that

$$\mathrm{FP} = \sum_{I=1}^{M} W(I)\big(R(I) - \hat{s}(\mathrm{TETA}(I),\ \mathrm{PHI}(I))\big)^{**}2 \le S$$

where equality holds unless $\hat{s}(\theta, \phi)$ is the least-squares polynomial $\hat{P}_{3,0}(\theta, \phi)$ (4.13).

Recommended values for S depend on the relative weights $W(I)$. If available, one should use an estimate δ_I of the standard deviation of the error in $R(I)$ and set $W(I) = (\delta_I)^{-2}$. If this value is used for $W(I)$, then a good S-value should be found in the range $M \pm \sqrt{2M}$ [13]. If nothing is known about the statistical error in $R(I)$, each $W(I)$ can be set equal to one and S determined by trial and error. To decide whether an approximation corresponding to a certain S is satisfactory or not, the user should then examine the fit graphically, e.g. by looking at its contour map. If S is too small, the spline approximation is too wiggly and picks up too much noise (overfit); if S is too large the spline is too smooth and signal will be lost (underfit). To economize the search for a good S-value, the user can proceed as follows. At the first call of the routine or whenever he wants to restart with the initial set of knots (see section 4.2), he must set IOPT $=0$. If he sets IOPT >0 the program will continue with the set of knots found at the last call of the routine. So, if he calls SMOSPH repeatedly with decreasing S-values, he can save a lot of computation time by specifying IOPT >0 from the second call on.

The fit $\hat{s}(\theta, \phi)$ is given in the standard B-spline representation (2.5) (B-spline coefficients $C(I)$, $I = 1, 2, \ldots, (\mathrm{NT}-4)(\mathrm{NP}-4)$) and can be evaluated by means of function program BISP. SMOSPH also returns the corresponding sum of squared residuals FP, the upper bound SUP for the smoothing factor S (if IOPT $=0$) and an error flag IER which is non-positive for a normal return and otherwise gives information about what went wrong.

For further explanation concerning the parameters of the routine, see the computer listing of the initial comments in the Appendix.

5.2. *Function BISP (TT, NT, TP, NP, KT, KP, C, TETA, PHI)*

This program can be used to evaluate the fit produced by SMOSPH, at an arbitrary point with coordinates (TETA, PHI) if we set KT $=$ KP $=3$ ($k = l = 3$).

6. Numerical Results

Example 1:

Using a random number generator, we generated a set of 192 data points (θ_q, ϕ_q), scattered uniformly over the approximation domain $D = [0, \pi] \times [0, 2\pi]$ and then we considered the data

$$(\theta_q, \phi_q),\ r_q = f(\theta_q, \phi_q),\ w_q = 1,\ q = 1, 2, \ldots, 192$$

in order to find an approximation for

$$f(\theta,\phi)=\sum_{i=1}^{3}\left[\left(\frac{\sin\theta\cos\phi}{A_i}\right)^2+\left(\frac{\sin\theta\sin\phi}{A_{i+1}}\right)^2+\left(\frac{\cos\theta}{A_{i+2}}\right)^2\right]^{-1/2}$$

$$0\leq\theta\leq\pi$$

$$0\leq\phi\leq 2\pi$$

with

$$\{A_i\,|\,i=1,\ldots,5\}=\{5,1,2,5,1\}.$$

Fig. 2 shows a contour map of this function and the position of the different data points.

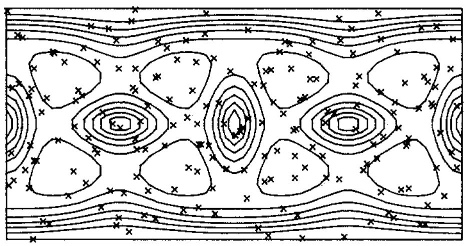

Fig. 2. $f(\theta,\phi)=\sum_{i=1}^{3}\left[\left(\frac{\sin\theta\cos\phi}{A_i}\right)^2+\left(\frac{\sin\theta\sin\phi}{A_{i+1}}\right)^2+\left(\frac{\cos\theta}{A_{i+2}}\right)^2\right]^{-1/2}$ with $\{A_i\,|\,i=1,\ldots,5\}=\{5,1,2,5,1\}$: Contour map and position of the data points

First a number of least-squares spherical harmonics (1.4) were determined. The results are summarized in Table 1. For each value of N, we give the dimension of the vector space (= the number of coefficients in the representation), the resulting sum of squared residuals SQ, the time T_1 in sec., needed on a IBM 3033 to determine the approximation (a general least-squares algorithm with Givens rotations without square roots [8] was used) and the time T_2 for evaluation (using a Fortran translation and somewhat optimized version of the program LEGENDREA [10]) this approximation on a 26×51 grid. The contour maps of Fig. 3 correspond to the spherical harmonics marked with * in Table 1. The spherical harmonic of Fig. 3.d. is the best least-squares approximation we could obtain for f, starting from our limited number of data. If we still increase the value of N, the residuals at the data points further decrease (SQ$=0.16$ for $N=11$) but, due to the addition of new basis functions into the representation (1.4) which become more and more oscillating as N gets larger, the global quality of approximation gets worser, especially in those regions where there is a lack of data.

Then we determined a number of bicubic least-squares tensor splines on the sphere with increasing numbers of knots. No attempt was made to locate the knots in an optimal way. They were simply chosen equidistantly, i.e. $\lambda_i=\pi i/(g+1)$; $\mu_j=2\pi j/(h+1)$ with $h=2g+1$. The results are summarized in Table 2 and Fig. 4 (the position of the interior knots of the spline approximations is marked by dot and dash lines). Comparing Tables 1–2 and Figs. 3–4, in particular the results

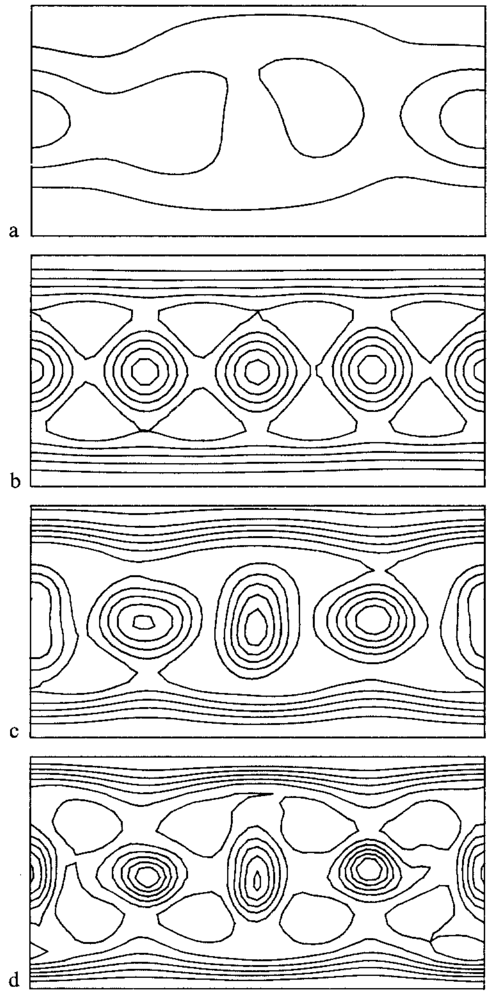

Fig. 3. Approximations with spherical harmonics; *a* $N=3$; *b* $N=4$; *c* $N=7$; *d* $N=10$

Table 1. *Least squares spherical harmonics*

N	dimension $(N+1)^2$	SQ	T_1	T_2
1	4	160.87	0.05	0.21
2	9	140.87	0.15	0.44
*3	16	136.75	0.35	0.76
*4	25	17.28	0.71	1.15
5	36	17.02	1.33	1.64
6	49	9.34	2.24	2.23
*7	64	8.78	3.56	2.93
8	81	1.26	5.37	3.70
9	100	0.98	7.74	4.63
*10	121	0.28	10.71	5.71

corresponding to approximations with about the same number of coefficients, we may conclude that the quality of fit obtained with both methods is nearly the same. However, as concerned computation times, the spline method is by far to be prefered. Especially we see that the evaluation time T_2 remains nearly constant with the increase of the number of knots.

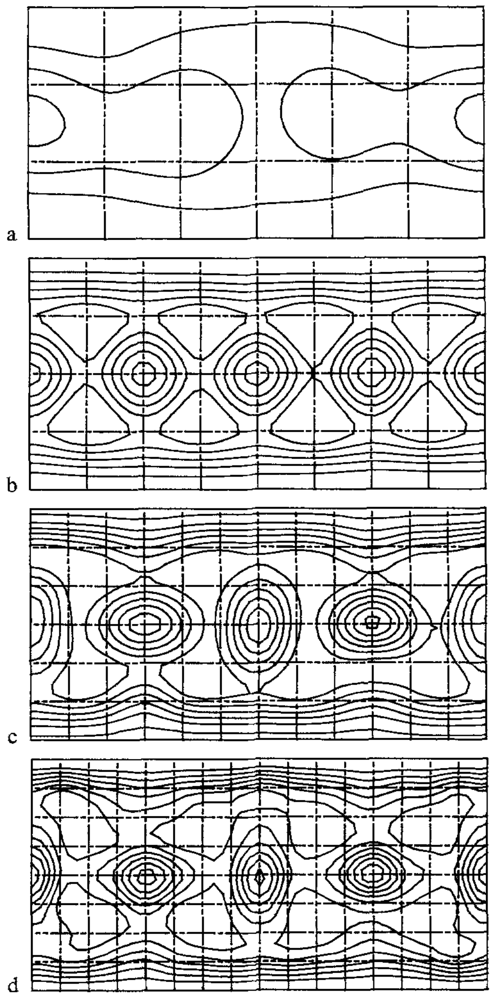

Fig. 4. Least-squares spline approximations on the sphere with equidistant knots λ_i, $i=1,2,\ldots,g$; $\mu_j, j=1,2,\ldots,h$; *a* $g=2, h=5$; *b* $g=3, h=7$; *c* $g=5, h=11$; *d* $g=7, h=15$

In Table 3 and Fig. 5 we give some results obtained with the program E02DAF of the NAG library [11] which computes a general least-squares bicubic spline. Again the knots were chosen equidistantly. The approximations are quite less accurate now. It clearly demonstrates the benefit of imposing the boundary conditions for the splines on the sphere. Also, E02DAF had to deal with rank deficiency very soon ($g > 3$). The dimension results in brackets in Table 3 give the rank of system (3.3) as determined by E02DAF.

Table 2. *Least-squares tensor splines on the sphere*

g	h	dimension $6+g(h+1)$	SQ	T_1	T_2
1	3	10	138.80	0.14	0.17
*2	5	18	132.97	0.34	0.17
*3	7	30	16.58	0.64	0.18
4	9	46	15.23	1.22	0.18
*5	11	66	5.00	1.68	0.18
6	13	90	4.81	2.19	0.18
*7	15	118	0.43	2.74	0.18

Table 3. *General least-squares splines*

g	h	dimension $(g+4)(h+4)$	SQ	T_1	T_2
0	1	20	109.02	0.20	0.17
*1	3	35	42.38	0.28	0.17
*2	5	54	40.38	0.35	0.17
*3	7	77	5.40	0.44	0.18
*4	9	104 (103)	8.19	0.59	0.18
5	11	135 (131)	583.66	0.78	0.18

Table 4. *Smoothing tensor splines on the sphere*

g	h	dimension $6+g(h+1)$	SQ	T_1	T_2
*2	3	14	135.	0.47	0.17
*5	7	46	15.	5.88	0.18
*7	7	62	5.	7.36	0.18
*9	11	114	0.5	40.83	0.18

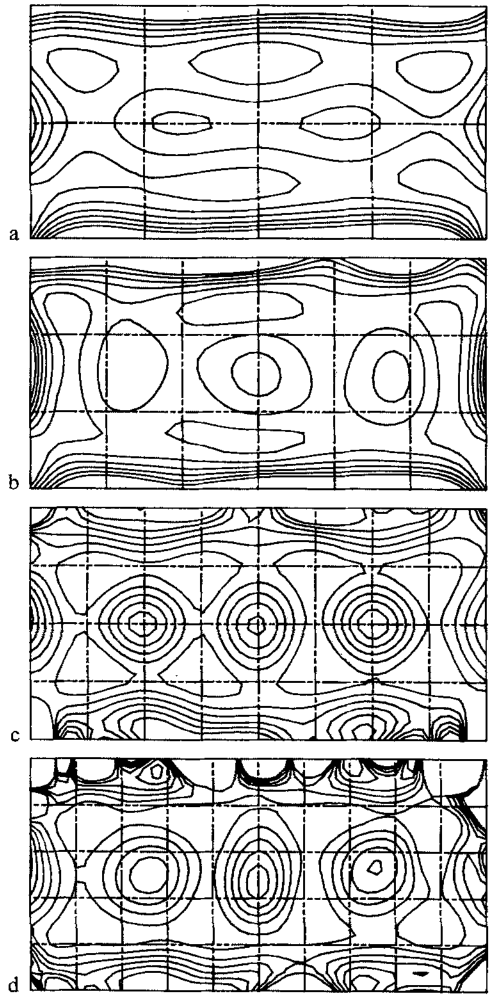

Fig. 5. General least-squares spline approximations with equidistant knots λ_i, $i=1,2,\ldots,g$; μ_j, $j=1,2,\ldots,h$; *a* $g=1, h=3$; *b* $g=2, h=5$; *c* $g=3, h=7$; *d* $g=4, h=9$

Finally, we determined a number of smoothing splines on the sphere. The results are given in Table 4 and Fig. 6. The smoothing factors were chosen as 135, 15, 5 and 0.5 in order to be able to compare with the least-squares splines of Fig. 4 (approximately the same sum of squared residuals SQ). We see that the smoothing splines are effectively smoother. Fig. 6 also demonstrates that our smoothing algorithm is adaptive in placing the knots; especially in Fig. 6.d. we see that there is a concentration of knots near the peaks. From Table 4 we notice that the computation time T_1 is less favorable for the smoothing splines. This has to do with the iterative nature of the algorithm (iterative determination of the knots, iterative calculation of the Lagrange parameter p). The evaluation time T_2 of course remains very small.

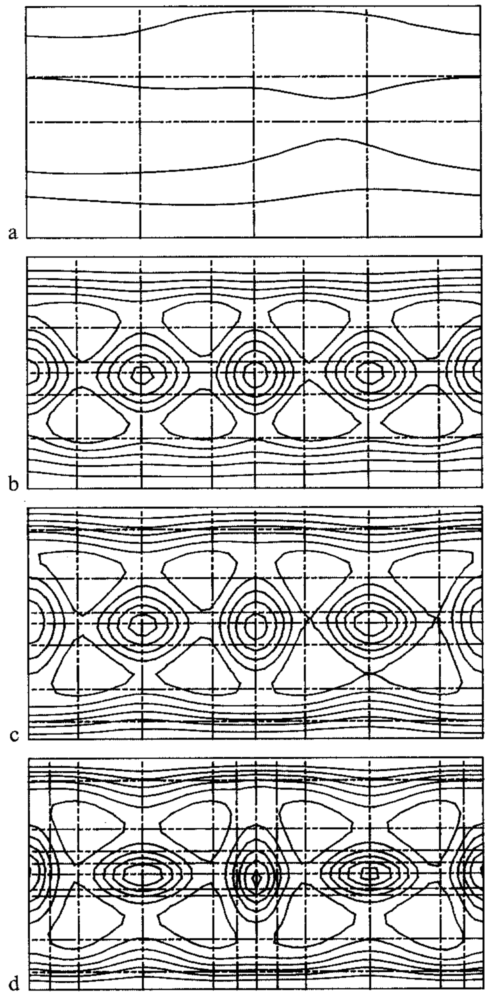

Fig. 6. Smoothing spline approximations on the sphere; *a* $S=135$; *b* $S=15$; *c* $S=5$; *d* $S=0.5$

Example 2:

In order to make stereotactic neurosurgery safer, possibilities were investigated for obtaining an integrated three-dimensional image of the cerebral blood vessels, the tumor and a simulated electrode trajectory [14]. Our algorithm was used to find a mathematical description of the surface of the tumor starting from a set of CT (computerized tomography)-slices.

Fig. 7 shows a three-dimensional depiction of such a spline approximation. In obtaining this parallel projection, another advantage of the use of tensor product splines was exploited, i.e. the fact that just like their evaluation, the calculation of partial derivatives can be performed in a quick and accurate way (as follows e.g. from (2.31)).

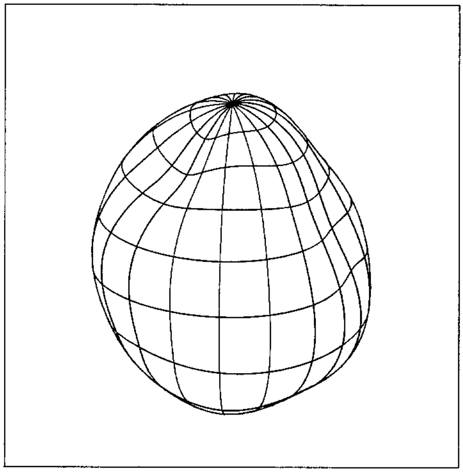

Fig. 7. Spline approximation of the surface of a tumor

Consequently we could also easily determine the normal

$$\vec{n}(\theta,\phi)=\left(\frac{\partial(y,z)}{\partial(\theta,\phi)},\ \frac{\partial(z,x)}{\partial(\theta,\phi)},\ \frac{\partial(x,y)}{\partial(\theta,\phi)}\right)$$

on the surface.

Then, by simply inspecting the sign of $\vec{n}(\theta,\phi)\cdot\vec{q}$ with $\vec{q}$ the projection direction, we decided whether the point on the surface was visible or not.

7. Appendix: Initial Comments of Subroutine SMOSPH

```
      SUBROUTINE SMOSPH(TETA,PHI,R,W,M,IOPT,S,NT,TT,NP,TP,SUP,C,FP,IER)
C  SUBROUTINE SMOSPH DETERMINES A SMOOTH BICUBIC SPHERICAL SPLINE
C  APPROXIMATION S(TETA,PHI), 0 <= TETA <= PI ; 0 <= PHI <= 2*PI
C  TO A GIVEN SET OF DATA POINTS (TETA(I),PHI(I),R(I)),I=1,2,...,M.
C  SUCH A SPLINE HAS THE FOLLOWING SPECIFIC PROPERTIES
C
C    (1) S(0,PHI)  = CONSTANT   0 <=PHI<= 2*PI.
C
C    (2) S(PI,PHI) = CONSTANT   0 <=PHI<= 2*PI
C
C          J              J
C        D S(TETA,0)    D S(TETA,2*PI)
C    (3) ----------- = ------------    0 <=TETA<=PI, J=0,1,2
C              J              J
C        D PHI          D PHI
C
C        D S(0,PHI)     D S(0,0)                  D S(0,PI/2)
C    (4) ----------  = -------- *COS(PHI) + ----------- *SIN(PHI)
C        D TETA         D TETA                    D TETA
C
C        D S(PI,PHI)    D S(PI,0)                 D S(PI,PI/2)
C    (5) ----------- = ---------*COS(PHI) + ------------*SIN(PHI)
C        D TETA         D TETA                    D TETA
C
C  IF IOPT < 0 SMOSPH CALCULATES A WEIGHTED LEAST-SQUARES SPHERICAL
C  SPLINE ACCORDING TO A GIVEN SET OF KNOTS IN TETA- AND PHI- DIRECTION.
C  IF IOPT >=0, THE NUMBER OF KNOTS IN EACH DIRECTION AND THEIR POSITION
C  TT(J),J=1,2,...,NT ; TP(J),J=1,2,...,NP ARE CHOSEN AUTOMATICALLY BY
C  THE ROUTINE. THE SMOOTHNESS OF S(TETA,PHI) IS THEN ACHIEVED BY MINI-
```

```
C  MALIZING THE DISCONTINUITY JUMPS OF THE DERIVATIVES OF THE SPLINE
C  AT THE KNOTS. THE AMOUNT OF SMOOTHNESS OF S(TETA,PHI) IS DETERMINED
C  BY THE CONDITION THAT FP = SUM(W(I)*(R(I)-S(TETA(I),PHI(I)))**2) BE
C  <= S, WITH S A GIVEN NON-NEGATIVE CONSTANT.
C  THE SPHERICAL SPLINE IS GIVEN IN THE STANDARD B-SPLINE REPRESENTATION
C  OF BICUBIC SPLINES AND CAN BE EVALUATED BY MEANS OF FUNCTION PROGRAM
C  BISP.
C
C  CALLING SEQUENCE:
C     CALL SMCSPH(TETA,PHI,R,W,M,IOPT,S,NT,TT,NP,TP,SUP,C,FP,IER)
C
C  INPUT PARAMETERS:
C    TETA  : ARRAY, MINIMUM LENGTH M, CONTAINING THE LATITUDE VALUES
C            TETA(I) OF THE DATA POINTS.
C    PHI   : ARRAY, MINIMUM LENGTH M, CONTAINING THE LONGITUDE VALUES
C            PHI(I) OF THE DATA POINTS.
C    R     : ARRAY, MINIMUM LENGTH M, CONTAINING THE DATA VALUES R(I).
C    W     : ARRAY, MINIMUM LENGTH M, CONTAINING THE WEIGHTS W(I).
C    M     : INTEGER VALUE, CONTAINING THE NUMBER OF DATA POINTS.
C    IOPT  : INTEGER FLAG WHICH SPECIFIES WHETHER A WEIGHTED LEAST-
C            SQUARES SPHERICAL SPLINE (IOPT<0) OR A SMOOTHING SPHERICAL
C            SPLINE (IOPT>=0) MUST BE DETERMINED. MOREOVER, IF IOPT>0
C            THE ROUTINE WILL CONTINUE WITH THE KNOTS FOUND AT THE LAST
C            CALL OF THE ROUTINE. ACCORDING TO THE VALUE OF IOPT THE
C            USER MUST PROVIDE A NUMBER OF ADDITIONAL PARAMETERS :
C      IOPT=0
C      ------
C      S   : REAL VALUE, CONTAINING THE SMOOTHING FACTOR.
C      IOPT<0
C      ------
C      NT  : INTEGER VALUE, GIVING THE TOTAL NUMBER OF KNOTS IN THE TETA
C            DIRECTION ; NT-8 GIVES THE NUMBER OF INTERIOR KNOTS.
C      TT  : ARRAY,MINIMUM LENGTH NTEST (SEE DATA INITIALIZATION), WHICH
C            CONTAINS THE POSITION OF THE INTERIOR KNOTS TT(I+4),I=1,2,.
C            ..NT-8 IN THE TETA-DIRECTION.
C      NP  : INTEGER VALUE, GIVING THE TOTAL NUMBER OF KNOTS IN THE PHI-
C            DIRECTION ; NP-8 GIVES THE NUMBER OF INTERIOR KNOTS.
C      TP  : ARRAY,MINIMUM LENGTH NPEST (SEE DATA INITIALIZATION), WHICH
C            CONTAINS THE POSITION OF THE INTERIOR KNOTS TP(I+4),I=1,2,.
C            ..NP-8 IN THE PHI-DIRECTION.
C      IOPT>0
C      ------
C      S   : REAL VALUE, CONTAINING THE SMOOTHING FACTOR.
C      NT,TT: THE NUMBER AND POSITION OF KNOTS OF THE SPHERICAL SPLINE
C      NP,TP  FOUND AT THE PREVIOUS CALL OF THE ROUTINE.
C      SUP : REAL VALUE, GIVING THE UPPER BOUND FOR THE SMOOTHING FACTOR
C            (SEE OUTPUT PARAMETER SUP)
C
C  OUTPUT PARAMETERS:
C    NT,TT : THE NUMBER AND POSITION OF KNOTS OF THE COMPUTED SPHERICAL
C    NP,TP   SPLINE ; BESIDE THE POSITION OF THE INTERIOR KNOTS IN TETA-
C            AND PHI-DIRECTION, THE ARRAYS TT AND TP WILL ALSO CONTAIN
C            THE BOUNDARY KNOTS WHICH ARE NEEDED FOR THE B-SPLINE RE-
C            PRESENTATION OF THE SPLINE.
C    SUP   : IF IOPT=0 THIS PARAMETER WILL CONTAIN THE UPPER BOUND FOR
C            THE SMOOTHING FACTOR.
C    C     : ARRAY, MINIMUM LENGTH (NPEST-4)*(NTEST-4),WHICH CONTAINS
C            THE B-SPLINE COEFFICIENTS IN THE STANDARD REPRESENTATION
C            OF THE SPHERICAL SPLINE.
C    FP    : REAL VALUE, GIVING THE WEIGHTED SUM OF SQUARED RESIDUALS
C            ACCORDING TO THE COMPUTED SPHERICAL SPLINE.
C    IER   : ERROR MESSAGE
C     IER = 0 : NORMAL RETURN.
C     IER =-1 : NORMAL RETURN; S(TETA,PHI) IS AN INTERPOLATING SPHERICAL
C               SPLINE (FP=0).
C     IER =-2 : NORMAL RETURN; S(TETA,PHI) IS THE LEAST-SQUARES POLYNOM-
C               IAL (FP=SUP<S).
C     IER > 0 : ABNORMAL TERMINATION.
C       IER= 1 : THE REQUIRED STORAGE SPACE EXCEEDS THE AVAILABLE SPACE,
C                SPECIFIED BY THE PARAMETERS NTEST AND NPEST ( SEE DATA
C                INITIALIZATION ).
C                PROBABLY CAUSES : S,NTEST OR NPEST TOO SMALL.
C       IER= 2 : A THEORETICALLY IMPOSSIBLE BEHAVIOUR OF THE LEAST-
C                SQUARES FUNCTION F(P) WAS FOUND DURING THE ITERATION
C                PROCESS.
C                PROBABLY CAUSES : TOL TOO SMALL.
C                                  BAD CHOICE OF EPS.
C       IER= 3 : THE MAXIMUM ALLOWABLE NUMBER OF ITERATIONS TO FIND THE
C                ROOT OF F(P)=S HAS BEEN REACHED.
C                PROBABLY CAUSES : MAXIT OR TOL TOO SMALL.
C       IER=10 : ONE OF THE FOLLOWING CONDITIONS WAS VIOLATED :
C                  NTEST >= 8  NPEST >= 8
```

```
C                W(I) > 0
C                0 <= TETA(I) <= PI , I=1,2,...,M
C                0 <= PHI(I) <= 2*PI
C       IER=20 : ONE OF THE FOLLOWING CONDITIONS WAS VIOLATED (IOPT<0)
C                8 <= NT <= NTEST  9 <= NP <= NPEST
C                0 < TT(I+4) < TT(I+5) < PI, I=1,2,...,NT-9
C                0 < TP(I+4) < TP(I+5) < 2*PI, I=1,2,...,NP-9
C       IER=30 : THE FOLLOWING CONDITION WAS VIOLATED (IOPT>=0)
C                S >= 0
C
C  OTHER SUBROUTINES REQUIRED :
C    ORDER,BSPLIN,BACK,COSSIN,ROTATE,RANDEF,DISCO,RATION AND STAREP.
C
      DIMENSION TETA(M),PHI(M),R(M),W(M),TT(17),TP(23),C(250),F(150),
     1 DIAG(150),A(150,70),INDEX(160),FPINT(26),COORD(26),NUMMER(200),
     2 HT(4),HP(4),H(70),DPRIME(150),FF(250),Q(150,70),BT(15,5),
     3 BP(15,5),SPT(200,4),SPP(200,4),ROW(23),COCO(23),COSI(23)
C  DATA INITIALIZATION STATEMENT TO SPECIFY
C    TOL  : THE REQUESTED RELATIVE ACCURACY FOR THE ROOT OF F(P)=S.
C    MAXIT: THE MAXIMUM ALLOWABLE NUMBER OF ITERATIONS TO FIND THE ROOT.
C    NTEST: OVER-ESTIMATES FOR THE NUMBERS NT AND NP. THESE PARAMETERS
C    NPEST: MUST BE SET BY THE USER TO INDICATE THE STORAGE SPACE
C           AVAILABLE TO THE ROUTINE. THE DIMENSION SPECIFICATIONS IN
C             SMCSPH:TT(NT),TP(NP),C,FF(NCOFF),F,DIAG,DPRIME(NCOF),
C                    A,Q(NCOF,BAND),H(BAND),INDEX(NREG),BT,BP(NMAX,5),
C                    FPINT,COORD(NRINT),ROW,COCO,COSI(NP),NUMMER(M),
C                    SPT,SPP(M,4)
C             BACK   :A(NCOF,...)
C             RANDEF:A(NCOF,...),AA(NCOF,BAND),FF,DD(NCOF),H(BAND)
C             DISCO :B(NMAX,...)
C           DEPEND (IMPLICITLY) ON M,NT AND NP, I.E.
C             NMAX=MAX(NT,NP)-8,NRINT=NT+NP-14,NCOFF=(NT-4)*(NP-4),
C             NREG=(NT-7)*(NP-7),NCOF=6+(NT-8)*(NP-7)
C             BAND=4*(NP-7)+3 IF NT>11
C                 =NCOF       IF NT<=11.
C           SINCE NT AND NP ARE UNKNOWN (IOPT>=0) AT THE TIME THE USER
C           SETS UP THE DIMENSION INFORMATION AN OVER-ESTIMATE OF THESE
C           ARRAYS MUST BE MADE. THE FOLLOWING REMARKS MAY HELP THE USER
C             (1) NT >= 8 , NP>=8
C             (2) THE SMALLER THE VALUE OF S, THE GREATER NT AND NP WILL
C                 BE.
C             (3) SINCE SMOSPH IS PRIMARILY INTENDED TO SMOOTH THE DATA
C                 (S NOT TOO SMALL), NT AND NP NORMALLY SHOULDN'T BE
C                 TAKEN TOO LARGE (E.G. NT=17,NP=23)
C    EPS  : THRESHOLD TO DETERMINE THE RANK OF A TRIANGULARIZED MATRIX;
C           THE RANK DEFICIENCY IS CONSIDERED TO BE THE NUMBER OF DIAG-
C           ONAL ELEMENTS DIAG(I) LESS THAN EPS*MAX(DIAG(I)).
      DATA TOL/0.001/,NTEST/17/,NPEST/23/,MAXIT/20/,EPS/0.1E-05/
C  BEFORE STARTING COMPUTATIONS A DATA CHECK IS MADE. IF THE INPUT DATA
C  ARE INVALID,CONTROLE IS IMMEDIATELY REPASSED TO THE DRIVER PROGRAM
```

Acknowledgement

This research is supported by the Belgian National Science Foundation (NFWO).

References

[1] Abramowitz, M., Stegun, I.: Handbook of Mathematical Functions. New York: Dover 1968.
[2] de Boor, C.: On calculating with *B*-splines. J. Approx. Theory *6*, 50–62 (1972).
[3] Boydstun, L. E., Armstrong, T. J., Bookstein, F. L.: A comparison of three dimensional maximum reach estimation techniques. J. Biomechanics *13*, 717–724 (1980).
[4] Cox, M. G.: The numerical evaluation of *B*-splines. J. Inst. Maths Applics *10*, 134–149 (1972).
[5] Cox, M. G.: The least-squares solution of overdetermined linear equations having band or augmented band structure. IMA J. Numer. Anal. *1*, 3–22 (1981).
[6] Dierckx, P.: An algorithm for surface fitting with spline functions. IMA J. Numer. Anal. *1*, 267–283 (1981).
[7] Dierckx, P.: Algorithms for smoothing data with periodic and parametric splines. Computer Graphics and Image Processing *20*, 171–184 (1982).
[8] Gentleman, W. M.: Least-squares computations by Givens transformations without square roots. J. Inst. Maths Applics *12*, 329–336 (1973).

[9] Hayes, J. G., Halliday, J.: The least-squares fitting of cubic spline surfaces to general data sets. J. Inst. Maths Applics *14,* 89–103 (1974).
[10] Herndon, J. R.: Algorithm 47: Associated Legendre Functions of the first kind for real or imaginary arguments. Comm. ACM *4,* 178 (1961).
[11] NAG Fortran Library Manual Mark 8, the Numerical Algorithms Group. Oxford: 1980.
[12] Peters, P., Wilkinson, J. H.: The least-squares problem and pseudo-inverses. Comput. J. *13,* 309–316 (1970).
[13] Reinsch, C.: Smoothing by spline functions. Numer. Math. *10,* 177–183 (1967).
[14] Suetens, P., Gijbels, J., Oosterlinck, A., Haegemans, A., Dierckx, P.: A three-dimensional image of the cerebral blood vessels and tumor for use in stereotactic neurosurgery. In: Proc. SPIE conference on Three-dimensional imaging, Genève, Vol. 375, 429–435 (1983).
[15] Wahba, G.: Spline interpolation and smoothing on the sphere. SIAM J. Sci. Stat. Comput. *2,* 5–16 (1981).

Dr. Ir. P. Dierckx
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
B-3030 Heverlee
Belgium