IMA Journal of Numerical Analysis (1981) **1**, 267–283

An Algorithm for Surface-Fitting with Spline Functions

P. DIERCKX

Applied Mathematics and Programming Division, Katholieke Universiteit Leuven, Celestijnenlaan 200 *A*, *B*-3030 *Heverlee, Belgium*

[Received 29 July 1980]

This paper presents an algorithm for fitting a bivariate spline function to a set of scattered data. The number of knots and their position are determined automatically. However, the user has to provide a non-negative constant to control the tradeoff between closeness of fit and smoothness of fit.

1. Introduction

THIS PAPER is concerned with the problem of fitting a two-dimensional spline function to a set of measured data. Such data usually have errors and so they have to be smoothed. One way to reduce the noise inherent in the raw data is to fit the approximating function by the method of least-squares. Hayes & Halliday (1974) for arbitrary data, and Dierckx (1977) for data on a rectangular grid, describe a computational method for fitting least-squares bicubic splines. However, automatic determination of knots is not considered. This problem is very important, because the goodness of fit with a least-squares spline function can be considerably affected by the number of knots and their position.

In this paper we will show how the one-dimensional smoothing criterion described by Dierckx (1975) and Dierckx & Piessens (1977) can be extended to two dimensions. This will allow us to automate the determination of knots. The basic idea of this approximation criterion is to define a measure for the smoothness of the approximating spline as well as for its closeness of fit, and then to find a way to control by a single parameter S, the extent to which these two (very often contradictory) properties will be satisfied.

2. Two-Dimensional Splines and Their B-spline Representation

In our approximation problem a closed rectangular domain $D = [a, b] \times [c, d]$ is concerned.

Consider the strictly increasing sequence of real numbers

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

and

$$c = \mu_0 < \mu_1 < \ldots < \mu_h < \mu_{h+1} = d, \tag{2.2}$$

0272–4979/81/030267 + 17 $02.00/0

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

(i) On any subrectangle

$$D_{i,j} = [\lambda_i, \lambda_{i+1}] \times [\mu_j, \mu_{j+1}], \quad i = 0, 1, \ldots, g, \quad j = 0, 1, \ldots, h$$

$s(x, y)$ is given by a polynomial of degree k in x and l in y.

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

Such a spline can be expressed (see for example Hayes & Halliday, 1974) as

$$s(x, y) = \sum_{q=-k}^{g} \sum_{r=-l}^{h} c_{q,r} M_{q,k+1}(x) N_{r,l+1}(y) \tag{2.3}$$

where $M_{q,k+1}(x)$ and $N_{r,l+1}(y)$ are normalized B-splines, defined on the knots

$$\lambda_q, \lambda_{q+1}, \ldots, \lambda_{q+k+1} \; (\lambda_{-k} = \ldots = \lambda_{-1} = a; \lambda_{g+2} = \ldots = \lambda_{g+k+1} = b),$$

respectively,

$$\mu_r, \mu_{r+1}, \ldots, \mu_{r+l+1} \; (\mu_{-l} = \ldots = \mu_{-1} = c; \mu_{h+2} = \ldots = \mu_{h+l+1} = d).$$

These B-splines enjoy the interesting property that

$$M_{q,k+1}(x) = 0 \quad \text{if } x \leqslant \lambda_q \quad \text{or } x \geqslant \lambda_{q+k+1} \tag{2.4}$$

$$N_{r,l+1}(y) = 0 \quad \text{if } y \leqslant \mu_r \quad \text{or } y \geqslant \mu_{r+l+1} \tag{2.5}$$

and they can be evaluated in a very stable way using the recurrence scheme of de Boor (1972) and Cox (1972).

3. The Approximation Criterion

Given the function values f_r at some points (x_r, y_r), $r = 1, 2, \ldots, m$ scattered arbitrarily in D, together with a set of positive weights w_r, $r = 1, 2, \ldots, m$, we will determine a spline approximation $s(x, y)$ of given degrees trying to find a compromise between the following objectives.

(i) The prescribed values f_r 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.

Now, in order to be able to formulate this criterion mathematically, first of all we need to find some measure of smoothness and some measure of closeness of fit. For the latter we can simply take the weighted sum of squared residuals

$$E = \sum_{r=1}^{m} w_r [f_r - s(x_r, y_r)]^2. \tag{3.1}$$

To find a suitable smoothing norm G, i.e. a measure of the lack of smoothness of $s(x, y)$, we consider the conditions for s to be a single polynomial on D, i.e.

$$\frac{\partial^{k+j}s(\lambda_i+0,y)}{\partial x^k\,\partial y^j}=\frac{\partial^{k+j}s(\lambda_i-0,y)}{\partial x^k\,\partial y^j},$$
$$i=1,2,\ldots,g;\quad j=0,1,\ldots,l;\quad \mu_0\leqslant y\leqslant\mu_{h+1}\tag{3.2}$$

$$\frac{\partial^{i+l}s(x,\mu_j+0)}{\partial x^i\,\partial y^l}=\frac{\partial^{i+l}s(x,\mu_j-0)}{\partial x^i\,\partial y^l},$$
$$j=1,2,\ldots,h;\quad i=0,1,\ldots,k;\quad \lambda_0\leqslant x\leqslant\lambda_{g+1}\tag{3.3}$$

Using the representation (2.3), (3.2) can be written as

$$\sum_{q=-k}^{g}a_{q,i}c_{q,r}=0,\qquad i=1,2,\ldots,g;\quad r=-l,\,-l+1,\ldots,h\tag{3.4}$$

with

$$a_{q,i}=M^{(k)}_{q,k+1}(\lambda_i+0)-M^{(k)}_{q,k+1}(\lambda_i-0),\text{ i.e. (Dierckx, 1975)}\tag{3.5}$$

$$\begin{aligned}a_{q,i}&=0,\quad \text{if } q<i-k-1,\quad \text{or } q>i\\&=\frac{(-1)^{k+1}k!\,(\lambda_{q+k+1}-\lambda_q)}{\pi'_{q,k}(\lambda_i)},\quad \text{if } i-k-1\leqslant q\leqslant i\end{aligned}\tag{3.6}$$

where

$$\pi_{q,k}(t)=(t-\lambda_q)(t-\lambda_{q+1})\ldots(t-\lambda_{q+k+1})\tag{3.7}$$

and the prime denotes derivative with respect to t.

Similarly (3.3) results in

$$\sum_{r=-l}^{h}b_{r,j}c_{q,r}=0,\qquad j=1,2,\ldots,h;\quad q=-k,\,-k+1,\ldots,g\tag{3.8}$$

with

$$b_{r,j}=N^{(l)}_{r,l+1}(\mu_j+0)-N^{(l)}_{r,l+1}(\mu_j-0),\tag{3.9}$$

i.e.

$$\begin{aligned}b_{r,j}&=0,\quad \text{if } r<j-l-1,\quad \text{or } r>j\\&=\frac{(-1)^{l+1}l!\,(\mu_{r+l+1}-\mu_r)}{\Pi'_{r,l}(\mu_j)},\quad \text{if } j-l-1\leqslant r\leqslant j\end{aligned}\tag{3.10}$$

where

$$\Pi_{r,l}(t)=(t-\mu_r)(t-\mu_{r+1})\ldots(t-\mu_{r+l+1}).\tag{3.11}$$

So altogether, we have found $g(h+l+1)+h(g+k+1)$ conditions which must be satisfied by the B-spline coefficients $\bar{c}$ for $s(x,y)$ to be a single polynomial on D.

One can easily prove, however, that $g\,.\,h$ of these equations are linearly dependent as might be expected according to the number of B-spline coefficients, i.e. $(g+k+1)(h+l+1)$, and the dimension of the polynomial, i.e. $(k+1)(l+1)$.

Now, we can define a smoothing norm G as follows

$$G(\bar{c})=\sum_{i=1}^{g}\sum_{r=-l}^{h}\left(\sum_{q=-k}^{g}a_{q,i}c_{q,r}\right)^2+\sum_{j=1}^{h}\sum_{q=-k}^{g}\left(\sum_{r=-l}^{h}b_{r,j}c_{q,r}\right)^2.\tag{3.12}$$

Then the approximation criterion can be formulated mathematically as follows

$$\text{Minimize } G(\bar{c}) \tag{3.13}$$

$$\text{Subject to the constraint } E(\bar{c}) \leqslant S \tag{3.14}$$

where S is a given, non-negative constant which will control the extent of smoothing and therefore is called the "smoothing factor".

4. The Solution of the Minimization Problem

4.1. *The Method of Lagrange*

Using the method of Lagrange we can solve (3.13)–(3.14) by searching the critical points of the function

$$K(p, z, \bar{c}) = G(\bar{c}) + p(E(\bar{c}) + z^2 - S). \tag{4.1}$$

Differentiating K with respect to p and z gives

$$E(\bar{c}) = S - z^2, \tag{4.2}$$

$$p \, . \, z = 0. \tag{4.3}$$

If $p = 0$, every polynomial of degree k in x and l in y, minimizes (4.1). So if the least-squares polynomial $P_{k,l}(x, y)$ satisfies the constraint

$$\sum_{r=1}^{m} w_r[f_r - P_{k,l}(x_r, y_r)]^2 \leqslant S$$

then it is a solution of our problem.

We are more interested in the non-trivial case $p \neq 0, z = 0$. Negative values of p do not have to be considered in this minimization problem (cf. Reinsch, 1967).

In Section 4.2 we will show that with every positive value of p there is a corresponding single spline $s_p(x, y)$ minimizing $K(p, 0, \bar{c})$. If we denote by $F(p)$ the function

$$F(p) = \sum_{r=1}^{m} w_r[f_r - s_p(x_r, y_r)]^2 \tag{4.4}$$

then, according to (4.2) with $z = 0$, the only thing left is to find the value of p such that $F(p) = S$. This problem will be handled in Section 4.3.

4.2. *The Calculation of the B-spline Coefficients* $\bar{c}$

In this section we consider the problem of minimizing $K(p, 0, \bar{c})$ for a fixed positive p. The determination of $s_p(x, y)$ is simply a matter of determining values for the B-spline coefficients $c_{q,r}$ as the least-squares solution of the following linear system of equations

$$\begin{aligned}
&\sqrt{w_i} \sum_{q=-k}^{g} \sum_{r=-l}^{h} M_{q,k+1}(x_i) N_{r,l+1}(y_i) c_{q,r} = \sqrt{w_i} f_i, \quad i = 1, 2, \ldots, m, \\
&\frac{1}{\sqrt{p}} \sum_{q=-k}^{g} a_{q,i} c_{q,r} = 0, \quad i = 1, 2, \ldots, g; \quad r = -l, \ldots, h, \\
&\frac{1}{\sqrt{p}} \sum_{r=-l}^{h} b_{r,j} c_{q,r} = 0, \quad j = 1, 2, \ldots, h; \quad q = -k, \ldots, g,
\end{aligned} \tag{4.5}$$

which may be written in matrix notation as

$$\mathbf{Qc} = \mathbf{e} \tag{4.6}$$

where $\mathbf{c}$ denotes the vector containing the coefficients $c_{q,r}$ in the order $r = -l, -l+1, \ldots, h$ for fixed q, in which $q = -k, -k+1, \ldots, g$ in turn. From (2.4), (2.5), (3.6) and (3.10) it follows that matrix $\mathbf{Q}$ contains many zeros.

Following Cox & Hayes (1973) we have used an orthogonalization method with Givens rotations without square roots (Gentleman, 1973) to determine the least-squares solution of (4.6). This method is very suitable for our problem because the equations are treated row by row and because zeros already present in $\mathbf{Q}$ are readily exploited to reduce computation time and memory requirements.

In the case $p = \infty$ only the first m equations of (4.5) are to be considered. If we arrange the data points (x_i, y_i) according to the panel $D_{q,r}$ containing them, in the order $r = 0, 1, \ldots, h$ for fixed q, in which $q = 0, 1, \ldots, g$ in turn, then matrix $\mathbf{Q}$ has a special bandstructure of which full advantage can be taken (see Hayes & Halliday, 1974). $\mathbf{Q}$ will be reduced to an upper triangular matrix $\mathbf{R}$ with bandwidth

$$\text{IBAND} = k(h+l+1)+l+1, \quad \text{if } p = \infty, \quad \text{or } g = 0. \tag{4.7.1}$$

Eventually the independent variables x and y (and the corresponding parameters) will be switched if this can reduce IBAND, i.e. if $l \,.\, g < k \,.\, h$. The case $p \neq \infty$ is treated in the following manner. We simply add to $\mathbf{R}$ the rows according to the second and third set of equations of (4.5) and then we "zero out" these extra rows by applying a new sequence of Givens transformations. The middle equations, when they exist ($g > 0$), will increase the bandwidth to become

$$\text{IBAND} = (k+1)(h+l+1)+1, \quad \text{if } p \neq \infty, \quad \text{and } g > 0. \tag{4.7.2}$$

If (4.5) has to be solved only for a single positive value of p, the proposed method is not the most efficient one. Indeed, to get the special bandstructure in $\mathbf{Q}$, the second and third set of equations must be interleaved with the first m equations. However, in our problem (see Section 4.3) (4.5) has to be solved for $p = \infty$ first and then eventually also for other values of $p \neq \infty$. So in that case we save a great deal of work since the reduction of the first m rows of $\mathbf{Q}$ (the determination of $\mathbf{R}$) is carried out only once.

Finally we mention that the matrix $\mathbf{Q}$ possibly is (numerically) rank deficient. In that case the minimal-length solution (see for example Peters & Wilkinson, 1970) is computed.

4.3. *Determination of the Lagrangian Parameter p*

In this section we consider the problem of finding a positive root of $F(p) = S$. From (4.4) and (4.5) we know that

$$F_{g,h}(\infty) = \sum_{r=1}^{m} w_r[f_r - S_{g,h}(x_r, y_r)]^2 \tag{4.8}$$

where $S_{g,h}(x, y)$ stands for the least-squares spline with fixed knots λ_i, $i = 1, 2, \ldots, g$; μ_j, $j = 1, 2, \ldots, h$.

If p tends to zero, the second and third set of equations in (4.5) have much more weight than the first set, so that, if possible they will have to be satisfied exactly. Therefore we may conclude [see (3.4) and (3.8)] that if p tends to zero, $s_p(x, y)$ will tend to the least-squares polynomial $P_{k,l}(x, y)$. Therefore, let

$$F(0) = \sum_{r=1}^{m} w_r[f_r - P_{k,l}(x_r, y_r)]^2. \tag{4.9}$$

If Q has full rank for $p = \infty$, we can prove similarly as in Dierckx (1975), that F is a convex and strictly decreasing function of p. Therefore we know that, once we have found a set of knots such that

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

there exists a unique positive root $\bar{p}$ of $F(p) = S$. This $\bar{p}$ can be found by means of a Newton–Raphson method (Dierckx, 1975) or in a more rapid way (Dierckx & Piessens, 1977) by means of an iterative scheme based on rational interpolation (Jarrat & Nudds, 1965).

Three values of p, starting with

$$p_1 = 0, \qquad p_2 = -\frac{F(0)-S}{F_{g,h}(\infty)-S}, \qquad p_3 = \infty, \tag{4.11}$$

and the corresponding values of $F(p)$ (F_1, F_2, F_3) are used to calculate the parameters u, v and w of an interpolating function of the form

$$R(p) = \frac{up+v}{p+w}. \tag{4.12}$$

Putting $R(p) = S$ gives a new approximation p^* for $\bar{p}$, i.e.

$$\begin{aligned} p^* &= \frac{p_1(F_1-F_3)(F_2-S)-p_2(F_2-F_3)(F_1-S)}{(F_1-F_2)(F_3-S)}, \text{ if } p_3 = \infty \\ &= -\frac{(F_3-S)(F_1-F_2)p_1p_2+(F_2-S)(F_3-F_1)p_3p_1+(F_1-S)(F_2-F_3)p_2p_3}{(F_3-S)(F_1-F_2)p_3+(F_2-S)(F_3-F_1)p_2+(F_1-S)(F_2-F_3)p_1}, \\ &\qquad \text{if } p_3 \neq \infty. \end{aligned} \tag{4.13}$$

After each iteration the values of p_1, p_2 and p_3 are adjusted as follows.

If
$$\begin{aligned} F_2 < S&: p_3 \leftarrow p_2, \quad p_2 \leftarrow p^* \\ F_2 > S&: p_1 \leftarrow p_2, \quad p_2 \leftarrow p^*. \end{aligned} \tag{4.14}$$

So the interval $[p_1, p_3]$ is permanently reduced, while

$$p_1 < p_2 < p_3, \qquad F_1 > S \quad \text{and} \quad F_3 < S, \tag{4.15}$$

which makes our iteration scheme strictly convergent. The iteration process is stopped as soon as

$$\frac{|F(p^*)-S|}{S} \leqslant \varepsilon \tag{4.16}$$

with ε a given relative tolerance. (Since two approximations, corresponding to slightly different S-values, are hardly distinguishable on a graphical display, ε should not be taken too small, for example $\varepsilon = 0{\cdot}001$.) Finally we mention that also in the rank-deficient case, this procedure has been found to work satisfactorily in practice.

5. Our Choice of Knots

In this section we will say something about the automatic determination of knots. Let us first state that we have not really tried to find the minimum number of knots nor their optimal positions.

The chosen set must satisfy condition (4.10). It is not so difficult to prove that there exists at least one such set. Clearly for any set of data, it is possible to choose a set of knots for which each panel $D_{i,j}$ contains at most one data point. Then it is easy to check that the least-squares solution (of minimal length) corresponding to such a set of knots must be an interpolating spline, i.e. $F_{g,h}(\infty) = 0$. (In the assumption there are no data points with coincident x and y values but differing f values.) Of course, we are interested in an approximation with fewer knots. Therefore we proceed in the following manner. First of all, we determine the least-squares polynomial $P_{k,l}(x, y)$ which simply is the least-squares spline $S_{0,0}(x, y)$. If $F_{0,0}(\infty) \leqslant S$ 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 $S_{g,h}(x, y)$ with an increasing number of knots, until we find that condition (4.10) is satisfied. At each iteration we compute the quantities

$$R_r = w_r[f_r - S_{g,h}(x_r, y_r)]^2, \quad r = 1, 2, \ldots, m, \tag{5.1}$$

$$\eta_i = \sum_{\lambda_i \leqslant x_r < \lambda_{i+1}} R_r, \quad i = 0, 1, \ldots, g, \tag{5.2}$$

$$\theta_j = \sum_{\mu_j \leqslant y_r < \mu_{j+1}} R_r, \quad j = 0, 1, \ldots, h, \tag{5.3}$$

$$\eta_u = \max \eta_i, \quad i = 0, 1, \ldots, g, \tag{5.4}$$

$$\theta_v = \max \theta_j, \quad j = 0, 1, \ldots, h, \tag{5.5}$$

and add a knot in the x-direction if $\eta_u \geqslant \theta_v$ and in the y-direction otherwise. More precisely, this additional knot is located at the point

$$X = \sum_{\lambda_u \leqslant x_r < \lambda_{u+1}} R_r x_r / \eta_u \tag{5.6}$$

if the x-direction is chosen and at the point

$$Y = \sum_{\mu_v \leqslant y_r < \mu_{v+1}} R_r y_r / \theta_v, \tag{5.7}$$

if the y-direction is chosen, i.e. at some weighted mean value of the x-values, respectively, y-values of the data in the region considered. (Note, in relation to Equations (5.2), (5.3), (5.6) and (5.7), that when an x_r (or y_r) value coincides with λ_{g+1} (or μ_{h+1}) it is treated as contained in the appropriate end interval.)

The object of this strategy is simply to add a knot at that part of the approximation domain where the fit $\breve{S}_{g,h}(x, y)$ is particularly poor. The choice of the direction can be justified in some sense by the consideration that if the function underlying the data has a pronounced difficulty in one direction (e.g. a strong peak at $x = a$) and a smooth behaviour in the other one, the total sum of squared residuals

$$\sum_{i=0}^{g} \eta_i = \sum_{j=0}^{h} \theta_j$$

will be spread unequally over the different intervals in the first direction (big and small η_i-values) and much more uniformly over the intervals in the second, most probably resulting in an addition in the first direction. The use of (5.6) and (5.7) will tend to place the knot where the residuals are largest. It will also guarantee that, if for example the x-direction is chosen, each of the regions $[\lambda_u, X] \times [c, d]$ and $[X, \lambda_{u+1}] \times [c, d]$ will contain data points. (This cannot be guaranteed for example if we locate the knot simply midway in the interval $[\lambda_u, \lambda_{u+1}]$.) As opposed to the method in one dimension (Dierckx, 1975), we only add one knot each time because here the dimension of the spline grows much faster. Indeed, if we add a single knot in the x-direction, the number of B-spline coefficients increases by $h+l+1$ ($g+k+1$ if we add a knot in the y-direction).

The consequences are

(i) on the one hand, that the sum of squared residuals possibly decreases rapidly, but

(ii) on the other hand, that the time for solving (4.6) may go up very fast, not only because of the increased dimension of Q but also (and primarily) because of a possible increase in the bandwidth (4.7). [In one dimension the bandwidth only depends on the degree of the spline and not on the number of knots (Dierckx, 1975).]

It also seems logical, therefore, to annihilate a knot addition before performing a new one (and to prevent herewith and also later on, another addition in that region) if it appears to give an insufficient reduction in the sum of squared residuals, for example if

$$F_{g,h}(\infty) - F_{g+1,h}(\infty) \leqslant \varepsilon S \quad \text{or} \quad F_{g,h}(\infty) - F_{g,h+1}(\infty) \leqslant \varepsilon S \tag{5.8}$$

according to the direction in which the knot was located, i.e. if the reduction is less than the tolerance on S (4.16).

6. Some Practical Considerations

The algorithm described in the previous sections has been implemented in a Fortran subroutine package, called SURFAC (Dierckx, 1980). A copy of this package, together with an example program can be obtained from the author, on magnetic tape.

Apart from the set of data points (x_r, y_r, f_r) with the corresponding weights w_r and the degrees (k, l) of the spline, the user merely has to provide the smoothing factor S, to control the tradeoff between the "roughness" of the fit, as measured by the smoothing norm G (3.12), and the infidelity to the data, as measured by the weighted sum of squared residuals E (3.1). Recommended values of S depend on the relative weights w_r. Reinsch (1967) suggests choosing S in the range $m \pm \sqrt{2m}$ if the weights are taken as $(\delta f_r)^{-2}$ with δf_r an estimate of the standard deviation of f_r. If nothing is known about the statistical error in f_r, each w_r can be set equal to one and S determined by trial and error. In this connection it is useful to note that 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 will be too smooth and signal will be lost (underfit). In the extreme cases the algorithm will return the least-squares polynomial $P_{k,l}(x, y)$ if S is very large and an interpolating spline if $S = 0$. To decide whether an approximation, corresponding to a certain S, is satisfactory, the user should examine it graphically, e.g. by looking at its contour map. Finally it should be mentioned that the value of S is normally not very crucial and also depends on the application; e.g. if the approximation is to be used for numerical differentiation, it is better to choose S rather too large than too small.

7. Numerical Results

Example 1

To check the influence of the smoothing factor S, we construct the following example.

Using a random number generator we generate

(i) a set of 160 points (x_r, y_r), scattered uniformly in the region $[-2, 2] \times [-2, 2]$
(ii) a set of normally distributed stochastic variates e_r with expected value 0 and standard deviation 0·01.

Then we calculate

$$\bar{e} = \frac{1}{160} \sum_{r=1}^{160} e_r, \qquad \delta^2 = \frac{1}{159} \sum_{r=1}^{160} (e_r - \bar{e})^2$$

and consider the data

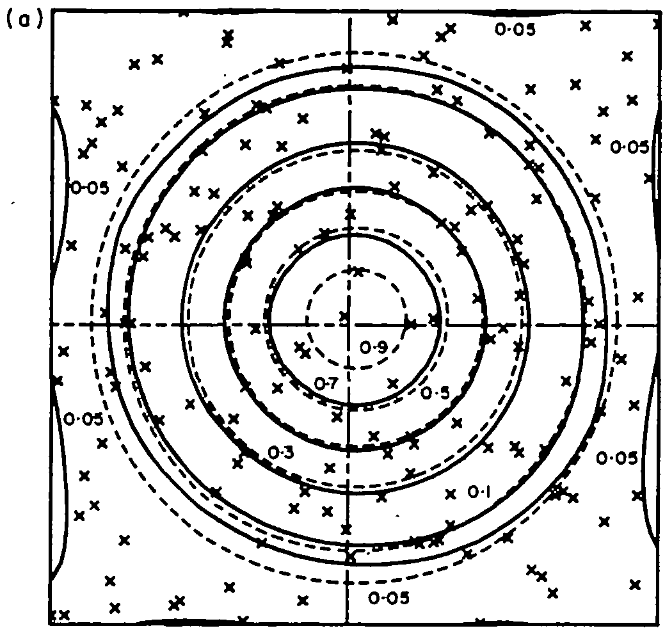
$$(x_r, y_r), \quad f_r = f(x_r, y_r) + e_r, \quad w_r = \delta^{-2}, \quad r = 1, 2, \ldots, 160, \tag{7.1}$$

in order to find a bicubic ($k = l = 3$) spline approximation for

$$f(x, y) = \exp(-x^2 - y^2), \quad -2 \leqslant x, y \leqslant 2,$$

by using subroutine SURFAC (Dierckx, 1980). Figure 1 shows the position of the different points (x_r, y_r), the contour map of the exact function $f(x, y)$ (dashed line) and the contour maps of some spline approximations according to different S-values. For each approximation the position of the interior knots is marked by dot and dash lines. Choosing S too high ($S = 1000$) results in oversmoothed data (Fig. 1a). Overfitting the data by choosing S too small ($S = 60$) results in a contour map which is highly influenced by the errors e_r (Fig. 1b). Finally Fig. 1c and Fig. 1d show the contour map of approximations ($S = 160$; $S = 120$) which could be accepted as satisfactory.

This example also demonstrates that the fit produced by SURFAC may become



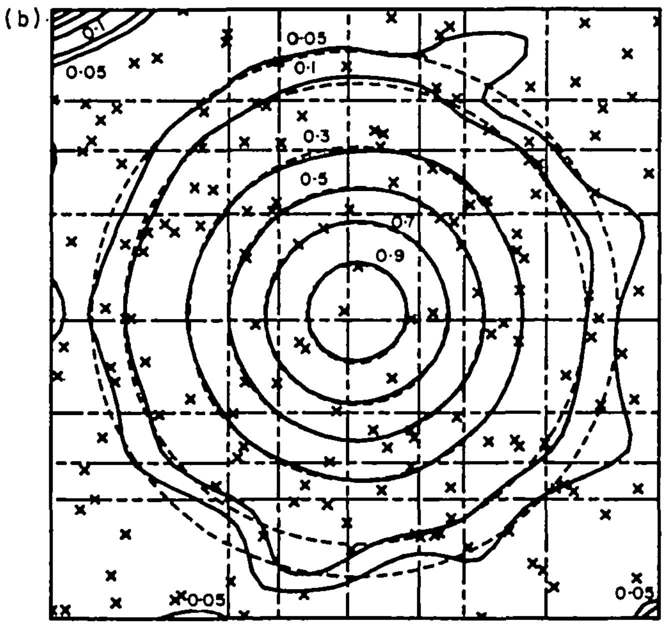
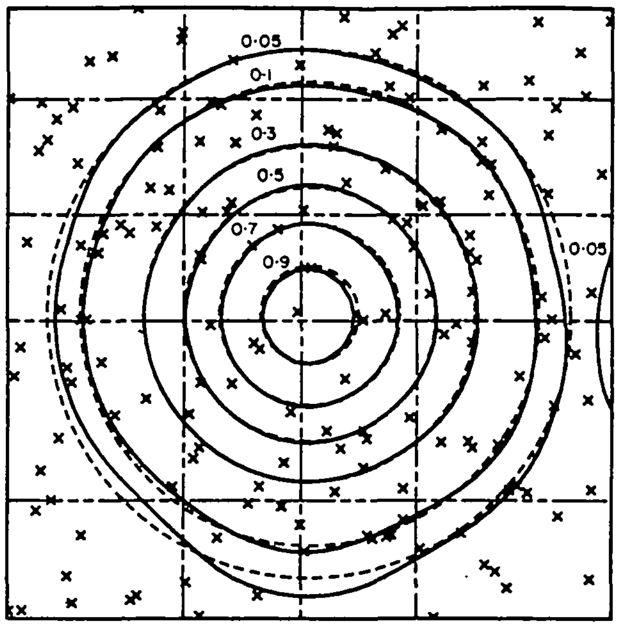


FIG. 1. A bicubic spline approximation for $f(x, y) = \exp(-x^2-y^2)$, $-2 \leqslant x, y \leqslant 2$. (a) Underfitting: $S = 1000$; (b) overfitting: $S = 60$.

(c)



(d)

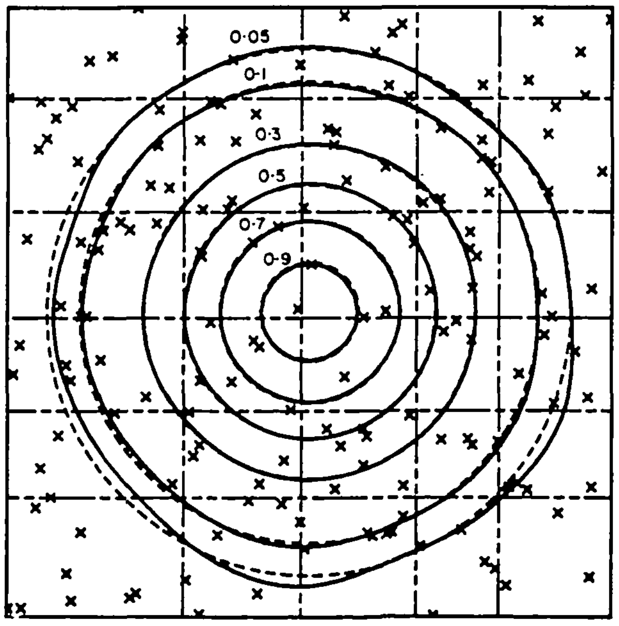


FIG. 1 (contd). (c) satisfactory fit: $S = 160$; (d) satisfactory fit: $S = 120$.

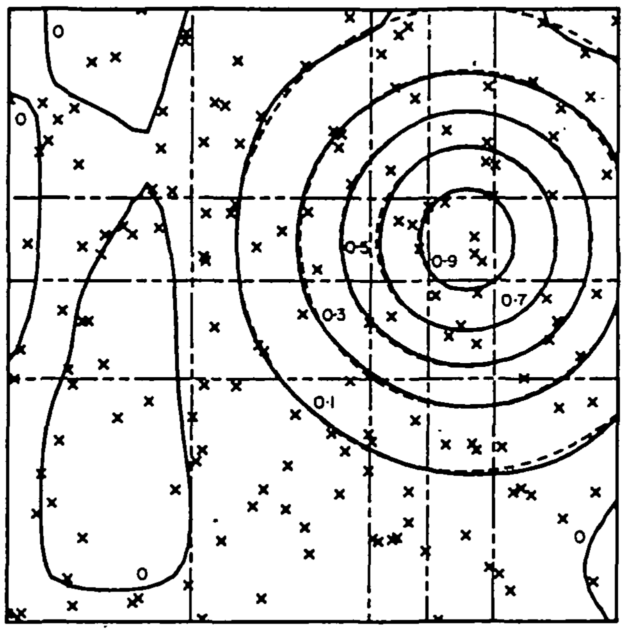


FIG. 2. A bicubic spline approximation for $f(x,y) = \exp[-(x-1)^2-(y-1/2)^2]$, $-2 \leqslant x, y \leqslant 2$; $S = 160$.

unreliable in regions, especially near the boundaries of the approximation domain, where there are no data points. This unfavourable situation is accentuated if S becomes smaller.

Example 2

As in example 1 we consider the data (7.1) but now with respect to the function

$$f(x,y) = \exp[-(x-1)^2-(y-1/2)^2].$$

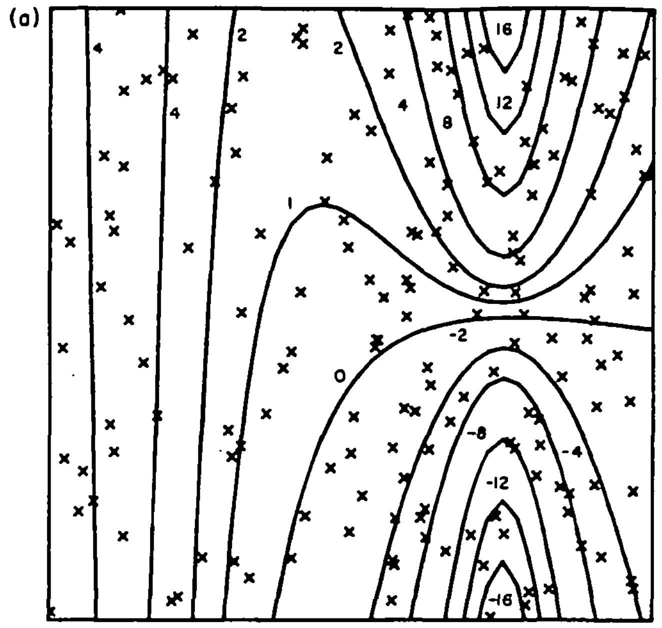
Figure 2 shows the approximation result according to the smoothing factor $S = 160$. This example demonstrates that the algorithm is adaptive in placing the knots.

Example 3

In a third test example we consider the function

$$f(x,y) = y/[(x-1)^2+0{\cdot}1]+1/[(x+1{\cdot}5)^2+0{\cdot}2], \quad -2 \leqslant x, y \leqslant 2,$$

which is far from symmetric between x and y. Figure 3a shows a contour map of this function and also the position of the data points (x_r, y_r). As one can see, the data are not uniformly scattered over the approximation domain. In fact they are twice as dense in $[0,2]\times[-2,2]$ than in $[-2,0]\times[-2,2]$. Altogether we have $m = 168$ points. The function values f_r are generated as in example 1 (stochastic errors with expected value 0 and standard deviation 0·05). The weights w_r are also likewise determined.



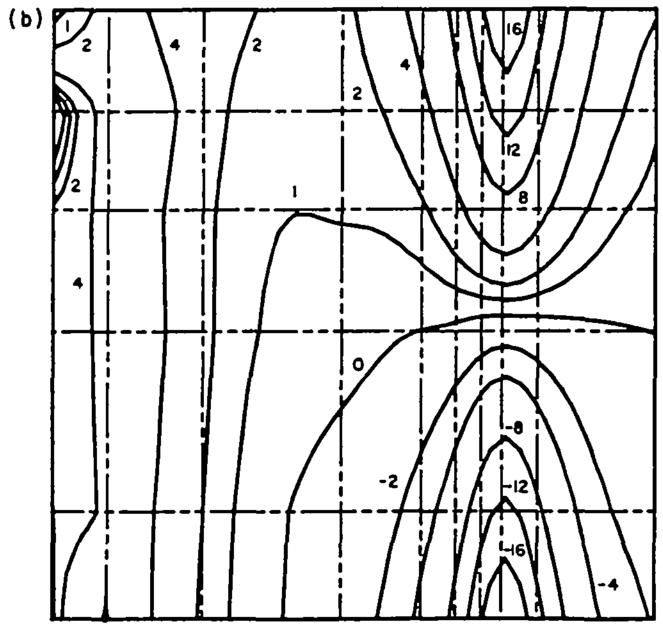


FIG. 3. A spline approximation of degrees $(k, l) = (3, 1)$ for
$f(x, y) = y/((x-1)^2+0{\cdot}1)+1/((x+1{\cdot}5)^2+0{\cdot}2), \quad -2 \leqslant x, y \leqslant 2.$
(a) Contour map of $f(x, y)$; position of the data points; (b) a spline approximation with $S = 200$.

(c)

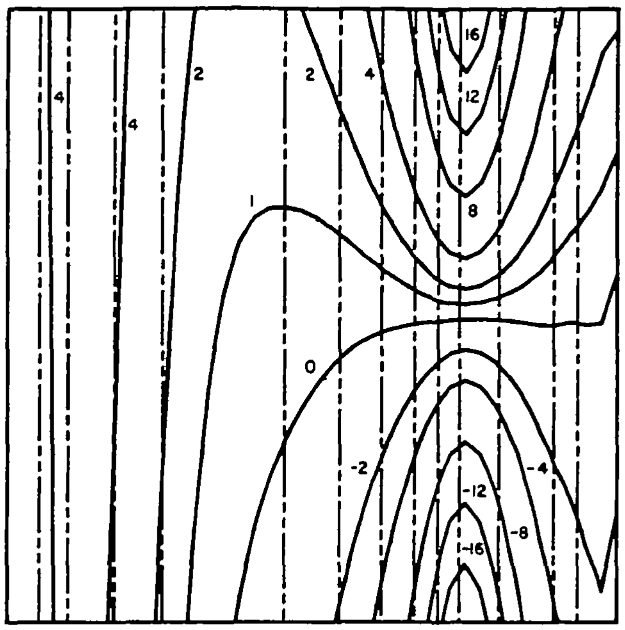


FIG. 3. (contd). (c) A spline approximation with $S = 200$ but linear in y.

Figure 3b shows the contour map of the spline approximation of degree $k = 3$, $l = 1$, according to $S = 200$. We see that the approximation is less accurate in $[-2, 0] \times [-2, 2]$ than in $[0, 2] \times [-2, 2]$ although $f(x, y)$ is smoother there. This is quite normal, however, since there are less data points in that region. Concerning the knots (see the dot and dashed lines) we remark that there is a high concentration of them near the steepest peak. We also find that there are twice as many interior knots in the x-direction as in the y-direction; the latter are scattered much more uniformly over the approximation interval.

To indicate the storage space which is available to the computer routine SURFAC, the user must specify an over-estimate for the number of knots in each direction. But with this he can also limit the number of knots, the algorithm will locate in a direction. In this example we can use this facility to compute a new approximation, now without interior knots in the y-direction. Figure 3c shows the result according to the same smoothing factor $S = 200$. The approximation is now more accurate in $[-2, 0] \times [-2, 2]$. This only confirms the general rule that the more information we give (here the fact that, as $f(x, y)$, the spline must be linear in the variable y) the more accurate the results we get.

Example 4

In a fourth example we generate a set of data which are scattered incompletely over $[0, 1{\cdot}5] \times [0, 1{\cdot}5]$, i.e. within a circle of radius 1·5 with the origin as centre. The function we want to approximate is

$$f(x, y) = \sin(4x)\sin(4y)/(xy).$$

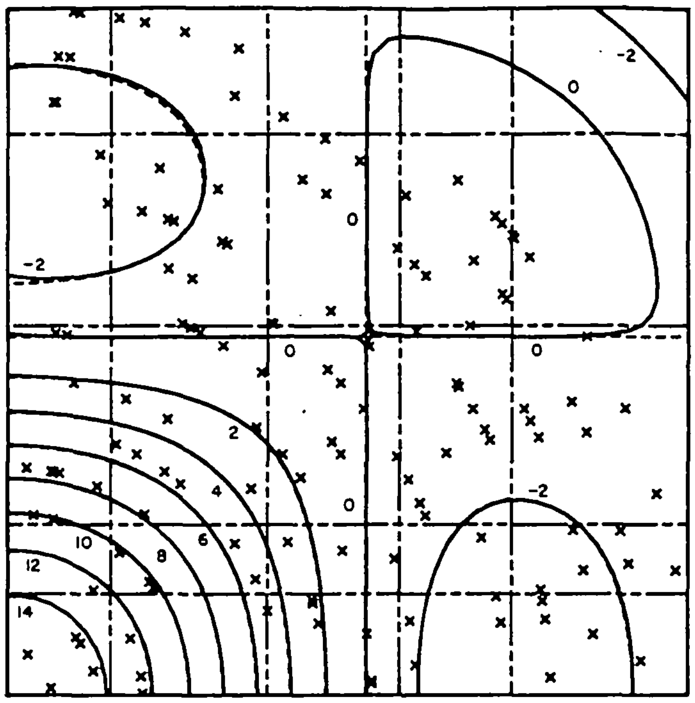


FIG. 4. A bicubic spline approximation for
$f(x, y) = \sin(4x)\sin(4y)/(xy)$, $0 \leqslant x, y \leqslant 1{\cdot}5$; $S = 0{\cdot}01$.

No errors are superposed on the function values. The weights are all taken as 1 except at the points within a distance 0·3 of the origin which get a weight of 10. Figure 4 shows the approximation result corresponding to a bicubic spline with $S = 0{\cdot}01$. In this example our algorithm had to deal with rank deficiency. Indeed, since there are no data points inside the two panels in the right upper corner, we know that matrix **Q** (4.6) has at least a rank deficiency 2 in case $p = \infty$.

Example 5

Finally, we consider a more practical problem, specifically the smoothing of some meteorological data. Figure 5 shows the position of 109 radiosonde-stations over Europe with the measured 700 mb geopotential height (in decametres). The contour map corresponds to a bicubic spline approximation with $S = 500$ (all weights $w_r = 1$).

8. Concluding Remarks

In the preceding sections we have described an algorithm for surface fitting with spline functions.

Beside the set of data points with the corresponding weights and the degrees of the spline, the user merely has to provide a non-negative parameter S to control the smoothness of the fit. The number of knots and their position are then determined

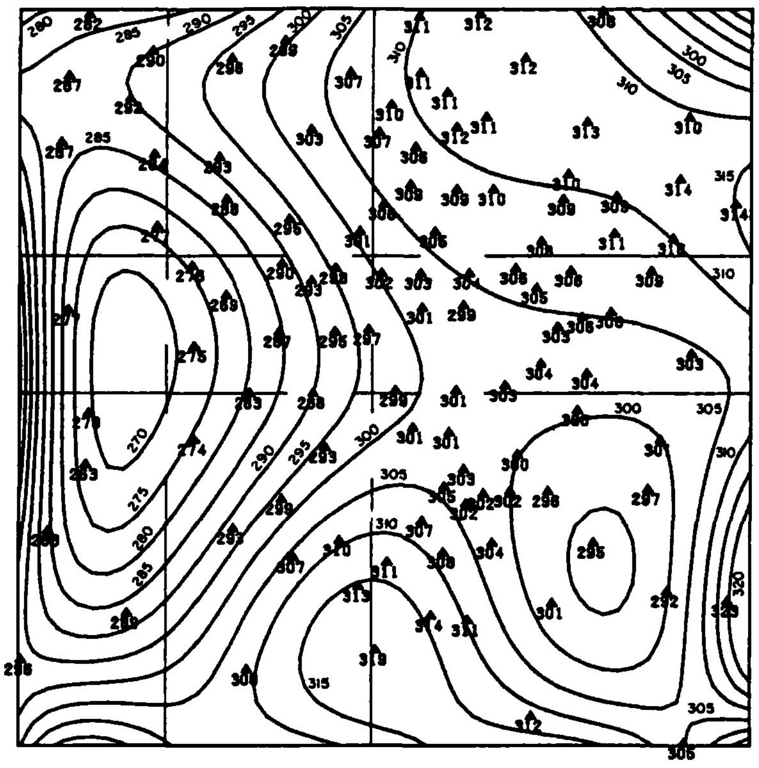


FIG. 5. The smoothing of geopotential height data; $S = 500$.

automatically. The spline is represented by means of tensor product B-splines. The coefficients in this representation are obtained by solution of a banded linear system, the bandwidth of which depends on the degree of the spline and on the number of knots. Once a smoothing spline is obtained, evaluation, differentiation and integration can be performed in a rapid and accurate way while using the recurrence schemes for evaluating, differentiating (de Boor, 1972; Cox, 1972) and integrating (Gaffney, 1976) B-splines. The smoothing factor S has to be chosen carefully: too small S-values will result in an overfitting, too large S-values in an underfitting of the data. Therefore, if possible, the user should always examine the fit graphically before accepting it as satisfactory.

Our algorithm is flexible in the sense that for S very large it returns the least-squares polynomial and for $S = 0$ an interpolating spline. However, our algorithm was primarily conceived for smoothing data, not for (quasi) interpolation. Choosing S very small will considerably increase computation time and memory requirements. It may also cause rank deficiency of the matrix of our system and so put in danger the numerical stability of the algorithm.

As for the degree of the spline, it is recommended to choose bicubic splines. This will usually give a good compromise between efficiency (computation time, memory requirements) and quality of fit. However, if the user is interested in high order derivatives of the fit, he should raise the degree of the spline, of course at the cost of efficiency.

The proposed method can easily be extended to higher dimensions. This seems not very practical, however, because the number of coefficients in the representation of the spline and therefore computation time and memory requirements will increase enormously. Besides, judgement of the fit will be much more difficult and time consuming than in the case of two dimensions.

I would like to thank Prof. Dr R. Piessens and Mrs J. Gilis for their careful reading of the manuscript and Mr C. De Ridder for providing the data of example 5. This research is supported by the FKFO, Belgium, under grant no. 2.0021.75.

References

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

Cox, M. G. 1972 The numerical evaluation of B-splines. *J. Inst. Maths Applics* **10**, 134–149.

Cox, M. G. & Hayes, J. G. 1973 Curve fitting: a guide and suite of algorithms for the non-specialist user, *Report* NAC 26, National Physical Lab., Teddington, Middlesex.

Dierckx, P. 1975 An algorithm for smoothing, differentiation and integration of experimental data using spline functions, *J. Comput. Applied Maths* **1**, 165–184.

Dierckx, P. & Piessens, R. 1977 Calculation of Fourier coefficients of discrete functions using cubic splines, *Report* TW 32, Applied Maths & Progr. Div., K. U. Leuven (Belgium).

Dierckx, P. 1977 An algorithm for least-squares fitting of cubic spline surfaces to functions on a rectilinear mesh over a rectangle. *J. Comput. Applied Maths* **3**, 113–129.

Dierckx, P. 1980 An algorithm for surface fitting with spline functions, *Report* TW 50, Applied Maths & Progr. Div., K. U. Leuven (Belgium).

Gaffney, P. W. 1976 The calculation of indefinite integrals of B-splines, *J. Inst. Maths Applics* **17**, 37–41.

Gentleman, M. C. 1973 Least-squares computations by Givens transformations without square roots. *J. Inst. Maths Applics* **12**, 329–336.

Hayes, J. G. & Halliday, J. 1974 The least-squares fitting of cubic spline surfaces to general data sets. *J. Inst. Maths Applics* **14**, 89–103.

Jarratt, P. & Nudds, D. 1965 The use of rational functions in the iterative solution of equations on a digital computer. *Comput. J.* **8**, 62–65.

Peters, G. & Wilkinson, J. H. 1970 The least-squares problem and pseudo-inverses. *Comput. J.* **13**, 309–316.

Reinsch, C. 1967 Smoothing by spline functions. *Num. Math.* **10**, 177–183.