Aplikace matematiky

Jiří Kobza

An algorithm for biparabolic spline

Aplikace matematiky, Vol. 32 (1987), No. 5, 401–413

Persistent URL: `http://dml.cz/dmlcz/104270`

32 (1987) APLIKACE MATEMATIKY No. 5, 401—413

AN ALGORITHM FOR BIPARABOLIC SPLINE

JIŘÍ KOBZA

(Received July 7, 1986)

Summary. The paper deals with the computation of suitably chosen parameters of a biparabolic spline (of the tensor product type) on a rectangular domain. Some possibilities of choosing such local parameters (concentrated, dispersed parameters) are discussed. The algorithms for computation of dispersed parameters (using the first derivative representation) and concentraced parameters (using the second derivative representation) are given. Both these algorithms repeatedly use the one-dimensional algorithms.

Keywords: spline functions, biparabolic splines, surface approximation

AMS classification: 41A15, 65D05

1. INTRODUCTION

Let us have a rectangular domain $D = \{(x, y);\ a \leqq x \leqq b,\ c \leqq y \leqq d\}$ in the (x, y)-plane with two sets of knots in each variable (the knots of a parabolic spline x_i, y_j; the knots of interpolation t_i, v_j in the variables x, y)

$(\Delta x, \Delta t)$

$$a = x_0 = t_0 < x_1 < t_1 < x_2 < t_2 < \ldots < x_{n-1} < t_{n-1} < x_n < t_n = x_{n+1} = b$$

$(\Delta y, \Delta v)$

$$c = y_0 = v_0 < y_1 < v_1 < y_2 < v_2 < \ldots < y_{m-1} < v_{m-1} < y_m < v_m = y_{m+1} = d\,.$$

As is well-known, under quite weak assumptions on the sets of knots there exists a unique one-dimensional parabolic spline in each variable interpolating the given values at the knots of interpolation and fulfilling appropriate boundary conditions (see e.g. [2], [3], [4]). In the case of bicubic splines it is known how to use one-dimensional algorithms for computation of the parameters for a two-dimensional spline on a rectangle (see [4] – [6]). The purpose of this paper is to show that with an appropriate representation of a biparabolic spline on the rectangle we may choose such a one-dimensional algorithm for a parabolic spline that we obtain the two-dimensional algorithm for the biparabolic spline which works completely analogously to the bicubic case (see also [5], p. 103 – 104).

2. FORMULATION OF THE PROBLEM

The given sets of knots $(\Delta x, \Delta t)$, $(\Delta y, \Delta v)$ divide the rectangle D into small rectangles $D_{ij} = \{(x, y);\ x \in [x_i, x_{i+1}],\ y \in [y_j, y_{j+1}]\}$. Thus, we have three different types of rectangles D_{ij} – lying in the interior, on the boundary, or at the corners of the rectangle D. The knots (points) of interpolation $P_{ij} = (t_i, v_j)$ lie inside, on the boundary or at the corners of the rectangle D_{ij}, respectively – see Fig. 1.

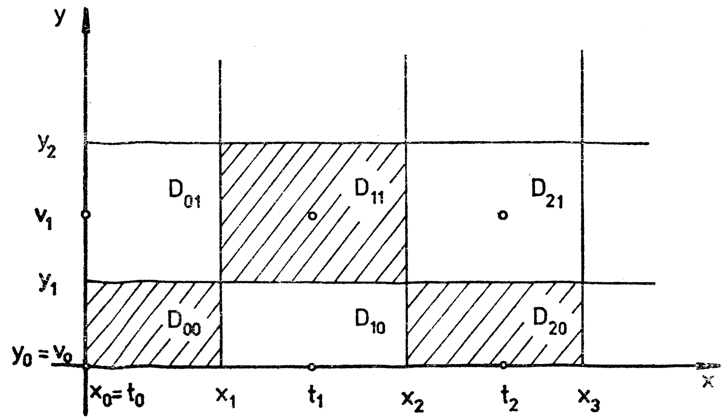

Fig. 1.

Definition 1. *$S(x, y)$ is called a biparabolic interpolating spline on the rectangle D with the knots $(\Delta x, \Delta t)$, $(\Delta y, \Delta v)$ to the given data (f_{ij}), $i = 0(1)\,n$, $j = 0(1)\,m$ if it has the following properties*:

1° $S(x, y) \in C^{(1,1)}(D)$ *(continuous first derivatives $S_x = S_{10}$, $S_y = S_{01}$ and mixed derivative $S_{xy} = S_{11}$)*;

2° *$S(x, y)$ is a polynomial of the second degree with respect to x, y on each rectangle D_{ij}, $i = 0(1)\,n$, $j = 0(1)\,m$*;

3° $S(t_i, v_j) = S(P_{ij}) = f_{ij}$, $i = 0(1)\,n$, $j = 0(1)\,m$ *(interpolation at the points $P_{ij} = (t_i, v_j)$)*.

For $S(x, y)$ we may use the piecewise polynomial representation

$$(P)\qquad S(x, y) = \\ = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^2y + a_7xy^2 + a_8x^2y^2$$

on each rectangle D_{ij} with nine coefficients a_k, $k = 0(1)\,8$, which are generally different on different rectangles D_{ij} (we omit here the indices i, j at the coefficients a_k).

The questions of existence and uniqueness are generally treated in ([1], [4]); here we find that with given $(\Delta x, \Delta t)$, $(\Delta y, \Delta v)$, (f_{ij}), the spline $S(x, y)$ on D is uniquely determined by the boundary conditions of various types – e.g.

a) $S_{10}(t_i, y_0)$, $S_{10}(t_i, y_{m+1})$ $i = 0(1)\,m$
$S_{01}(x_0, v_j)$, $S_{01}(x_{n+1}, v_j)$ $j = 0(1)\,m$ (*prescribed values*)
$S_{11}(x_k, y_p)$, $k = 0, n+1$, $p = 0, m+1$;

b) $S_{20}(x_0, v_j)$, $S_{20}(x_{n+1}, v_j)$ $j = 0(1)\,m$
$S_{02}(t_i, y_0)$, $S_{02}(t_i, y_{m+1})$ $i = 0(1)\,n$
$S_{11}(x_k, y_p)$, $k = 0, n+1$, $p = 0, m+1$

(*we use the notation, e.g.*, $S_{12}(x_i, y_j) = S_{xyy}(x_i, y_j)$).
We will use also another type of boundary conditions in our algorithm ($D1$) (with interchanged first derivatives on the horizontal and vertical boundaries as compared to the case a)); the existence and uniqueness follows from the algorithm and can be proved analogously.

To be able to work with the spline $S(x, y)$ we need the algorithm which for $S(x, y)$ computes the local parameters on D_{ij} from the global data defining the spline on D. These local parameters may be concentrated at one point of D_{ij} or dispersed at several ones.

3. CONCENTRATED PARAMETERS

The piecewise-polynomial representation (P) of the spline $S(x, y)$ defines uniquely the spline and its derivatives at all points of D_{ij}. On the other hand, given the values of $S(x, y)$ and its derivatives

$$(T) \qquad S, S_{10}, S_{01}, S_{20}, S_{11}, S_{02}, S_{21}, S_{12}, S_{22}$$

at some point $(x, y) \in D_{ij}$, we can compute the coefficients a_k of the representation (P) using the relations

$$\begin{aligned}
&(1) & a_0 &= S - xS_{10} - yS_{01} + (x^2S_{20} + 2xyS_{11} + y^2S_{02})/2 - \\
&(\mathrm{T-P}) & &\quad - (x^2yS_{21} + xy^2S_{12})/2 - x^2y^2S_{22}/4\,, \\
& & a_1 &= S_{10} - xS_{20} - yS_{02} + y^2S_{12}/2 - xy^2S_{22}/\,2\,, \\
& & a_4 &= S_{11} - xS_{21} - yS_{12} + xyS_{22}\,, \\
& & 2a_3 &= S_{20} - yS_{21} + y^2S_{22}/2\,, \quad 2a_5 = S_{02} - xS_{12} + x^2S_{22}/2\,, \\
& & 2a_6 &= S_{21} - yS_{22}\,, \qquad\qquad 2a_7 = S_{12} - xS_{22}\,, \\
& & 4a_8 &= S_{22}\,.
\end{aligned}$$

4. DISPERSED PARAMETERS

From the computational (or practical) point of view we sometimes prefer the local parameters of the spline to be dispersed and separated at the corners of D_{ij} and at the points of interpolation. For example, with bicubic splines we use the parameters $S, S_{10}, S_{01}, S_{11}$ at each of the corners of the element D_{ij} (see [4]–[6]), to be able to use repeatedly the one-dimensional algorithm with the first derivatives. For a biparabolic spline, we have to choose such parameters which uniquely determine the biparabolic polynomial on D_{ij} and together with an appropriately chosen one-dimensional algorithm will result in the two-dimensional algorithm.

At first sight, the nine parameters for the biparabolic polynomial on D_{ij} can be chosen as follows:

$$f_{ij} = S(P_{ij}) = S(t_i, v_j)\,; \tag{2}$$

$$S_{10}, S_{01} \quad \textit{at each of the corners of } D_{ij}\,.$$

However, these local parameters are not suitable, as the following lemma shows.

Lemma 1. *The nine parameters* (2) *do not generally determine the biparabolic polynomial* $S(x, y)$ *on the rectangle* D_{ij}.

Proof. When we try to compute the coefficients of Taylor's expansion

$$S(x, y) = S(x_i, y_j) + S_{10}(x - x_i) + S_{01}(y - y_j) +$$
$$+ \tfrac{1}{2}[S_{20}(x - x_i)^2 + 2S_{11}(x - x_i)(y - y_j) + S_{02}(y - y_j)^2] +$$
$$+ \tfrac{1}{2}[S_{21}(x - x_i)^2 (y - y_j) + S_{12}(x - x_i)(y - y_j)^2] + \tfrac{1}{4}S_{22}(x - x_i)^2 (y - y_j)^2$$

of $S(x, y)$ at the point (x_i, y_j) (all derivatives S_{rs} are taken at this point) from the given information (2), we obtain a system of linear equations which generally has no solution.

We obtain the same result when trying to compute the coefficients of Taylor's expansion of $S(x, y)$ at the point (t_i, v_j), or to compute the coefficients of the piece-wise polynomial representation of $S(x, y)$ on D_{ij} from such data.

We shall explain the details on the following example. Given S_{10}, S_{01} at the corners of the unit square $[0, 1] \times [0, 1]$ and the value $S(1/2, 1/2) = p$, we try to find the coefficients a_k of the (P) – representation to fit the given data. By comparison we obtain the system of equations

$$\begin{aligned}
(3) \qquad & a_1 = S_{10}(0, 0)\,, \qquad a_2 = S_{01}(0, 0)\,,\\
& a_1 + 2a_3 = S_{10}(1, 0)\,, \quad a_2 + 2a_5 = S_{01}(0, 1)\,,\\
& a_2 + a_4 + a_6 = S_{01}(1, 0)\,,\\
& a_1 + a_4 + a_7 = S_{10}(0, 1)\,,\\
& a_1 + 2a_3 + a_4 + 2a_6 + a_7 + 2a_8 = S_{10}(1, 1),\\
& a_2 + a_4 + 2a_5 + a_6 + 2a_7 + 2a_8 = S_{01}(1, 1)\,,\\
& a_0 + a_1/2 + a_2/2 + (a_3 + a_4 + a_5)/4 + (a_6 + a_7 + a_8)/8 = p\,.
\end{aligned}$$

We can eliminate a_1, a_2, a_3, a_5 from the first four equations and a_0 from the last equation, but the rest of the system forms a system of four equations with the determinant equal to zero – so we generally have no solution of the whole system.

5.1. On the rectangle $D_{ij} = \{(x, y); x_i \leqq x \leqq x_{i+1}, y_j \leqq y \leqq y_{j+1}\}$ we denote

$$f_{ij} = S(t_i, v_j) \quad (\textit{the function value at the point of interpolation}) ;$$

$$(R_{ij}) \qquad m_{ij} = S_{10}(x_i, v_j)\,, \quad m_{i+1,j} = S_{10}(x_{i+1}, v_j)\,,$$
$$n_{ij} = S_{01}(t_i, v_j)\,, \quad n_{i,j+1} = S_{01}(t_i, y_{j+1})$$

(*the values of the first derivatives at the intermediate points on the boundary of* D_{ij}) ;

$$s_{ij} = S_{11}(x_i, y_j)\,, \quad s_{i+1,j} = s_{i+1,j} = S_{11}(x_{i+1}, y_j)\,,$$
$$s_{i,j+1} = S_{11}(x_i, y_{j+1})\,, \quad s_{i+1,j+1} = S_{11}(x_{i+1}, y_{j+1})$$

(*the values of the mixed derivative at the corners of* D_{ij}).

Altogether we again have nine parameters for $S(x, y)$ dispersed on D_{ij} as shown in Fig. 2.

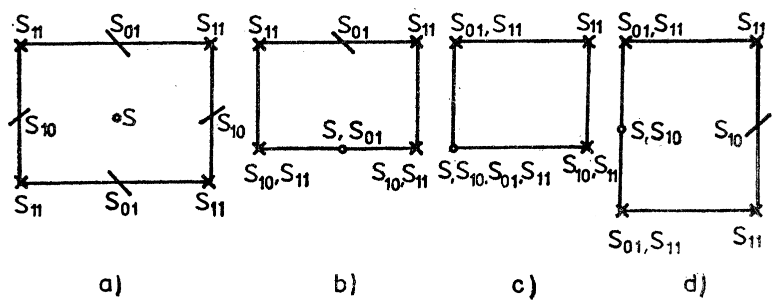

Fig. 2.

Theorem 1. *The nine parameters* (R_{ij}) *determine uniquely a biparabolic polynomial* $S(x, y)$ *on the rectangle* D_{ij}.

Proof. Let us denote

$$(4) \qquad h_i = x_{i+1} - x_i\,, \quad k_j = y_{j+1} - y_j\,,$$
$$t_i - x_i = p_i\,, \quad x_{i+1} - t_i = x_i + h_i - t_i = h_i - p_i\,,$$
$$v_j - y_j = q_j\,, \quad y_{j+1} - v_j = y_j + k_j - v_j = k_j - q_j\,,$$
$$S_{10}, S_{01}, S_{20}, S_{11}, S_{02}, S_{21}, S_{12}, S_{22} -$$

the values of the derivatives of $S(x, y)$ *at the point* $P_{ij} = (t_i, v_j)$.

From Taylor's expansion of $S(x, y)$ at the point P_{ij}

$$(T_{ij}) \qquad S(x, y) = f_{ij} + S_{10}(x - t_i) + S_{01}(y - v_j) +$$
$$+ \tfrac{1}{2}[S_{20}(x - t_i)^2 + 2S_{11}(x - t_i)(y - v_j) + S_{02}(y - v_j)^2] +$$
$$+ \tfrac{1}{2}[S_{21}(x - t_i)^2 (y - v_j) + S_{12}(x - t_i)(y - v_j)^2] +$$
$$+ \tfrac{1}{4}S_{22}(x - t_i)^2 (y - v_j)^2$$

we obtain

$$S_{10}(x, y) = S_{10} + S_{20}(x - t_i) + S_{11}(y - v_j) + S_{21}(x - t_i)(y - v_j) + \\ + \tfrac{1}{2}S_{12}(y - v_j)^2 + \tfrac{1}{2}S_{22}(x - t_i)(y - v_j)^2 ,$$

$$S_{01}(x, y) = S_{01} + S_{11}(x - t_i) + S_{02}(y - v_j) + \tfrac{1}{2}S_{21}(x - t_i)^2 + \\ + S_{12}(x - t_i)(y - v_j) + \tfrac{1}{2}S_{22}(x - t_i)^2 (y - v_j) ,$$

$$S_{11}(x, y) = S_{11} + S_{21}(x - t_i) + S_{12}(y - v_j) + S_{22}(x - t_i)(y - v_j) .$$

Applying these expansions to the parameters in (R_{ij}) at the appropriate points, we get the following system of linear equations for the eight parameters S_{kr} $(k, r = 0, 1, 2)$:

$$\begin{aligned}
(5) \qquad & m_{ij} = S_{10} - p_i S_{20} , \\
& m_{i+1,j} = S_{10} + (h_i - p_i) S_{20} , \\
& n_{ij} = S_{01} - q_j S_{02} , \\
& n_{i,j+1} = S_{01} + (k_j - q_j) S_{02} , \\
& s_{ij} = S_{11} - p_i S_{21} - q_j S_{12} + p_i q_j S_{22} , \\
& s_{i+1,j} = S_{11} + (h_i - p_i) S_{21} - q_j S_{12} - (h_i - p_i) q_j S_{22} , \\
& s_{i,j+1} = S_{11} - p_i S_{21} + (k_j - q_j) S_{12} - p_i(k_j - q_j) S_{22} , \\
& s_{i+1,j+1} = S_{11} - (h_i - p_i) S_{21} + (k_j - q_j) S_{12} + (h_i - p_i)(k_j - q_j) S_{22} .
\end{aligned}$$

This system has a unique solution

$$\begin{aligned}
(R - T) \qquad & S_{20} = (m_{i+1,j} - m_{ij})/h_i , \quad S_{02} = (n_{i,j+1} - n_{ij})/k_j , \\
& S_{10} = m_{ij} + m_{i+1,j} \cdot p_i/h_i , \quad S_{01} = n_{ij} + n_{i,j+1} \cdot q_j/k_j , \\
& S_{22} = (s_{i+1,j+1} - s_{i+1,j} - s_{i,j+1} + s_{ij})/(h_i k_j) , \\
& S_{21} = (s_{i+1,j} - s_{ij})/h_i + q_j S_{22} , \\
& S_{12} = (s_{i,j+1} - s_{ij})/k_j + p_i S_{22} , \\
& S_{11} = p_i S_{21} + q_j S_{12} - p_i q_j S_{22} + s_{ij} .
\end{aligned}$$

The boundary rectangles D_{0j}, $D_{n+1,j}$, D_{i0}, $D_{i,m+1}$ can be treated analogously by putting $p_0 = 0$, $p_{m+1} = 0$, $q_0 = 0$, $q_{n+1} = 0$, respectively.

5.2. Similarly, we can compute all coefficients of the piecewise polynomial representation (P) from the data (R_{ij}). Applying (P) to the parameters in (R_{ij}), we obtain the relations

$$\begin{aligned}
(6) \qquad & f_{ij} = a_0 + a_1 t_i + a_2 v_j + a_3 t_i^2 + a_4 t_i v_j + a_5 v_j^2 + a_6 t_i^2 v_j + a_7 t_i v_j^2 + a_8 t_i^2 v_j^2 , \\
& m_{ij} = a_1 + 2a_3 x_i + a_4 v_j + 2a_6 x_i v_j + a_7 v_j^2 + 2a_8 x_i v_j^2 , \\
& m_{i,j+1} = a_1 + 2a_3 x_{i+1} + a_4 v_j + 2a_6 x_{i+1} v_j + a_7 v_j^2 + 2a_8 x_{i+1} v_j^2 , \\
& n_{ij} = a_2 + a_4 t_i + 2a_5 y_j + a_6 t_i^2 + 2a_7 t_i y_j + 2a_8 t_i^2 y_j , \\
& n_{i,j+1} = a_2 + a_4 t_i + 2a_5 y_{j+1} + a_6 t_i^2 + 2a_7 t_i y_{j+1} + 2a_8 t_i^2 y_{j+1} ,
\end{aligned}$$

$$
\begin{aligned}
s_{ij} &= a_4 + 2a_6x_i + 2a_7y_j + 4a_8x_iy_j\,,\\
s_{i+1,j} &= a_4 + 2a_6x_{i+1} + 2a_7y_j + 4a_8x_{i+1}y_j\,,\\
s_{i,j+1} &= a_4 + 2a_6x_i + 2a_7y_{j+1} + 4a_8x_iy_{j+1}\,,\\
s_{i+1,j+1} &= a_4 + 2a_6x_{i+1} + 2a_7y_{j+1} + 4a_8x_{i+1}y_{j+1}\,.
\end{aligned}
$$

The last four relations form a system of linear equations for a_4, a_6, a_7, a_8 with a unique solution; then we can calculate a_1, a_3, a_2, a_5 from the second till the fifth relation and, finally, a_0 from the first relation. We obtain

$$
(R-P)\quad
\begin{aligned}
a_4 &= s_{ij} - (s_{i+1,j} - s_{ij})\,x_i/h_i - (s_{i,j+1} - s_{ij})\,y_j/k_j + \\
&\quad + (s_{i+1,j+1} - s_{i+1,j} - s_{i,j+1} + s_{ij})\,x_iy_j/(h_ik_j)\\
a_6 &= (s_{i+1,j} - s_{ij})\,y_{j+1} - (s_{i+1,j+1} - s_{i,j+1})\,y_j/(2h_ik_j)\\
a_7 &= (s_{i,j+1} - s_{ij})\,x_{i+1} - (s_{i+1,j+1} - s_{i+1,j})\,x_i/(2h_ik_j)\\
a_8 &= (s_{i+1,j+1} - s_{i+1,j} - s_{i,j+1} + s_{ij})/(4h_ik_j)\\
a_3 &= (m_{i,j+1} - m_{ij})/(2h_i) - v_j(a_6 + a_8v_j)\\
a_1 &= m_{ij} - (a_4 + a_7v_j)\,v_j - 2x_iv_j(a_6 + a_8v_j) - 2x_ia_3\\
a_5 &= (n_{i,j+1} - n_{ij})/(2k_j) - t_i(a_7 + a_8t_i)\\
a_2 &= n_{ij} - (a_4 + a_6t_i)\,t_i - 2t_iy_j(a_7 + a_8t_i) - 2a_5y_j\\
a_0 &= f_{ij} - a_1t_i - a_2v_j - a_3t_i^2 - a_4t_iv_j - a_5v_j^2 - a_6t_i^2v_j - \\
&\quad - a_7t_iv_j^2 - a_8t_i^2v_j^2\,.
\end{aligned}
$$

5.3. It is a little surprising that the interchange of the derivatives S_{01} and S_{01} as in Fig. 3 leads to an unsuitable set of parameters for $S(x, y)$.

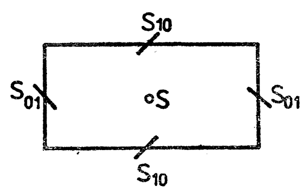

Fig. 3.

Lemma 2. *The nine parameters*

$$
(\bar{R}_{ij})\quad
\begin{aligned}
&S(t_i, v_j),\ S_{10}(t_i, y_j),\ S_{10}(t_i, y_{j+1}),\ S_{01}(x_i, v_j),\ S_{01}(x_{i+1}, v_j)\,,\\
&S_{11}(x_i, y_j),\ S_{11}(x_{i+1}, y_j),\ S_{11}(x_i, y_{j+1}),\ S_{11}(x_{i+1}, y_{j+1})
\end{aligned}
$$

do not generally determine a biparabolic polynomial on D_{ij}.

Proof. Denoting (for this proof only)

$$
m_{ij} = S_{10}(t_i, y_j)\,, \quad n_{ij} = S_{01}(x_i, v_j)\,, \quad s_{ij} = S_{11}(x_i, y_j)
$$

we get analogously to (5) the system of equations with different first four equations

$$(7) \qquad \begin{aligned} m_{ij} &= S_{10} - q_j S_{11} + q_j^2 S_{12}/2\,, \\ m_{i,j+1} &= S_{10} + (k_j - q_j)\, S_{11} + (k_j - q_j)^2\, S_{12}/2\,, \\ n_{ij} &= S_{01} - p_i S_{11} + p_i^2 S_{21}/2\,, \\ n_{i+1,j} &= S_{01} + S_{11}(h_i - p_i) + (h_i - p_i)^2\, S_{21}/2\,. \end{aligned}$$

The next four equations coincide with those in (5). We have now eight equations, but six unknowns only (S_{20}, S_{02} do not appear in this system) – the system is overdetermined and generally has no solution.

6. ALGORITHMS FOR ONE-DIMENSIONAL PARABOLIC SPLINES

6.1. Algorithm with the first derivatives

As is known (see e.g. [2], [3], [5]) a one-dimensional parabolic spline $S_2(x)$ for a given set of knots $(\Delta x, \Delta t)$ and prescribed values (g_i) at the points of interpolation (t_i) can be expressed by

$$(8) \qquad S_2(x) = g_i + (x - x_i - p_i)\left[m_i + (m_{i+1} - m_i)(x - x_i + p_i)/(2h_i)\right]$$

for $x \in [x_i, x_{i+1}]$, where $h_i = x_{i+1} - x_i$, $p_i = t_i - x_i$, $g_i = S_2(t_i)$, $m_i = S_2'(x_i)$. The continuity of $S_2(x)$, $S_2'(x)$ at the knots x_i, $i = 1(1)\,n$ yields the following relations for the parameters m_i, $i = 0(1)\,n + 1$:

$$(m) \qquad a_i m_{i-1} + b_i m_i + c_i m_{i+1} = f_i\,, \quad i = 1(1)\,n\,,$$

where

$$\begin{aligned} a_i &= (h_{i-1} - p_{i-1})^2 > 0\,, \\ (9) \qquad b_i &= p_i(2h_i - p_i)\, h_{i-1}/h_i + (h_{i-1}^2 - p_{i-1}^2) > 0\,, \\ c_i &= p_i^2 h_{i-1}/h_i > 0\,, \\ f_i &= 2h_{i-1}(g_i - g_{i-1})\,. \end{aligned}$$

We have to choose two other (usually boundary or periodicity) conditions to determine the spline uniquely (the simplest possibility is to prescribe m_0, m_{n+1} and to solve the tridiagonal system (m)). As shown in detail in [2], under quite weak conditions on the geometry of the sets of knots $(\Delta x, \Delta t)$,

$$|b_0| > |c_0|,\ |b_{n+1}| > |a_{n+1}|\,,$$

the boundary conditions usually used (prescribed values of the first or second derivative, or linear combinations of their values) lead to a tridiagonal system of linear equations with strictly dominating diagonal. Periodicity conditions lead to a system with a cyclic tridiagonal matrix. So in such cases we have a unique solution (m_i) which can be computed effectively by algorithms for special systems of this kind.

6.2. Algorithm with the second derivatives

We have also another representation for the parabolic spline. Denoting

$$(10) \qquad M_i = S_2''(t_i)\,, \quad i = 0(1)\,n\,,$$

$$n_i = S_2'(t_i) = (g_{i+1} - g_i)/(\Delta t_i) - (x_{i+1} - t_i)(t_{i+1} - x_{i+1} + \Delta t_i)\,M_i/(2\Delta t_i) - $$
$$- (t_{i+1} - x_{i+1})^2\,M_{i+1}/(2\Delta t_i)\,, \quad i = 0(1)\,n - 1\,,$$

$$n_n = \{g_n - g_{n-1} + [(x_n - t_{n-1})^2\,M_{n-1} + (t_n - x_n)(\Delta t_{n-1} + x_n - t_{n-1})\;M_n]/2\} :$$
$$: \Delta t_{n-1}$$

we have

$$(11) \qquad S_2(x) = g_i + n_i(x - t_i) + M_i(x - t_i)^2/2 \quad for \quad x \in [x_i, x_{i+1}]\,.$$

The continuity conditions of the spline at the knots lead to the relations

$$(M) \qquad a_i M_{i-1} + b_i M_i + c_i M_{i+1} = f_i\,, \quad i = 1(1)\,n - 1$$

with

$$a_i = [(x_i - t_{i-1})/\Delta t_{i-1}]^2\,\Delta t_{i-1}/(\Delta t_{i-1} + \Delta t_i)\,,$$
$$b_i = [(t_i - x_i)(1 + (x_i - t_{i-1})/\Delta t_{i-1}) +$$
$$+ (x_{i+1} - t_i)(1 + (t_{i+1} - x_{i+1})/\Delta t_i)]/(\Delta t_{i-1} + \Delta t_i)$$

$$(12) \qquad c_i = [(t_{i+1} - x_{i+1})/\Delta t_i]^2\,\Delta t_i/(\Delta t_{i-1} + \Delta t_i)$$
$$f_i = 2[(g_{i+1} - g_i)/\Delta t_i - (g_i - g_{i-1})/\Delta t_{i-1}]/(\Delta t_{i-1} + \Delta t_i)$$

for the unknown values M_i. Two boundary conditions (e.g. conditions fixing M_0, M_{n+1} or the values of the first derivative, or some periodicity conditions) complete this system to a tridiagonal (or cyclic tridiagonal, under periodicity conditions) system of linear equations. More details are given in [2].

7. ALGORITHM FOR BIPARABOLIC SPLINE USING THE FIRST DERIVATIVES

7.1. Suppose the biparabolic spline $S(x, y)$ is uniquely determined by the data

$$(13) \quad (\Delta x, \Delta t)\,, \quad (\Delta y, \Delta v) \quad (\textit{the sets of knots})\,,$$

$$f_{ij} = S(t_i, v_j) \qquad i = 0(1)\,n\,; \qquad j = 0(1)\,m \quad (\textit{conditions of interpolation}),$$

$$\left.\begin{aligned} m_{ij} &= S_{10}(x_i, v_j) & i &= 0, n + 1\,; & j &= 0(1)\,m \\ n_{ij} &= S_{01}(t_i, y_j) & j &= 0, m + 1\,; & i &= 0(1)\,n \\ s_{ij} &= S_{11}(x_i, y_j) & i &= 0, n + 1\,; & j &= 0, m + 1 \end{aligned}\right\} \quad (\textit{boundary conditions})\,.$$

The nine dispersed parameters (R_{ij}) defining the spline $S(x, y)$ on each of the rectangles D_{ij} according to Theorem 1 can be computed from (13) via the following algorithm, based on the one-dimensional algorithm given in 6.1:

Algorithm D 1

1° Compute $m_{ij} = S_{10}(x_i, v_j)$, $i = 1(1)\,n$ on the horizontal lines $y = v_j$, $j = 0(1)\,m$ from the values f_{ij}, m_{0j}, $m_{n+1,j}$.

2° Compute $n_{ij} = S_{01}(t_i, y_j)$, $j = 1(1)\, m$ on the vertical lines $x = t_i$, $i = 0(1)\, n$ using the values f_{ij}, n_{i0}, $n_{i,m+1}$.

3° Compute $s_{ij} = S_{11}(x_i, y_j)$, $j = 1(1)\, m$ on two vertical boundaries $(i = 0, n + 1)$ using the values m_{ij}, s_{i0}, $s_{i,m+1}$
(here we use the fact that $S_{10}(x, y)$ is a parabolic spline with respect to the variable y, determined by the values of S_{10} and the boundary values S_{11} on a vertical line; likewise, S_{01} is a parabolic spline with respect to the variable x determined by the values of S_{01} and the boundary values S_{11} on a horizontal line).

4° Compute $s_{ij} = S_{11}(x_i, y_j)$, $i = 1(1)\, n$ from the values n_{ij}, s_{0j}, $s_{n+1,j}$, $j = 0(1)\, m + 1$ on horizontal lines.

In all these four steps we use the one-dimensional algorithm as described in 6.1, working with tridiagonal systems of linear equations. As the result we obtain all the nine dispersed parameters (R_{ij}) of the spline $S(x, y)$ on each rectangle D_{ij}. Using relations $(R - T)$ or $(R - P)$, we can pass to either Taylor's or a piecewise-polynomial representation of $S(x, y)$.

If we need to compute the function value of the spline $S(x, y)$ at a given point (x, y), we may repeatedly apply formula (8) in the following algorithm.

Algorithm FV.

1° Find i, j such that $x_i \leqq x \leqq x_{i+1}$, $y_j \leqq y \leqq y_{j+1}$ $(x, y$ given$)$.

2° Compute $S(x, v_j)$ using m_{ij}, S, $m_{i+1,j}$ (formula (8)).

3° Compute $S_{01}(x, y_j)$ from s_{ij}, n_{ij}, $s_{i+1,j}$,
$S_{01}(x, y_{j+1})$ from $s_{i,j+1}$, $n_{i,j+1}$, $s_{i+1,j+1}$.

4° Compute $S(x, y)$ using $S_{01}(x, y_j)$, $S(x, v_j)$, $S_{01}(x, y_{j+1})$.

7.2. Other types of boundary conditions

The algorithm described in 7.1 can be used with boundary conditions of a more general type

$$
(14) \qquad
\begin{aligned}
& \left.\begin{aligned}
& b_{0j}m_{0j} + c_{0j}m_{1j} = f_{0j}\,, \\
& a_{n+1,j}m_{nj} + b_{n+1,j}m_{n+1,j} = f_{n+1,j}\,,
\end{aligned}\right\} \quad j = 0(1)\, m \\
& \left.\begin{aligned}
& b_{i0}n_{i0} + c_{i0}n_{i1} = f_{i0}\,, \\
& a_{i,m+1}n_{im} + b_{i,m+1}n_{i,m+1} = f_{i,m+1}\,,
\end{aligned}\right\} \quad i = 0(1)\, n \\
& s_{ij}\,, \quad i = 0, n + 1\,; \quad j = 0, m + 1 \quad \textit{given}\,.
\end{aligned}
$$

Such a type of boundary conditions includes several important cases:

a) The values m_{0j}, $m_{n+1,j}$, n_{i0}, $n_{i,m+1}$ are prescribed as in the basic Algorithm **D 1**.

b) The values of the second derivatives are given on the boundary

$$r_{0j} = S_{20}(x_0, v_j)\,, \quad r_{n+1,j} = S_{20}(x_{n+1}, v_j)\,, \quad j = 0(1)\,m + 1\,,$$
$$p_{i0} = S_{02}(t_i, y_0)\,, \quad p_{i,m+1} = S_{02}(t_i, y_{m+1})\,, \quad i = 0(1)\,n + 1\,,$$
$$s_{00}, \quad s_{0,m+1}, \; s_{n+1,0}, \; s_{n+1,m+1}\,.$$

(A combination of the first and second derivatives on the vertical and horizontal lines near the boundary is also possible.) For transforming such boundary conditions to the type (14), we use the relations

$$r_{0j} = (m_{1,j} - m_{0j})/h_0\,, \quad r_{n+1,j} = (m_{n+1,j} - m_{nj})/h_n$$
$$p_{i0} = (n_{i1} - n_{i0})/k_0\,, \qquad p_{i,m+1} = (n_{i,m+1} - n_{im})/k_m\,.$$

c) The approximation of the first or second derivatives of $S(x, y)$ using the given function values f_{ij} and an appropriate formula of numerical differentiation (if there is no reason to prescribe the boundary conditions differently).

7.3. The periodicity conditions for the one-dimensional spline lead to an algorithm in which we have to solve a cyclic tridiagonal system of linear equations (see [2]). With the help of this algorithm (in combination with the foregoing) we get, for the two-dimensional case on a rectangle:

a) an algorithm for the two-periodical spline,

b) an algorithm for the spline which is periodical in one variable and fulfils some other type of boundary conditions in the second variable.

8. ALGORITHM FOR BIPARABOLIC SPLINE USING THE SECOND DERIVATIVES

Using repeatedly the algorithm described in 6.2, we can compute concentrated local parameters (T) of the spline $S(x, y)$ determined for all D_{ij} by the global values

$$(14) \qquad \begin{array}{ll} (\Delta x, \Delta t)\,, \; (\Delta y, \Delta v) & (\textit{the sets of knots})\,, \\ f_{ij} = S(t_i, v_j) \quad i = 0(1)\,n\,, \quad j = 0(1)\,m & (\textit{interpolation conditions})\,, \\ \left.\begin{array}{lll} S_{20}(t_i, v_j) & i = 0, n\,; & j = 0(1)\,m \\ S_{02}(t_i, v_j) & j = 0, m\,; & i = 0(1)\,n \\ S_{22}(t_i, v_j) & i = 0, n\,; & j = 0, m \end{array}\right\} & (\textit{boundary conditions})\,. \end{array}$$

Algorithm D 2

1° Compute $S_{20}(t_i, v_j)\,, \quad i = 0(1)\,n$
$S_{10}(t_i, v_j)\,, \quad i = 0(1)\,n \quad$ (using (10))
from the values f_{ij}, $S_{20}(t_0, v_j)$, $S_{20}(t_n, v_j)$, $j = 0(1)\,m$ (horizontal lines).

2° Compute $S_{02}(t_i, v_j)\,, \quad j = 0(1)\,m$
$S_{01}(t_i, v_j)\,, \quad j = 0(1)\,m$

from the values $S_{02}(t_i, v_0)$, $S_{02}(t_i, v_{m+1})$, f_{ij}, $i = 0(1)n$ (vertical lines).

3° Compute $S_{22}(t_0, v_j)$,
$S_{22}(t_n, v_j)$ $\quad j = 0(1)\, m$
$S_{21}(t_0, v_j)$, $S_{21}(t_n, v_j)$

using the values on vertical boundaries

$$S_{20}(t_0, v_j)\,, \quad S_{22}(t_0, v_0)\,, \quad S_{22}(t_0, v_m)$$
$$S_{20}(t_n, v_j)\,, \quad S_{22}(t_n, v_0)\,, \quad S_{22}(t_n, v_m)\,; \quad j = 0(1)\, m\,.$$

4° Compute $\begin{matrix} S_{22}(t_i, v_j)\,, & i = 1(1)\, n - 1 \\ S_{12}(t_i, v_j)\,, & i = 0(1)\, n \end{matrix} \quad j = 1(1)\, m - 1$

from the values $S_{02}(t_i, v_j)$, $S_{22}(t_0, v_j)$, $S_{22}(t_n, v_j)$ on the boundaries,

$$\begin{matrix} S_{21}(t_i, v_j)\,, & i = 1(1)\, n - 1 \\ S_{11}(t_i, v_j)\,, & i = 0(1)\, n\,, \end{matrix} \quad j = 1(1)\, m - 1$$

from the values $S_{01}(t_i, v_j)$, $S_{21}(t_0, v_j)$, $S_{21}(t_n, v_j)$ (horizontal lines).

So we have computed all the nine local parameters (T) concentrated at the point of interpolation for each rectangle D_{ij}. We can obtain a piecewise-polynomial representation for $S(x, y)$ using relations $(T - P)$ given in 3.

9. IMPLEMENTATION AND NUMERICAL RESULTS

In both algorithms described above we repeatedly have to solve systems of linear equations with tridiagonal or cyclic tridiagonal matrices. The components of these matrices depend on the geometry of the sets of knots and on the boundary conditions. Under several types of boundary conditions we have to solve many systems with the same matrix (e.g. in the case of prescribed first or second derivatives on the boundary when computing on parallel lines). In such cases the decomposition algorithms can be used sucessfully.

The first algorithm has been implemented on PMD-85 personal computer (BASIC-G) with the following possible boundary conditions

- the general type (14) of boundary conditions;
- given the first derivative on the boundary;
- given the second derivative on the boundary;
- the user has no conditions on the boundary (the program automatically approximates the first derivatives from the given function values);
- periodicity conditions.

We obtained exact results for the biparabolic polynomial $S(x, y) = x^2 - y^2 + xy - 1$ (under proper boundary conditions) and satisfactory results for freely chosen periodic data (including graphic visualisation in one dimension).

References

[1] *C. de Boor*: A practical guide to splines. Springer, N. Y. 1978.

[2] *J. Kobza:* On algorithms for parabolic splines. Acta UPO, FRN, Vol. 88, Math. XXVI (to appear)

[3] *J. Kobza:* Evaluation and mapping of parabolic interpolating spline. (Czech). Knižnica algoritmov, IX. diel, str. 51—58; JSMF Bratislava 1987.
[4] *В. Л. Макаров, В. В. Хлобыстов*: Сплайн-аппроксимация функций. Москва, Наука 1983.
[5] *С. Б. Стечкин, Й. Н. Субботин*: Сплайны в вычислительной математике. Москва, Наука 1976.
[6] *M. Schultz:* Spline analysis. Prentice-Hall, N. J. 1973.
[7] *Ю. С. Завялов, Б. И. Квасов, В. Л. Мирошниченко*: Методы сплайн-функций. Москва, Наука 1980.

Souhrn

ALGORITMY PRO BIPARABOLICKÉ SPLAJNY

JIŘÍ KOBZA

Práce se zabývá výpočtem parametrů biparabolického splajnu (typu tenzorového součinu) na obdélníkové oblasti. Po prozkoumání některých možností vhodného výběru parametrů určujících takový splajn (rozložené parametry (R_{ij}), soustředěné parametry (T)) jsou uvedeny algoritmy pro výpočet rozložených parametrů (algoritmus D 1 — s prvními derivacemi) a soustředěných parametrů (algoritmus D 2 — s druhými derivacemi) z podmínek interpolace a z vhodných okrajových podmínek. Oba algoritmy opakovaně využívají algoritmů pro výpočet parametrů jednodimenzionálního splajnu. K výpočtu funkčních hodnot splajnu je pak možno použít (PP)-reprezentace nebo algoritmus (FV) s opakovaným výpočtem hodnoty jednodimenzionálního splajnu.

Резюме

АЛГОРИТМЫ ДЛЯ ДВУМЕРНЫХ ПАРАБОЛИЧЕСКИХ СПЛАЙНОВ

JIŘÍ KOBZA

В работе изучаются алгоритмы вычисления параметров двумерного параболического сплайна на прямоугольнике. Исследуются возможности подходящего выбора параметров сплайна (разложенные параметры (R_{ij}), сосредоточенные параметры (T)). Приведены алгоритмы вычисления таких параметров из условий интерполяции и подходящих краевых условий (алгоритм D1 использует первые производные, алгоритм D2 вторые произвводные). Оба алгоритма используют циклическим образом алгоритмы для вычисления параметров одномерных сплайнов. Для вычисления значения сплайна из его параметров можно пользоваться его многочленным представлением (PP) или алгоритмом (FV), в котором повторно используется алгоритм вычисления значений одномерного сплайна.

Author's adress: Doc. RNDR. *Jiří Kobza, CSc.* PřF UP Olomouc, Gottwaldova 15, 77 146 Olomouc.