Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium. Mathematica

Jiří Kobza

On algorithms for parabolic splines

Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium. Mathematica, Vol. 26 (1987), No. 1, 169--185

Persistent URL: http://dml.cz/dmlcz/120181

ACTA UNIVERSITATIS PALACKIANAE OLOMUCENSIS
FACULTAS RERUM NATURALIUM

1987 Mathematica XXVI Vol. 88

Katedra matematické analýzy a numerické matematiky
přírodovědecké fakulty University Palackého v Olomouci
Vedoucí katedry: Miroslav Laitoch, Prof., RNDr., CSc.

ON ALGORITHMS FOR PARABOLIC SPLINES

JIŘÍ KOBZA

(Received January 15th, 1986)

Dedicated to Professor M.Laitoch on his 65th birthday

1. Introduction

The parabolic splines have as yet been most extensively studied in [6], [4]. With respect to the more general concept of the polynomial splines and other computational aspects, two different sets of knots (the points of interpolation t_i, the knots of spline x_i) are used - see also [2]. There are also various types of representation used for the parabolic spline in the intervals between the neighbouring knots. This article presents such a representation of the parabolic spline, which enables us to use the one-dimensional algorithm repeatedly in the algorithm for two-dimensional parabolic spline. This tensor product technique is frequently used for bicubic splines (see [5], [6], [7]) and for parabolic splines will be described in [3].

Definition: Let us have two sets of knots on the interval $\langle a,b\rangle$
(Δt): $a = t_0 < t_1 < \dots < t_n = b$ - knots (points) of interpolation;
(Δx): $a = x_0 < x_1 < \dots < x_n < x_{n+1} = b$ - knots of spline
with the property $t_{i-1} < x_i < t_i$, $i=1(1)n$,
and the given values $g_i \in R$, $i=0(1)n$ at the knots of interpolation.
We call $S_2(x)$ a (interpolating) parabolic spline for the values

(g_i) on the sets of knots (Δt), (Δx) if it has the following properties

1° $S_2(x)$ has a continuous first derivative on the interval $\langle a,b\rangle$

2° $S_2(x)$ is a polynomial of the second degree on every interval $\langle x_i, x_{i+1}\rangle$, $i=0(1)n$

3° $S_2(t_i) = g_i$, $i=0(1)n$ (interpolation conditions).

In the following, we choose such a representation of a parabolic spline which seems to be quite simple and enables us to build up the algorithms with first or second derivatives in a quite elementary way and - what is more important - it enables us to use this algorithm for the biparabolic splines on rectangle.

2. The algorithm with first derivatives

Given two sets of knots (Δt), (Δx) and the values (g_i) let us denote $S_2'(x_i) = m_i$, $q = x - x_i$,

$h_i = x_{i+1} - x_i$, $d_i = t_i - x_i$, $i=0(1)n$.

For the parabolic spline $S_2(x)$ we can write now

$$S_2'(x) = m_i + (m_{i+1}-m_i)(x-x_i)/h_i, \quad x \in \langle x_i, x_{i+1}\rangle$$

Integrating over the interval $\langle x_i, x\rangle$ and using the conditions of interpolation, we get the following representation for S_2:

$$\text{(Sm)} \qquad S_2(x) = g_i + (q-d_i)\left[m_i+(m_{i+1}-m_i)(q+d_i)/(2h_i)\right]$$

on the interval $\langle x_i, x_{i+1}\rangle$. Given (Δt), (Δx), (g_i) and (m_i), we can compute $S_2(x)$ for any $x \in \langle a,b\rangle$. The continuity of $S_2'(x)$ at the knots x_i is guaranteed by the common values $m_i = S_2'(x_i)$ in this representation: the values m_i are determined particularly by the condition of continuity of $S_2(x)$ at the knots x_i, $i=1(1)n$. Comparing the values for $S_2(x_i)$ from the representations on the intervals $\langle x_{i-1}, x_i\rangle$ and $\langle x_i, x_{i+1}\rangle$, we obtain the system of equations

$$\text{(1)} \qquad g_i - d_i\left[m_i + d_i(m_{i+1}-m_i)/(2h_i)\right] =$$
$$= g_{i-1} + (h_{i-1}-d_{i-1})\left[m_{i-1} + (h_{i-1}+d_{i-1})(m_i - m_{i-1})/(2h_{i-1})\right]$$

We can write it as follows

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

with
$$a_i = (h_{i-1} - d_{i-1})^2$$
$$b_i = d_i(2h_i - d_i)h_{i-1}/h_i + (h_{i-1}^2 - d_{i-1}^2)$$
$$(2) \qquad c_i = d_i^2\, h_{i-1}/h_i$$
$$f_i = 2h_{i-1}(g_i - g_{i-1}) \ .$$

We have $h_i = h$ in the special case of equidistant knots x_i; if we further choose $t_i = (x_i + x_{i+1})/2$, we get in this case $d_i = h/2$. The system of equations (m_i) has now a quite simple form (given e.g. in [6])

$$(3) \qquad m_{i-1} + 6m_i + m_{i+1} = 8(g_i - g_{i-1})/h , \qquad i=1(1)n .$$

The system (m_i) resp. (3) has n linear equations for n + 2 unknown numbers m_i, $i=0(1)n+1$; the matrix of the system is "tridiagonal". We have to choose two another conditions for the unique determination of the spline $S_2(x)$. From the computational and stability considerations (see [2], [6]), the boundary conditions of various types are the most frequently used.

2.1 The quite general type of boundary conditions

$$\text{(BC)} \qquad \begin{aligned} b_o m_o + c_o m_1 &= f_o \\ a_{n+1} m_n + b_{n+1} m_{n+1} &= f_{n+1} \end{aligned}$$

represents the frequently used types of conditions, such as

a) prescribing the values m_o, m_{n+1}
(case $c_o = a_{n+1} = 0$, $b_o = b_{n+1} = 1$, $f_o = m_o$, $f_{n+1} = m_{n+1}$);

b) approximating the values m_o, m_{n+1} from the data g_i on the boundaries, if there is no reason to choose it otherwise (e.g. $m_o = (g_1-g_o)/(t_1-t_o)$, $m_{n+1} = (g_n-g_{n-1})/(t_n-t_{n-1})$);

c) prescribing the values $S_2''(x_o)$, $S_2''(x_{n+1})$ - in using relations

$$S_2''(x_o) = (m_1 - m_o)/h_o \ , \quad S_2''(x_{n+1}) = (m_{n+1} - m_n)/h_n$$

written in the form of (BC) as

$$m_o - m_1 = -h_o S_2''(x_o) \qquad (b_o = b_{n+1} = 1, \quad c_o = a_{n+1} = -1,$$

$$-m_n + m_{n+1} = h_n S_2''(x_{n+1}) \qquad f_o = -h_o S_2''(x_o), \ f_{n+1} = h_n S_2''(x_{n+1})),$$

The continuity conditions (m_i) together with the boundary conditions (BC) give the complete tridiagonal system of linear equations for the unknown derivatives $m_i = S_2'(x_i)$, $i=0(1)n+1$

$$(m) \qquad \begin{bmatrix} b_o & c_o & & & \\ a_1 & b_1 & c_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_n & b_n & c_n \\ & & & a_{n+1} & b_{n+1} \end{bmatrix} \begin{bmatrix} m_o \\ m_1 \\ \cdot \\ m_n \\ m_{n+1} \end{bmatrix} = \begin{bmatrix} f_o \\ f_1 \\ \cdot \\ f_n \\ f_{n+1} \end{bmatrix}$$

The matrix of this system is (strictly) diagonally dominant under conditions (the conditions on the geometry of the sets of knots (Δt), (Δx))

$$(4) \qquad |b_o| \geq |c_o|, \ |b_{n+1}| \geq |a_{n+1}| \ \ (|b_o| > |c_o| \ , \ |b_{n+1}| > |a_{n+1}|)$$

(it is irreducible diagonally dominant in the case c) of (BC)), because the inequality $b_i - a_i - c_i > 0$, which is equivalent to the inequality

$$h_{i-1}(d_i + d_{i-1}) > d_{i-1}^2 + d_i^2 \, h_{i-1}/h_i \ , \quad i=1(1)n$$

is valid under our assumptions on (Δx), (Δt).
We have then the unique parabolic spline $S_2(x)$ for every prescribed values (g_i) at the knots (t_i) under those conditions.

2.2 Periodicity conditions on the spline $S_2(x)$ - which is a piecewise polynominal of the second degree- can be described in the following way

$g_n = g_o$ (the data g_i must have this property)

(Pm) $m_{n+1} = m_o$ (it reduces the number of unknowns in the system (m))

$S_2''(x_o) = S_2''(x_{n+1})$.

The last condition we can rewrite as $(m_1-m_o)/h_o = (m_{n+1}-m_n)/h_n$, or $\quad - m_n-(h_n/h_o)m_1 + (1+h_n/h_o)m_{n+1} = 0$.

Rhe continuity conditions (m_i) together with the periodicity conditions (P1) give the system of n+1 equations for n+1 unknown derivatives m_i , i = 1(1)n+1

$$(mp)\begin{bmatrix} b_1 & c_1 & & & & a_1 \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & a_n & b_n & c_n \\ c_{n+1} & & & & a_{n+1} & b_{n+1} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \cdot \\ m_n \\ m_{n+1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ f_n \\ 0 \end{bmatrix}, \quad \begin{aligned} a_{n+1} &= -1 \\ b_{n+1} &= 1+h_n/h_o \\ c_{n+1} &= -h_n/h_o \end{aligned}$$

The matrix of this system is cyclic tridiagonal; we have thus -under the conditions quite similar to (4) - unique solution of the system for any data (g_i) with $g_o = g_n$.

From the computational point of fiew, the tridiagonal system (m) is a special case of cyclic tridiagonal system (mp). So we can compute the coefficients m_i for the spline $S_2(x)$ using the subroutine for cyclic tridiagonal systems only - for all types of boundary conditions mentioned.

The values of $S_2(x)$ can be computed using the representation (Sm).

3. The algorithm with second derivatives

The algorithms with first or second derivatives for computation with cubic splines are wellknown (see [1], [6],[7]). The user can choose one or onother with respect to other needs of the problem. We have also the possibility to use the second derivatives when working with parabolic splines - the algorithms for some special cases can be found in [6],[4],[1]. The approach used there for deriving algorithm formulas seems to be too complicated for our purpose. We can show an elementary approach to derive the formulas and relations needed for our algorithm, based on a suitable form of representation of the spline $S_2(x)$.

3.1 The second derivative of $S_2(x)$ is discontinuous at the knots of spline - so it is suitable to express $S_2(x)$ by its Taylor's expansion at the point t_k on the interval $\langle x_k, x_{k+1}\rangle$ (remember that $x_k < t_k < x_{k+1}$). We have

$$S_2(x) = S_2(t_k) + S_2'(t_k)(x-t_k) + \frac{1}{2} S_2''(t_k)(x-t_k)^2, \quad k=0(1)n.$$

Denoting now $S_2(t_k) = g_k$, $S_2'(t_k) = n_k$, $S_2''(t_k) = M_k$

for $k=0(1)n$, we can write

$$S_2(x) = g_i + n_i(x-t_i) + M_i(x-t_i)^2/2 \qquad x \in \langle x_i, x_{i+1}\rangle$$

(5) for $i=0(1)n-1$

$$S_2(x) = g_{i+1}+n_{i+1}(x-t_{i+1})+M_{i+1}(x-t_{i+1})^2/2 \qquad x \in \langle x_{i+1}, x_{i+2}\rangle$$

The continuity conditions for $S_2(x)$ and $S_2'(x)$ at the knot x_{i+1}, $i=0(1)n-1$ can now be expressed as

$$g_i+n_i(x_{i+1}-t_i) + M_i(x_{i+1}-t_i)^2/2 = g_{i+1}+n_{i+1}(x_{i+1}-t_{i+1}) + $$
$$+ M_{i+1}(x_{i+1}-t_{i+1})^2/2$$

(n_i)

$$n_i + M_i(x_{i+1}-t_i) = n_{i+1} + M_{i+1}(x_{i+1}-t_{i+1}) \quad .$$

We can compute n_i, n_{i+1} uniquely from g_i, g_{i+1}, M_i, M_{i+1} solving the system

$$n_i(x_{i+1}-t_i)-n_{i+1}(x_{i+1}-t_{i+1}) = g_{i+1}-g_i- \tfrac{1}{2}M_i(x_{i+1}-t_i)^2+$$

(6)
$$+ M_{i+1}(x_{i+1}-t_{i+1})^2/2$$

$$n_i - n_{i+1} = -M_i(x_{i+1}-t_i) + M_{i+1}(x_{i+1}-t_{i+1})$$

Denoting $k_i = t_{i+1}-t_i$ we obtain for $i=0(1)n-1$

(7a) $$n_i k_i = g_{i+1}-g_i-\left[(x_{i+1}-t_i)(t_{i+1}-x_{i+1}+k_i)\, M_i - (t_{i+1}-x_{i+1})^2\, M_{i+1}\right]/2$$

(tb) $$n_{i+1} k_i = g_{i+1}-g_i+\left[(x_{i+1}-t_i)^2\, M_i + (t_{i+1}-x_{i+1})(k_i+x_{i+1}-t_i)\, M_{i+1}\right]/2 \; .$$

In the special case of $x_{i+1} = (t_i+t_{i+1})/2$ the relations (7) have a simple form

(8a) $$n_i = (g_{i+1}-g_i)/k_i - k_i(3M_i+M_{i+1})/8$$

(8b) $$n_{i+1} = (g_{i+1}-g_i)/k_i + k_i(M_i+3M_{i+1})/8$$

Substituting n_k from (7) to (5) we obtain for $S_2(x)$ the representation

(SM) $$S_2(x) = g_k+n_k(x-t_k) + M_k(x-t_k)^2/2$$

with n_k from (7a) for $k=0(1)n-1$ and n_k from (7b) for $k=n$. Conditions (7) assure the continuity of $S_2(x)$, $S_2'(x)$ at the knot x_{i+1} of the spline with the M_i given. We have two conditions for every value M_i, $i=1(1)n-1$, because the value M_i appears in the continuity conditions for the knots x_i, x_{i+1}. Comparing these relations we obtain for $i=1(1)n-1$

$$k_i\left\{g_i-g_{i-1} +\left[(x_i-t_{i-1})^2 M_{i-1} + (t_i-x_i)(k_i+x_i-t_{i-1})M_i\right]/2\right\} =$$

$$= k_{i-1}\left\{g_{i+1}-g_i-\left[(x_{i+1}-t_i)(t_{i+1}-x_{i+1}+k_i)M_i - (t_{i+1}-x_{i+1})^2\, M_{i+1}\right]/2\right.$$

Rearranging it, we obtain a system of n-1 equation for the n+1 unknown values of second derivative M_i, $i=0(1)n$:

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

with $a_i = \left[(x_i-t_{i-1})/k_{i-1}\right]^2 k_{i-1}/(k_{i-1}+k_i) > 0$

$$b_i = (t_i - x_i)(1 + (x_i - t_{i-1})/k_{i-1}) +$$
$$+ (x_{i+1} - t_i)(1 + (t_{i+1} - x_{i+1})/k_i)/(k_{i-1} + k_i)$$
$$c_i = \left[(t_{i+1} - x_{i+1})/k_i\right]^2 k_i/(k_{i-1} + k_i) > 0$$

(9)
$$f_i = 2\left[(g_{i+1} - g_i)/k_i - (g_i - g_{i-1})/k_{i-1}\right]/(k_{i-1} + k_i) .$$

In the special case of $x_{i+1} = (t_i + t_{i+1})/2$ the system (M_i) reads

(10)
$$(k_{i-1}/(k_{i-1}+k_i))M_{i-1} + 3M_i + (k_i/(k_{i-1}+k_i))M_{i+1} = 8g\left[t_{i-1}, t_i, t_{i+1}\right]$$

(divided difference on the right) , $i=1(1)n-1$.

We assure the continuity of $S_2'(x)$ at the knots x_i, $i=1(1)n$ computing the values $M_i = S_2''(t_i)$ from the system (M_i). Using (7) for the computation of $n_i = S_2'(x_i)$, we can now express $S_2(x)$ with the help of (SM). For M_i to be uniquelly determined, we need two another conditions.

3.2 Boundary conditions of a general type

(BCM)
$$b_o M_o + c_o M_1 = f_o$$
$$a_n M_{n-1} + b_n M_n = f_n$$

complete the system (M_i) to the system of n+1 equations for the n+1 unknown values of second derivative $M_i = S_2''(x_i)$, $i = 0(1)n$. In this form we can express such usual cases as the following ones:

a) Prescription of the values M_o, M_n - the case of

(11) $b_o = b_n = 1$, $c_o = a_n = 0$, $f_o = M_o$, $f_n = M_n$.

b) Prescription of the values of the first derivative -

$n_o = S_2'(t_o)$, $n_n = S_2'(t_n)$.

According to (7) we have

$$2k_o n_o = 2(g_1 - g_o) - (x_1 - t_o)(t_1 - x_1 + k_o)M_o - (t_1 - x_1)^2 M_1$$

$$2k_{n-1}n_n = 2(g_n-g_{n-1})+(x_n-t_{n-1})^2M_{n-1}+(t_n-x_n)(k_{n-1}+x_n-t_{n-1})M_n$$

Rearranging, we can write these relations in the form of (BCM) with the coefficients

(12)
$$b_0 = (1+(t_1-x_1)/k_0)(x_1-t_0)/(k_0+k_1)\ , \quad a_0 = c_n = 0$$
$$c_0 = ((t_1-x_1)/k_0)^2k_0/(k_0+k_1)$$
$$f_0 = 2((g_1-g_0)/k_0-n_0)/(k_0+k_1)$$
$$a_n = ((x_n-t_{n-1})/k_{n-1})^2k_{n-1}/(k_{n-1}+k_{n-2})$$
$$b_n = (1+(x_n-t_{n-1})/k_{n-1})(t_n-x_n)/k_{n-2}+k_{n-1})$$
$$f_n = 2((g_n-g_{n-1})/k_{n-1}-n_n)/(k_{n-2}+k_{n-1}).$$

c) The approximation of the first or second derivative on the boundaries using boundary values of g_i and a suitable formula of numerical differentiation. Doing in such a way, we can follow then the cases a) or b).

The continuity conditions (M_i) together with the boundary conditions (BCM) form the system of n+1 equations for the n+1 unknown values M_i, i=0(1)n

(M)
$$\begin{bmatrix} b_0 & c_0 & & & \\ a_1 & b_1 & c_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \cdot \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \cdot \\ f_{n-1} \\ f_n \end{bmatrix}$$

with the coefficients a_i, b_i, c_i, f_i given by (9), (11) or (12). The strict diagonal dominancy condition for the matrix of this system looks now as

(13)
$$(k_{i-1}-d)(k_{i-1}-2d_i)/k_{i-1} < h_i+d_{i+1}(k_i-2d_{i+1})/k_i\ ,$$
$$|b_0|>|c_0|,\ |b_n|>|a_n|;$$

it is fulfilled in the special case of $x_{i+1} = (t_i+t_{i+1})/2$ and is sufficient for the existence of the unique spline $S_2(x)$ with any data (g_i) given.

3.3 The periodicity conditions for $S_2(x)$ are

$$\text{(PM)}\quad \begin{aligned} &g_n = g_o \text{ (it must be satisfied by the data given)}\\ &n_n = n_o\\ &M_n = M_o \text{ (this reduces the number of unknowns in } (M_i)). \end{aligned}$$

Using (7), the first and third periodicity condition, then the second condition may be expressed as

$$k_{n-1}\left\{(g_1-g_o)-\left[(x_1-t_o)(t_1-x_1+k_o)M_o-(t_1-x_1)^2M_1\right]/2\right\} =$$

$$= k_o\left\{g_o-g_{n-1}+\left[(x_n-t_{n-1})^2M_{n-1} + (t_n-x_n)(k_{n-1}+x_n-t_{n-1})M_o\right]/2\right\}$$

or finally, with $K = k_o + k_{n+1}$

$$\text{(14)}\quad \begin{aligned} &\left[(1-(d_1/k_o)^2)k_o/K + (2-d_n/k_{n-1})d_n/K\right]M_o + (d_1/k_o)^2(k_o/K)M_1 +\\ &+ \left[(1-d_n/k_{n-1})^2k_{n-1}/K\right]M_{n-1} = 2\left[(g_o-g_{n-1})/k_{n-1}-(g_1-g_o)/k_o\right]/K. \end{aligned}$$

The last equation in (M_i) reads under that conditions

$$\text{(15)}\quad \begin{aligned} &(d_n/k_{n-1})^2(k_{n-1}/D)M_o+\left[(k_{n-2}-d_{n-1})/k_{n-1}\right]^2(k_{n-2}/D)M_{n-2} +\\ &+ \left\{\left[d_{n-1}(1+(k_{n-2}-d_{n-1})/k_{n-2})+(k_{n-1}-d_n)(1+d_n/k_{n-1})\right]/D\right\}M_{n-1} =\\ &= 2\left[(g_n-g_{n-1})/k_{n-1}-(g_{n-1}-g_{n-2})/k_{n-2}\right]/D \;;\; D = k_{n-1}+k_{n-2}\,. \end{aligned}$$

Collecting (M_i), $i=1(1)n-2$, (14) and (15), we have the system of the n equations for n unknown values M_i, $i=0(1)n-1$

$$\text{(Mp)}\quad \begin{bmatrix} b_o & c_o & & a_o\\ a_1 & b_1 & c_1 & \\ & \ddots & \ddots & \ddots\\ & a_{n-2} & b_{n-2} & c_{n-2}\\ c_{n-1} & & a_{n-1} & b_{n-1} \end{bmatrix} \begin{bmatrix} M_o\\ M_1\\ \vdots\\ M_{n-2}\\ M_{n-1} \end{bmatrix} = \begin{bmatrix} f_o\\ f_1\\ \vdots\\ f_{n-2}\\ f_{n-1} \end{bmatrix}$$

with

a_o, b_o, c_o, f_o given in (14),

a_i, b_i, c_i, f_i given by (9) for $i \neq 0$, n-1

a_{n-1}, b_{n-1}, c_{n-1}, f_{n-1} given in (15).

The matrix of this system is cyclic tridiagonal, ; under the conditions (13) with i=1(1)n-1 and

$$b_o > a_o + c_o \ , \qquad b_{n-1} > a_{n-1} + c_{n-1}$$

it is strictly diagonally dominant - we have the unique periodic spline $S_2(x)$ with any data (g_i) given.

4. Algorithm and implementation

4.1 Using the results stated above we can describe the algorithm for computing parameters of the parabolic spline from the given data in the following stages:

1^o Choose the type of the representation - the first or second derivative (m, M -algorithm).

2^o Read the data determining the sets (Δx), (Δt) and control the right ordering, resp. conditions (4) or (12).

3^o Read the data (g_i), choose the type of boundary conditions (under periodicity condition control $g_0 = g_n$); read the data from boundary conditions.

4^o Compute the coefficients a_i, b_i, c_i, f_i of the system (m) or (M).

5^o Solve the system of linear equations (using special algorithms for systems with tridiagonal or cyclic tridiagonal matrix); on the output we have then all parameters needed for determining the spline $S_2(x)$ on any interval of our partition of $\langle a,b \rangle$ given by (Δx), (Δt).

6^o We can compute the values of $S_2(x)$ for any $x \in \langle a,b \rangle$ using the representation chosen, draw a picture of our spline, or make other operation with the spline computed.

4.2 The algorithm with the first derivatives and boundary conditions mentioned has been implemented on the PMD-85 personal computer for educational purposes. Varying the knots, the boundary conditions and the data (g_i), it is possible to follow on the screen the dependency of the spline on the changes of these parameters. Listing of our programm is given below (in BASIC-G):

```
-
10 REM PROGRAM PBSG1 - PARABOLICKY SPLAJN-
20 REM S OBECNOU SITI V JEDNE DIMENZI-
50 DIM A(20),B(20),C(20),D(20),H(20),F(20)-
60 DIM X(20),T(20),Y(20),AL(20),BL(20),GL(20)-
70 DIM VU(20),WU(20),W(20),XB(20)-
100 REM ZAPIS POTREBNE DIMENZE POLI-
101 GCLEAR-
105 PRINT "PARABOLICKY SPLAJN", PRINT-
110 PRINT "PROGRAM POCITA KOEFICIENTY M(I)-REPREZENTACE"-
115 PRINT "FUNKCNI HODNOTY V ZADANYCH BODECH"-
120 PRINT "KRESLI CELKOVY PRUBEH PARABOLICKEHO SPLAJNU"-
125 PRINT "URCENEHO NASLEDUJICIMI DATY", PRINT-
130 PRINT "SITI UZLU SPLAJNU X(I), I=0(1)N+1"-
135 PRINT "SITI UZLU INTERPOLACE T(I), I=0(1)N"-
140 PRINT "S PREDEPSANYMI HODNOTAMI Y(I), I=0(1)N"-
145 PRINT "PREDPOKLADA SE USPORADANI UZLU"-
150 PRINT "X(0)=T(0)<X(1)<T(1)<X(2)<...<X(N)<T(N)=X(N+1)"-
151 PRINT:PRINT "UPRAVTE DIMENZE POLI PODLE POCTU BODU", PRINT-
155 PRINT-
160 PRINT "PROGRAM UMOZNUJE ZPRACOVAT NAJEDNOU"-
165 PRINT "VICENASOBNE HODNOTY Y(I) VE STEJNYCH UZLECH"-
170 PRINT "POTREBNA DATA N,X,T,Y,Y...ULOZTE DO RADKU 4000"-
171 DISP "MATE ZADANA DATA?  [A/N]-
172 INPUT A$: IF A$="N"THEN STOP-
180 PRINT:PRINT "DALE JE TREBA ZVOLIT VHODNY TYP"-
185 PRINT "OKRAJOVYCH PODMINEK-MATE TENTO VYBER":PRINT-
190 PRINT "BC1 B(0)*M(0)+C(0)*M(1)=F(0)"-
195 PRINT "    A(N+1)*M(N)+B(N+1)*M(N+1)=F(N+1)"-
200 PRINT-
205 PRINT "BC2 NEMATE VHODNE OKRAJOVE PODMINKY -"-
210 PRINT "    PROGRAM SI JE ZVOLI PODLE DAT Y(1)"-
211 PRINT-
215 PRINT "BC3 CHCETE PRO SPLAJN PRIMO PREDEPSAT"-
220 PRINT "    HODNOTY PRVE DERIVACE NA OKRAJICH"-
221 PRINT-
225 PRINT "BC4 CHCETE PRO SPLAJN PREDEPSAT HODNOTY"-
230 PRINT "    DRUHE DERIVACE NA OKRAJICH", PRINT-
231 PRINT "PER SPLAJN MA BYT PERIODICKY":PRINT-
235 PRINT "VYPISTE VASI VOLBU OKRAJOVYCH PODMINEK"-
236 PRINT "PODLE OZNACENI NA LEVEM OKRAJI RADKU":PRINT-
240 INPUT P$ -
250 READ N-
255 FOR I=0 TO N+1-
260 READ X(I): NEXT I-
262 FOR I=0 TO N: H(I)=X(I+1)-X(I): NEXT I-
265 FOR I=0 TO N: READ T(I):NEXT I-
```

```
272 FOR I=0 TO N: D(I)=T(I)-X(I): NEXT I-
275 IF NOT (X(0)=T(0) AND X(N+1)=T(N)) GOTO 2540
280 FOR I=1 TO N-
285 IF NOT (T(I-1)<X(I) AND X(I)<T(I)) GOTO 295-
290 NEXT I-
300 IF P ="BC1" GOTO 330-
305 IF P ="BC2" GOTO 350-
310 IF P ="BC3" GOTO 375-
315 IF P ="BC4" GOTO 400-
320 IF P ="PER" GOTO 430-
330 PRINT:PRINT "ZADEJ B(0),C(0),F(0),A(N+1),B(N+1),F(N+1)"-
335 INPUT B(1),C(1),F(1),A(N+2),B(N+2),F(N+2)-
336 A(1)=0: C(N+2)=0-
340 PRINT: GOTO 500-
345 PRINT-
350 PRINT "SPOCITAM DERIVACE ZE DVOU HODNOT Y(I)"-
355 B(1)=1: C(1)=0 : A(1)=0-
356 A(N+2)=0: B(N+2)=1: C(N+2)=0-
371 GOTO 500-
375 PRINT "ZADEJ HODNOTY PRVE DERIVACE NA OKRAJICH"-
380 INPUT F(1),F(N+2)-
385 B(1)=1: C(1)=0:A(1)=0-
390 A(N+2)=0:B(N+2)=1:C(N+2)=0-
396 GOTO 500-
400 PRINT "ZADEJ DRUHE DERIVACE NA OKRAJICH"-
405 INPUT SL,SP-
410 A(1)=0:B(1)=1:C(1)=-1:F(1)=-(X(1)-X(0))*SL-
415 A(N+2)=-1:B(N+2)=1:C(N+2)=0:F(N+2)=(X(N+1)-X(N))*SP-
425 GOTO 500-
430 REM PERIODICKY SPLAJN-
431 IF Y(0)<>Y(N) THEN PRINT "Y(0)<>Y(N)":STOP-
435 HP=H(0): HM=H(N):A(N+1)=-1-
440 B(N+1)=1+HM/HP: C(N+1)=-HM/HP-
465 FOR I=1 TO N-
470 HM=H(I-1): HP=H(I): DM=D(I-1): DP=D(I)-
475 A(I)=(HM-DM)*(HM-DM): C(I)=DP*DP*HM/HP-
480 B(I)=DP*(2*HP-DP)*HM/HP+(HM-DM)*(HM+DM)-
490 NEXT I-
492 N1=N-
495 GOTO 1145 : REM K ROZKLADU MATICE-
500 REM VYPOCET KOEFICIENTU MATICE PRO BC1_4-
505 FOR I=1 TO N-
510 HM=H(I-1): HP=H(I): DM=D(I-1): DP=D(I)-
515 A(I+1)=(HM-DM)*(HM-DM): C(I+1)=DP*DP*HM/HP-
520 B(I+1)=DP*(2*HP-DP)*HM/HP+(HM-DM)*(HM+DM)-
530 NEXT I-
532 N1=N+1-
535 REM DOKONCENA TVORBA MATICE - K ROZKLADU-
1145 REM ZACATEK ROZKLADU MATICE-
1150 BL(1)=B(1): GL(1)=0: VU(1)=0: WU(1)=0-
1160 GL(2)=C(N1+1): Z=1/B(1): WU(2)=Z*A(1)-
1170 VU(2)=Z*C(1)-
1180 FOR K=2 TO N1-1-
1190 P=A(K): BL(K)=B(K)-P*VU(K): Z=1/BL(K)-
1200 VU(K+1)=C(K)*Z: GL(K+1)=-VU(K)*GL(K)-
1210 WU(K+1)=-P WU(K)*Z-
```

```
1220 NEXT K-
1230 BL(N1)=B(N1)-A(N1)*VU(N1): AL(1)=0-
1240 FOR K=2 TO N1: AL(K)=A(K): NEXT K-
1250 AL(N1+1)=A(N1+1)-GL(N1)*VU(N1)-
1260 VU(N1+1)=(C(N1)-A(N1)*WU(N1))/BL(N1)-
1270 S=B(N1+1)-
1280 FOR I=2 TO N1: S=S-GL(I)*WU(I): NEXT I-
1290 BL(N1+1)=S-AL(N1+1)*VU(N1+1)-
1300 GL(N1+1)=0: WU(N1+1)=0-
1310 REM KONEC ROZKLADU-
1400 REM CTENI DAT Y(I)-
1405 FOR I=0 TO N: READ Y(I): NEXT I-
1410 REM VYPOCET PRAVYCH STRAN SOUSTAVY-
1415 IF P ="PER" GOTO 1440-
1420 FOR I=1 TO N-
1425 F(I+1)=2*H(I-1)*(Y(I)-Y(I-1))-
1430 NEXT I-
1435 GOTO 1460-
1440 FOR I=1 TO N-
1445 F(I)=2*H(I-1)*(Y(I)-Y(I-1))-
1450 NEXT I-
1452 F(N+1)=0-
1455 GOTO 1520-
1460 IF NOT (P$="BC2") GOTO 1520-
1470 F(1)=(Y(1)-Y(0))/(T(1)-T(0))-
1475 F(N+2)=(Y(N)-Y(N-1))/(T(N)-T(N-1))-
1520 REM RESENI SOUSTAVY A*M=F-
1540 W(1)=F(1)/BL(1)-
1550 FOR K=2 TO N1-
1560 W(K)=(F(K)-AL(K)*W(K-1))/BL(K)-
1570 NEXT K-
1580 S=F(N1+1)-
1590 FOR I=1 TO N1-1-
1600 S=S-GL(I+1)*W(I)-
1610 NEXT I-
1620 W(N1+1)=(S-AL(N1+1)*W(N1))/BL(N1+1)-
1630 M(N1+1)=W(N1+1)-
1640 M(N1)=W(N1)-VU(N1+1)*M(N1+1): Z=M(N1+1)-
1650 FOR K=N1-1 TO 1 STEP -1-
1660 M(K)=W(K)-VU(K+1)*M(K+1)-WU(K+1)*Z-
1670 NEXT K-
1680 REM KONEC VYPOCTU JEDNE SOUSTAVY-
1682 IF P ="PER" THEN M(0)=M(N+1): GOTO 1700-
1685 FOR I=1 TO N+2: M(I-1)=M(I): NEXT I-
1690 REM OZNACENI UPRAVENO NA OBVYKLE-
1700 GCLEAR-
1710 PRINT "HODNOTY DERIVACI V UZLECH? [A/N]"-
1720 INPUT A$: IF A$="N" GOTO 1800-
1730 PRINT: PRINT "I - X(I) - M(I)": PRINT-
1740 FOR I=0 TO N+1-
1750 PRINT I, X(I), M(I)-
1760 NEXT I-
1770 PRINT-
1800 PRINT "HODNOTY SPLAJNU V NEKTERYCH BODECH? [A/N]"-
1805 INPUT A$: IF A$="N" GOTO 2000-
1810 PRINT "ZADEJ POCET BODU": INPUT NB-
```

```
1815 PRINT "ZADEJ JEJICH SOURADNICE X"-
1820 FOR I=1 TO NB: INPUT XB(I): NEXT I-
1822 PRINT:PRINT "BOD - HODNOTA SPLAJNU": PRINT-
1825 FOR I=1 TO NB-
1830 XP=XB(I)-
1835 IF(XP<X(O) OR XP>X(N+1)) THEN PRINT "MIMO INTERVAL"-
1840 FOR J=O TO N-
1845 IF XP<X(J+1) GOTO 1855-
1850 NEXT J-
1855 Q=XP-X(J): DJ=D(J): MJ=M(J)-
1890 SX=Y(I)+(Q-DJ)*(MJ+.5*(M(J+1)-MJ*(Q+DJ)/H(J))-
1895 PRINT XP, SX-
1900 NEXT I-
2000 PRINT-
2010 PRINT "GRAFICKE ZNAZORNENI SPLAJNU?  [A/N]"-
2020 INPUT A$: IF A$="N" GOTO 2500-
2030 REM ROZMERY OBRAZU-
2040 YL=Y(O): YU=Y(O)-
2050 FOR I=1 TO N-
2060 IF Y(I)<YL THEN YL=Y(I)-
2070 IF Y(I)>YU THEN YU=Y(I)-
2080 NEXT I-
2090 RX=(X(N+1)-X(O))/16:RY=(YU-YL)/8-
2100 GCLEAR-
2110 SCALE X(O)-RX,X(N+1)+RX,YL-RY,YU+RY-
2120 FOR I=O TO N-
2130 MOVE T(I),Y(I)-
2140 LABEL 1,1; "*"-
2150 NEXT I-
2160 MOVE X(O),Y(O)-
2170 FOR I=O TO N-
2180 YI=Y(I): MI=M(I):KI=.5*(M(I+1)-MI)/H(I)-
2190 X1=X(I+1):DI=D(I): XI=X(I)-
2200 HH=(X1-XI)/20: IF RX>HH THEN RX=HH-
2210 FOR XP=XI TO X1 STEP RX-
2220 Q=XP-XI-
2230 SX=YI+(Q-DI)*(MI+KI*(Q+DI))-
2240 PLOT XP,SX-
2250 NEXT XP-
2260 MOVE X1,SX-
2270 LABEL 1,1;"+"-
2280 MOVE X1,SX-
2290 NEXT I-
2500 DISP "DALSI DATA?  [A,N]"-
2510 INPUT A$: IF A$="N" THEN STOP-
2520 GOTO 1400-
2530 END-
2540 PRINT: PRINT "CHYBA V ZADANI SITE": END
4000 DATA
```

The programm allows us

- to choose one of the five types of boundary conditions (BC1-BC4, PER - see lines 190-240)
- to print the values of derivatives m_i at the knots of the spline (see lines 1710-1760)

- to compute and print the values of the spline at the points choosen by the user (lines 1800-1900)
- to draw a graph of the spline on the interval (x_o, x_{n+1})
- to repeat the computation with the previous sets of knots and boundary conditions and another set of data (g_i) (denoted as y_i in the programm - see 2500-2520).

REFERENCES

[1] A h l b e r g, J.H. - N i l s o n, E.N. - W a l s h, J.L.: The theory of splines and its applications (russian transl. Moscow 1972)

[2] de B o o r, C.: The practical guide to splines. Springer, N.Y. 1978

[3] K o b z a, J.: An algorithm for biparabolic spline. Aplikace matematiky 32 (1987), 5, 401-413.

[4] M a k a r o v, V.L. - Ch l o b y s t o v, V.V.: Splajn approksimacija funkcij. Moskva, 1983

[5] S ch u l t z, M.H.: Spline analysis. Prentice-Hall, N.J. 1973

[6] S t ě č k i n, S.B. - S u b b o t i n, J.N.: Splajny v vyčislitelnoj matematike. Moskva 1976

[7] Z a v j a l o v, J.S. - K v a s o v, B.I. - M i r o š n i - č e n k o, V.L.: Metody splajn -funkcij. Moskva, Nauka 1980

SOUHRN

O algoritmech pro výpočet parabolických splajnů

J i ř í K o b z a

V práci jsou studovány algoritmy pro výpočet parametrů parabolického interpolujícího splajnu, zadaného pomocí sítě uzlů splajnu (x_i), uzlů (bodů) interpolace (t_i), předepsaných hodnot

(g_i) v uzlech interpolace a okrajových podmínek různých typů. Pro případ obecné sítě jsou podrobně prostudovány algoritmy užívající k reprezentaci splajnu prvé derivace v uzlech splajnu (m-algoritmus), resp. druhé derivace v uzlech interpolace (M--algoritmus). Jsou uvedeny postačující podmínky pro existenci a jednoznačnost splajnu. V závěru je uvedena implementace M-algoritmu ve formě programu pro osobní počítač PMD-85 v jazyku BASIC G.

РЕЗЮМЕ

Алгоритм для параболических сплайнов

И . Кобза

В работе исследуются алгоритмы для вычисления параметров интерполяционного параболического сплайна, определенного при помощи сетки узлов сплайна (x_i), точек интерполяции (t_i), предписанных значений (g_i) в точках интерполяции и краевых условий разных типов. Для общей сетки подробно изучены алгоритмы, которые для представления сплайна используют первые производные сплайна в узлах сплайна или вторые производные в узлах интерполяции. Приведены достаточные условия существования и единственности сплайна. Приведена программа на языке BASIC G, в которой используются первые производные.