KYBERNETIKA — VOLUME 13 (1977), NUMBER 4

An Algorithm for the Computation of Polynomial Splines of Odd Degree

VÍTĚZSLAV VESELÝ

On computers with small memory size it may be a serious problem to compute interpolation polynomial splines as far as the degree is required to be greater than three. Here an algorithm for splines of odd degree is described that considerably reduces the number of linear equations to be solved. Also a simple FORTRAN subroutine is given which enables a direct application of this method.

INTRODUCTION

The polynomial splines are continuous piecewise polynomial functions with many applications to problems in numerical analysis. Especially they prove to be an effective tool in the elementary processes of interpolation. Because of their easy computation the cubic splines (polynomial components are of degree 3) are mostly used for this purpose (see chapter II. in [1]). In this case a special procedure is used leading to the solution of a system of linear equations with a tridiagonal matrix. The number of equations and unknowns is equal to that of the points of interpolation. For higher-degree splines this approach becomes considerably more complex and practically useful only for uniformly spaced mesh points.

To solve the problem in general it is possible to write down in a straightforward manner a system of $(m + 1) N$ equations in $(m + 1) N$ unknowns where N is the number of mesh intervals and m the degree of spline. For large m and N one encounters the problem how to solve such a large system on a computer with small memory size.

The aim of this contribution is to describe a special procedure for splines of odd degree that enables to reduce the number of linear equations to be solved to the number of prescribed initial conditions. As we usually prescribe the values of the function and some of its derivatives at mesh points, this number is not much greater than N. The simple Fortran subroutine presented at the end of the article enables not only to generate this reduced system but also to compute the coefficients of the polynomials by means of which the searched spline is pieced together.

The restriction on the odd degree is necessary because there is an essential difference between splines of even and odd degree (see [1]). One finds, for example, that polynomial splines of even degree interpolating to a prescribed function need not exist. But for practical use this restriction does not seem to be substantial.

NOTATION

In further considerations $S_\Delta(x)$ denotes a spline function of degree $2n - 1$ $(n \geqq 1)$ defined on a mesh $\Delta : a = x_0 < x_1 < \ldots < x_N = b$. Let $h_i = x_i - x_{i-1}$ for $i = 1, 2, \ldots, N$. By definition $S_\Delta(x)$ coincides on each mesh interval $[x_{i-1}, x_i]$ with some polynomial P_i of degree $2n - 1$. So

$$S_\Delta(x)|_{[x_{i-1},x_i]} = P_i(x - x_{i-1}) =$$

$$= c_{i1} + c_{i2}\frac{x - x_{i-1}}{1!} + c_{i3}\frac{(x - x_{i-1})^2}{2!} + \ldots + c_{i,2n}\frac{(x - x_{i-1})^{2n-1}}{(2n-1)!}.$$

If we denote

$$u_j(t) \begin{cases} = t^j/j! & \text{for } j = 0, 1, 2, \ldots, \\ = 0 & \text{for } j < 0, \end{cases}$$

we can write

$$S_\Delta(x)|_{[x_{i-1},x_i]} = P_i(x - x_{i-1}) = c_{i1} + c_{i2}\, u_1(x - x_{i-1}) + \ldots \\ \ldots + c_{i,2n}\, u_{2n-1}(x - x_{i-1}). \tag{1}$$

For derivatives of order $p = 0, 1, 2, \ldots$, it holds

$$u_j^{(p)}(t) = u_{j-p}(t)$$

and therefore

$$P_i^{(p)}(x - x_{i-1}) = c_{i,p+1} + c_{i,p+2}\, u_1(x - x_{i-1}) + \ldots \\ \ldots + c_{i,2n}\, u_{2n-p-1}(x - x_{i-1}). \tag{2}$$

Definition. We call $S_\Delta(x)$ a polynomial spline with *deficiency* k $(1 \leqq k \leqq n)$ if it has continuous derivatives of orders $0, 1, \ldots, 2n - k - 1$ at interior mesh points.

Further k is always used in this sense.

The following matrices will be useful in later considerations:

$$\mathbf{U}(t) = \begin{pmatrix} 1 & u_1(t) & u_2(t) & \ldots & u_{2n-k-1}(t) \\ 0 & 1 & u_1(t) & \ldots & u_{2n-k-2}(t) \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix} \tag{3}$$

 of type $2n - k/2n - k$,

(4)
$$\bar{\mathbf{U}}(t) = \begin{pmatrix} u_{2n-k}(t) & u_{2n-k+1}(t) & \dots & u_{2n-1}(t) \\ u_{2n-k-1}(t) & u_{2n-k}(t) & \dots & u_{2n-2}(t) \\ \vdots & \vdots & & \vdots \\ u_1(t) & u_2(t) & \dots & u_k(t) \end{pmatrix}$$

of type $2n - k/k$,

(5)
$$\mathbf{E}_q = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

unit matrix of type q/q.

(6) $\mathbf{c} = (c_{11}, \dots, c_{1,2n}, c_{21}, \dots, c_{2,2n}, \dots, c_{N1}, \dots, c_{N,2n})$

(7) $\mathbf{c}_k = (c_{11}, \dots, c_{1,2n-k}, c_{21}, \dots, c_{2,2n-k}, \dots, c_{N-1,1}, \dots, c_{N-1,2n-k})$

(8) $\mathbf{d} = (d_{11}, \dots, d_{1k}, d_{21}, \dots, d_{2k}, \dots, d_{N-1,1}, \dots, d_{N-1,k}, d_{N1}, \dots, d_{N,2n})$

where

$$d_{ij} = c_{i,2n-k+j} \quad \text{for} \quad i = 1, 2, \dots, N-1 \quad \text{and} \quad j = 1, \dots, k$$
$$d_{Nj} = c_{Nj} \quad \text{for} \quad j = 1, \dots, 2n\,.$$

CONTINUITY CONDITIONS AT INTERIOR MESH POINTS ($N \geqq 2$)

Requiring derivatives of orders $0, 1, \dots, 2n - k - 1$ to be continuous at each interior mesh point we get $(N - 1)(2n - k)$ conditions

(9)
$$P_i^{(p)}(h_i) = P_{i+1}^{(p)}(0) \quad \text{for} \quad i = 1, 2 \dots, N-1 \quad \text{and}$$
$$p = 0, 1, \dots, 2n - k - 1\,.$$

In view of (2) we can write them in the form

(10)
$$c_{i,p+1} + c_{i,p+2}\, u_1(h_i) + \dots + c_{i,2n}\, u_{2n-p-1}(h_i) = c_{i+1,p+1}\,.$$

Using matrix notation (3)–(6) these continuity conditions acquire the form

$$\begin{pmatrix} \mathbf{U}(h_1) & \bar{\mathbf{U}}(h_1) & -\mathbf{E}_{2n-k} & 0 & 0 & \dots & & & & 0 \\ 0 & 0 & \mathbf{U}(h_2) & \bar{\mathbf{U}}(h_2) & -\mathbf{E}_{2n-k} & \dots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & & \dots & \underbrace{\mathbf{U}(h_{N-1}) \quad \bar{\mathbf{U}}(h_{N-1})}_{2n} & & \underbrace{-\mathbf{E}_{2n-k}}_{2n-k} & \underbrace{0}_{k} \end{pmatrix} \mathbf{c}^T = 0\,.$$

Notation (7), (8) enables to rewrite this equation into another equivalent form:

$$\mathbf{A}\mathbf{c}_k^T = \mathbf{B}\mathbf{d}^T \tag{12}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{U}(h_1) & -\mathbf{E}_{2n-k} & 0 & \ldots & 0 & 0 \\ 0 & \mathbf{U}(h_2) & -\mathbf{E}_{2n-k} & & & \vdots \\ \vdots & \vdots & \vdots & & & \\ 0 & & & & \mathbf{U}(h_{N-2}) & -\mathbf{E}_{2n-k} \\ 0 & \ldots & & \ldots & 0 & \mathbf{U}(h_{N-1}) \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} -\bar{\mathbf{U}}(h_1) & 0 & \ldots & & & 0 & 0 \\ 0 & -\bar{\mathbf{U}}(h_2) & & & & \vdots & \vdots \\ \vdots & & & & & & \\ 0 & & & -\bar{\mathbf{U}}(h_{N-2}) & 0 & 0 & 0 \\ 0 & \ldots & \ldots & 0 & \underbrace{-\bar{\mathbf{U}}(h_{N-1})}_{2n-k} & \underbrace{\mathbf{E}_{2n-k}}_{2n-k} & \underbrace{0}_{k} \end{pmatrix}$$

In view of (3) we have $|\mathbf{A}| = 1$ which means that $\mathbf{A}$ is invertible.

Thus we have proved the following

Theorem. $S_\Delta(x)$ is a spline with deficiency k $(1 \leqq k \leqq n)$ if and only if it satisfies

$$\mathbf{c}_k^T = \mathbf{A}^{-1}\mathbf{B}\mathbf{d}^T\,. \tag{13}$$

INITIAL CONDITIONS

With respect to $(N - 1)(2n - k)$ continuity conditions for a spline of deficiency k there are still leaving $2nN - (N - 1)(2n - k) = k(N - 1) + 2n$ degrees of freedom. Therefore to determine $S_\Delta(x)$ completely it is necessary to add some $k(N - 1) + 2n$ initial conditions. One usually prescribes k conditions at interior mesh points (the values of derivatives of order $0, 1, \ldots, k - 1$) and n end conditions at $x = x_0$ and $x = x_N$.

The standard form of initial conditions (see (1), (2), too):

– at interior mesh points x_i $(i = 1, \ldots, N - 1)$ we prescribe the values of derivatives of order $p = 0, 1, \ldots, k - 1$

$$c_{i+1,p+1} = y_i^{(p)}\,; \tag{14}$$

– end conditions at x_0, x_N assume one of the following three forms

I. we prescribe the values of derivatives of order $p = 0, 1, \ldots, n - 1$

$$c_{1,p+1} = y_0^{(p)}$$

$$c_{N,p+1} + c_{N,p+2}\, u_1(h_N) + \ldots + c_{N,2n}\, u_{2n-p-1}(h_N) = y_N^{(p)}\,; \tag{15}$$

II. we prescribe the values of derivatives of order
$p = 0, n, n+1, \ldots, 2n-2$

(16) $$c_{1,p+1} = y_0^{(p)}$$

$$c_{N,p+1} + c_{N,p+2}\, u_1(h_N) + \ldots + c_{N,2n}\, u_{2n-p-1}(h_N) = y_N^{(p)}\,;$$

III. we prescribe periodicity of derivatives of order $p = 0, 1, \ldots, 2n-k-1$

(17) $$c_{1,p+1} = c_{N,p+1} + c_{N,p+2}\, u_1(h_N) + \ldots + c_{N,2n}\, u_{2n-p-1}(h_N)$$

and the values of derivatives of order $p = 0, 1, \ldots, k-1$

$$c_{1,p+1} = y_0^{(p)}(= y_N^{(p)})\,.$$

Theorem. ([1], Theorem 5.8.2). There exists a unique polynomial spline $S_\Delta(x)$ of degree $2n-1$ and with deficiency k $(1 \leqq k \leqq n)$ satisfying standard initial conditions of type I. (equations (14), (15)) or of type III. (equations (14), (17)) The same is true for initial conditions of type II. (equations (14), (16)) provided $k(N-1) \geqq n-2$ holds.

In general the initial conditions may assume any other form leading to $k(N-1) + 2n$ equations linear in c_{ij} supposing, of course, the corresponding spline exists.

The principal idea of the computation of a polynomial spline with given initial conditions and of prescribed degree and deficiency is based on the equation (13). The searched spline must satisfy (13) and the initial conditions at the same time. Substituting from (8) and (13) for c_{ij} in the initial conditions we obtain a system of $k(N-1) + 2n$ linear equations in $k(N-1) + 2n$ unknowns d_{ij}. We solve this system and compute the remaining c_{ij} using (13) once more.

Now we shall deal with the matrix $\mathbf{A}^{-1}\mathbf{B}$ in detail.

THE COMPUTATION OF THE m-th ROW OF $\mathbf{A}^{-1}\mathbf{B}$

Let us write m in the form $m = (\bar{\imath} - 1)(2n-k) + \bar{\jmath}$ where $\bar{\imath} \in \{1, \ldots, N-1\}$ and $\bar{\jmath} \in \{1, \ldots, 2n-k\}$. In (13) this row is corresponding to $c_{\bar{\imath}\bar{\jmath}}$.

First we compute the m-th row $\boldsymbol{a}^m$ of $\mathbf{A}^{-1}$.

Let us denote

$$\boldsymbol{a}^m = (a_{11}^m, \ldots, a_{1,2n-k}^m, a_{21}^m, \ldots, a_{2,2n-k}^m, \ldots, a_{N-1,1}^m, \ldots, a_{N-1,2n-k}^m)$$

and

$$\boldsymbol{a}_i^m = (a_{i1}^m, a_{i2}^m, \ldots, a_{i,2n-k}^m) \quad \text{for} \quad i = 1, 2, \ldots, N-1\,.$$

As $\mathbf{A}^{-1}\mathbf{A} = \mathbf{E}_{(N-1)(2n-k)}$, we have $\mathbf{A}^T(\mathbf{A}^{-1})^T = \mathbf{E}_{(N-1)(2n-k)}$. Thus we can compute $\boldsymbol{a}^m$ solving the system of linear equations

$$\mathbf{A}^T(\boldsymbol{a}^m)^T = (0, \ldots, 0, \underset{\substack{\uparrow \\ m\text{-th position}}}{1}, 0, \ldots, 0)^T$$

Considering the form of $\boldsymbol{A}^T$ we get

$$(18) \qquad \boldsymbol{a}_1^m = \boldsymbol{a}_2^m = \ldots = \boldsymbol{a}_{i-1}^m = 0\,,$$
$$a_{i1}^m = a_{i2}^m = \ldots = a_{i,j-1}^m = 0\,, \quad a_{ij}^m = 1\,.$$

For a_{ij}^m with $(i-1)(2n-k)+j > m$ we obtain the following recurrent formula supposing the $2n-k$ elements preceding a_{ij}^m are already known

$$(19) \qquad a_{ij}^m = a_{i-1,j}^m - \sum_{p=1}^{j-1} u_{j-p}(h_i)\, a_{i,p}^m\,.$$

In (19) the sum $\sum_{p=1}^{j-1} u_{j-p}(h_i)\, a_{i,p}^m$ is to be left out whenever $j = 1$. For $i = 1$ we set $a_{0,j}^m = 0$.

Considering the form of $\boldsymbol{B}$ we can write for the m-th row of $\boldsymbol{A}^{-1}\boldsymbol{B}$

$$(20) \qquad \boldsymbol{b}^m = (b_{11}^m, \ldots, b_{1k}^m, b_{21}^m, \ldots, b_{2k}^m, \ldots, b_{N-1,1}^m, \ldots, b_{N-1,k}^m, a_{N-1,1}^m, \ldots$$
$$\ldots, a_{N-1,2n-k}^m, \underbrace{0, \ldots, 0}_{k})$$

where

$$(b_{i1}^m, b_{i2}^m, \ldots, b_{ik}^m) = -(a_{i1}^m, a_{i2}^m, \ldots, a_{i,2n-k}^m)\, \overline{\boldsymbol{U}}(h_i)$$

for $i = 1, 2, \ldots, N-1$.

The following Fortran subroutine computes the rows of $\boldsymbol{A}^{-1}\boldsymbol{B}$ using (18), (19), (20).

```
   SUBROUTINE C(II,JJ,R)
   DIMENSION AM(2,21),R(561)
   COMMON N,MN,K,X(51),FAK(22)
   NM1 = N-1
   MNK = 2*MN-K
   J1 = JJ
   IP1 = 1
   IP2 = 2
   DO 1 I=1,MNK
     AM(1,I)=0.
1    AM(2,I)=0.
   AM(2,JJ)=1.
   IBEG=(II-1)*K
   DO 6 I=II,NM1
     DO 5 J=J1,MNK
     MNKJ=MNK-J
     HSI=(X(I+1)-X(I))**MNKJ
     AM(IP1,J)=AM(IP2,J)
     IF(J.EQ.J1)GOTO 3
```

```
      HI=1.
      JM1=J-1
      DO 2 IT=1,JM1
        JIT=J-IT
        HI=HI*(X(I+1)-X(I))
2       AM(IP1,J)=AM(IP1,J)-AM(IP1,JIT)*HI/FAK(IT+1)
3     DO 4 IS=1,K
        ICUR=IBEG+IS
        IFAK=MNKJ+IS+1
        HSI=HSI*(X(I+1)-X(I))
4       R(ICUR)=R(ICUR)-AM(IP1,J)*HSI/FAK(IFAK)
5     CONTINUE
      IBEG=IBEG+K
      J1=1
      JIT=IP1
      IP1=IP2
6     IP2=JIT
    DO 7 IS=1,MNK
      ICUR=IBEG+IS
7     R(ICUR)=AM(IP2,IS)
    RETURN
    END
```

Input parameters: $\bar{\imath}$ stored in II
$\bar{\jmath}$ stored in JJ

Output parameter: R containing the m-th row of $\boldsymbol{A}^{-1}\boldsymbol{B}$ where $m = (\bar{\imath} - 1) \cdot (2n - k) + \bar{\jmath}$. Zeros must be stored in R before calling C.

Common variables: N (number of mesh intervals)
MN (contains n where $2n - 1$ is the degree)
K (deficiency of spline)
X (contains mesh points $x_0, x_1, \ldots, x_N$)
FAK (contains $0!, 1!, \ldots, (2n - 1)!$)

Dimensions of X, AM, FAK, R are appointed with respect to the maximum values of degree $2n_{max} - 1$ and the number of mesh intervals N_{max} in this way:

$$\mathrm{X}(N_{max} + 1)\,, \quad \mathrm{AM}(2{,}2n_{max} - 1)\,, \quad \mathrm{FAK}(2n_{max})\,,$$
$$\mathrm{R}(n_{max}(N_{max} - 1) + 2n_{max})\,.$$

In the presented subroutine is $n_{max} = 11$, $N_{max} = 50$.

The subroutine was tested on TESLA 200 computer. As far as the computation time is concerned, for example, interpolation with a spline of degree 5, deficiency 1 with standard initial conditions of type I. on 30 mesh intervals takes approximately 5 minutes time.

This procedure was implemented also on HP 9820 A programmable desk calculator with 400 register memory size equipped with one cassette memory unit.

(Received January 14, 1977.)

REFERENCES

[1] J. H. Ahlberg, E. N. Nilson, J. L. Walsh: The Theory of Splines and Their Applications. Academic Press Inc., New York—London 1967.

RNDr. Vítězslav Veselý, Čejkova 74, 615 00 Brno. Czechoslovakia.