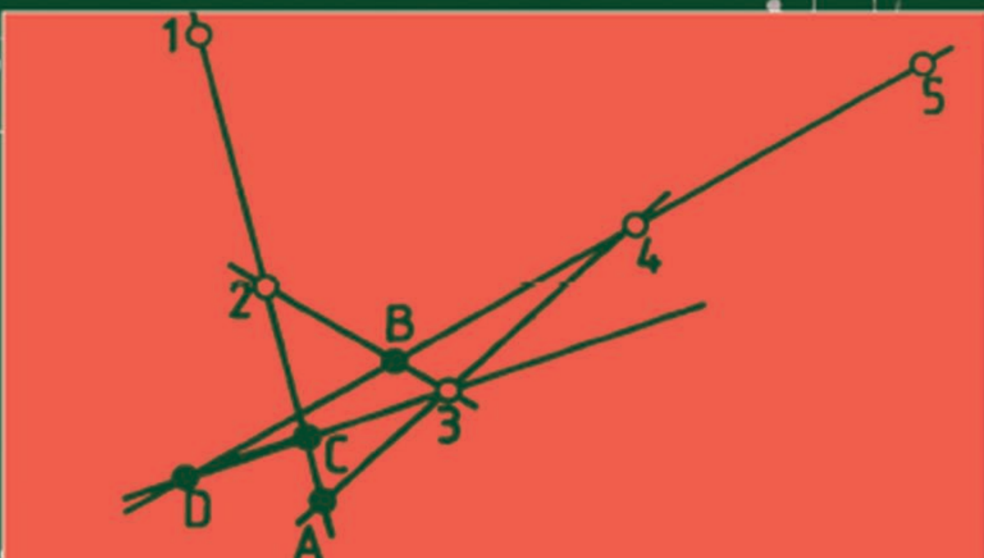
ONE DIMENSIONAL SPLINE INTERPOLATION ALGORITHMS



CRC Press
Taylor & Francis Group
AN A K PETERS BOOK

Helmuth Späth

One Dimensional Spline Interpolation Algorithms

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

One Dimensional Spline Interpolation Algorithms

Helmuth Späth
Universität Oldenburg
Oldenburg, Germany

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN A K PETERS BOOK

First published 1995 by A K Peters, Ltd.

Published 2018 by CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

ISBN-13: 978-1-56881-016-4 (hbk)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Library of Congress Cataloging-in-Publication Data

Späth, Helmuth.
[Eindimensionale Spline-Interpolations Algorithmen. English]
One dimensional spline interpolation algorithms / H. Späth.
p. cm,
Includes bibliographic references and index.
ISBN 1-56881-016-4
1. Spline theory--Data processing. 2. Functions--Data processing.
I. Title.
QA224.S59513 1995 95-40858
511'.42--dc20 CIP

Contents

Preface ix

1 Polynomial Interpolation 1

1.1. The Lagrange Form of the Interpolating Polynomial 1

1.2. The Newton Form of the Interpolating Polynomial 2

2 Polygonal Paths as Linear Spline Interpolants 9

2.1. General Spline Interpolants 9

2.2. Various Representations of a Polygonal Path 10

2.3. Evaluation by Searching an Ordered List 14

2.4. Properties of Polygonal Paths 14

2.5. When the Knots and Interpolation Nodes Are Different . . 16

2.6. Parametric Polygonal Paths 17

2.7. Smoothing with Polygonal Paths I 19

2.8. Smoothing with Polygonal Paths II 21

3 Quadratic Spline Interpolants 29

3.1. Knots the Same as Nodes 29

3.2. Optimal Initial Slope 30

3.3. Periodic Quadratic Spline Interpolants with Knots the Same as Nodes 39

3.4. Knots at the Midpoints of the Nodes 40

3.5. Knots Variable between the Nodes 45
3.6. Periodic Quadratic Spline Interpolants with Variable Knots between the Nodes 50
3.7. Nodes Variable between Knots 55
3.8. Quadratic Histosplines 55
3.9. Quadratic Hermite Spline Interpolants 61
3.10. Approximation of First Derivative Values I 65
3.11. Shape Preservation through the Choice of Additional Knots 71
3.12. Quadratic Splines with Given Slopes at Intermediate Points 77

4 Cubic Spline Interpolants **83**
4.1. First Derivatives as Unknowns 83
4.2. Second Derivatives as Unknowns 87
4.3. Periodic Cubic Spline Interpolants 96
4.4. Properties of Cubic Spline Interpolants 108
4.5. First Derivatives Prescribed 111
4.6. Second Derivatives Prescribed 116
4.7. Smoothing with Cubic Spline Functions 119
4.8. Smoothing with Periodic Cubic Spline Functions 125
4.9. Cubic X-Spline Interpolants 129
4.10. Discrete Cubic Spline Interpolants 133
4.11. Local Cubic Hermite Spline Interpolants 136
4.12. Approximation of Values for the First Derivative II 138

5 Polynomial Spline Interpolants of Degree Five and Higher **181**
5.1. Spline Interpolants of Degree Five 181
5.2. Quintic Hermite Spline Interpolants where First Derivative Values Are Specified 189
5.3. Periodic Quintic Hermite Spline Interpolants 201
5.4. Quintic Hermite Spline Interpolants where Second Derivatives also Are Specified 201
5.5. Quartic Histosplines 203
5.6. Higher-Degree Spline Interpolants 212

6 Rational Spline Interpolants **213**
6.1. Rational Splines with One Freely Varying Pole 213
6.2. Adaptive Rational Spline Interpolants 219
6.3. Rational Spline Interpolants with a Prescribable Pole 235
6.4. Rational Spline Interpolants with Two Prescribable Real Poles 249
6.5. Periodic Rational Spline Interpolants with Two Prescribable Real Poles 272

6.6. Rational Spline Interpolants with Two Prescribable Real or Complex Poles 277
6.7. Shape Preservation of Monotone Data 287
6.8. Shape Preservation of Convex or Concave Data 298
6.9. Rational Histosplines 305

7 Exponential Spline Interpolants **323**
7.1. First Derivatives as Unknowns 323
7.2. Second Derivatives as Unknowns 337
7.3. Periodic Exponential Spline Interpolants 340
7.4. Shape Preservation and Other Considerations 342

8 Spline Interpolation and Smoothing in the Plane **345**
8.1. Interpolation with Arbitrary Splines 345
8.2. Smoothing with Cubic Spline Interpolants 357

Postscript **365**

A Appendix **367**

B List of Subroutines **385**

Bibliography **389**

Index **403**

Preface

This is the result, after almost two decades of research, of bringing up-to-date and recapitulating (first for one dimension) my little book *Spline Algorithms for Curves and Surfaces*, of which almost five thousand copies in four editions have been sold, and which has also been translated into English.

Our intention, as it was previously, is to provide an elementary and directly applicable introduction to the computation of those (as simple as possible) spline functions, which are determined by the requirement of smooth and shape-preserving interpolation and (in two cases) the smoothing of measured or collected data.

By elementary, we mean in particular that we have chosen to give explicit and easily evaluated forms of the spline interpolants (instead of in terms of recursive B-splines) and that in general existence and uniqueness can be decided, since we can demonstrate strict diagonal dominance of tridiagonal and (in two cases) five-diagonal coefficient matrices of linear systems of equations in appropriate unknowns.

This book should also be useful for applications, since not only do we derive the formulas and algorithms as such, but we also give efficient Fortran-77 subroutines. These are used to calculate numerous examples that in turn allow the reader to assess how the various spline interpolants perform depending on the configuration of the data.

Since the earlier book, much is new, especially reasearch that has appeared in the literature in the last two decades. In this regard, local

Hermite quadratic and cubic C^1−splines should especially be mentioned. For the required purposes, these seem to be superior to other polynomial spline interpolants. Also, the numerous variants of simplest possible rational spline interpolants are especially emphasized.

Just as for my last book, *Mathematical Software for Linear Regression* (1987), the implementation of most, and the testing of all, the subroutines was carried out by Mr. Jörg Meier (Dipl. Math.), scientific assistant at the Department of Mathematics; without him this book would not have been possible. Students J. Haschen, R. Obst, and A. Stark contributed to the literature searches as well as to various preliminary studies. The non-trivial task of text preparation was carried out with care and patience by Mrs. Büsselmann, also of the department.

Oldenburg, May 1989 H. Späth

Preface to the English Edition

A number of typographical errors and small discrepancies have been corrected from the German edition. Most of these were discovered by Prof. Len Bos during the translation, which could not have been carried out in a more congenial manner. Many thanks! The handling of publication matters, in this case by Alice Peters, was very supportive and extremely reliable.

H. Späth

1

Polynomial Interpolation

1.1. The Lagrange Form of the Interpolating Polynomial

Suppose that we are given n points (x_k, y_k), $k = 1, \cdots, n$ with pairwise distinct x_k. Equivalently, by renumbering if necessary, we may assume that $x_1 < \cdots < x_n$. Then there is a unique polynomial, p_{n-1}, of degree $n - 1$, which interpolates this data. Indeed, the *Lagrange form* of p_{n-1} may be explicitly given by means of the *fundamental polynomials* or *cardinal functions*, L_i, defined by

$$L_i(x) = \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k}. \tag{1.1}$$

Some plotted examples of such L_i are given, for example, in [121], p. 83. Then, since $L_i(x_k) = \delta_{ik}$, we have

$$p_{n-1}(x) = \sum_{i=1}^{n} y_i L_i(x). \tag{1.2}$$

This representation requires $O(n^2)$ arithmetic operations for each evaluation of p_{n-1}. A more economical and also numerically more stable form can be obtained as follows. Set

$$\lambda_i = \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{1}{x_i - x_k}. \tag{1.3}$$

Notice that these factors are independent of the point of evaluation, v, and thus need only be computed once. Also, the special case of (1.2) with $y_i = 1$, $i = 1, \cdots, n$, yields the relation,

$$\sum_{i=1}^{n} L_i(x) = 1.$$

Together, these may be used to rewrite (1.2) in the so-called *barycentric representation*,

$$p_{n-1}(x) = \frac{\sum_{i=1}^{n} y_i \frac{\lambda_i}{x - x_i}}{\sum_{i=1}^{n} \frac{\lambda_i}{x - x_i}}, \tag{1.4}$$

of the Lagrange interpolating polynomial. This formula is well-defined for $x \neq x_i$ and may be extended continuously by setting $p_{n-1}(x_i) := y_i$, $i = 1, \cdots, n$. Using (1.4) requires only a further $O(n)$ operations per evaluation. For numerical reasons ([171]) it is good policy to renumber the *interpolation nodes* x_i so that

$$|\overline{x} - x_1| \geq |\overline{x} - x_2| \geq \cdots \geq |\overline{x} - x_n| \tag{1.5}$$

holds, where $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$. Since the values (1.3) are independent of the y_i, the barycentric representation is especially recommended when several polynomials with different y_i but the same nodes x_i are to be evaluated.

1.2. The Newton Form of the Interpolating Polynomial

If this is not the case, or if the intention is to add to the number of given points one by one, then the Newton form of the interpolating polynomial is preferred. We write

$$\begin{aligned} p_{n-1}(x) &= a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \\ &\quad \cdots + a_n(x - x_1)(x - x_2) \cdots (x - x_{n-1}), \end{aligned} \tag{1.6}$$

where the coefficients a_k are denoted by

$$a_k = f[x_1, x_2, \cdots, x_k]. \tag{1.7}$$

```
      SUBROUTINE NEWDIA(N,X,Y,A,IFLAG)
      DIMENSION X(N),Y(N),A(N)
      IFLAG=0
      IF (N.LT.1) THEN
         IFLAG=1
         RETURN
      END IF
      DO 10 I=1,N
         A(I)=Y(I)
10    CONTINUE
      DO 30 K=N,2,-1
         DO 20 I=K,N
            A(I)=(A(I)-A(I-1))/(X(I)-X(K-1))
20       CONTINUE
30    CONTINUE
      RETURN
      END
```

Calling sequence:

CALL NEWDIA(N,X,Y,A,IFLAG)

Purpose:
The determination of the coefficients of the Newton interpolating polynomial of degree N−1.

Description of the parameters:

N	Number of given points. N must be at least 1.	
X	ARRAY(N):	Upon calling must contain the abscissas x_k, $k = 1, \cdots, n$, with $x_i \neq x_j$ for $i \neq j$.
Y	ARRAY(N):	Upon calling must contain the ordinates y_k, $k = 1, \cdots, n$.
A	ARRAY(N):	Upon successful execution (IFLAG=0) contains the required polynomial coefficients.
IFLAG	=0:	Normal execution.
	=1:	N< 1 not permitted.

Remark: The difference scheme is worked out in a diagonal fashion.

Figure 1.1. Subroutine NEWDIA and its description.

```
      SUBROUTINE NEWSOL(N,X,A,T,F,IFLAG)
      DIMENSION X(N),A(N)
      IFLAG=0
      IF (N.LT.1) THEN
         IFLAG=1
         RETURN
      END IF
      F=A(N)
      DO 10 K=N-1,1,-1
         F=F*(T-X(K))+A(K)
10    CONTINUE
      RETURN
      END
```

Calling sequence:

CALL NEWSOL(N,X,A,T,F,IFLAG)

Purpose:
The calculation of the function value of the Newton interpolating polynomial at the point T.

Description of the parameters:

N	Number of given points. $N \geq 1$ is required.	
X	ARRAY(N):	Upon calling must contain the abscissas x_k, $k = 1, \cdots, n$, with $x_i \neq x_j$ for $i \neq j$.
A	ARRAY(N):	Upon calling must contain the polynomial coefficients $a_1, a_2, \cdots, a_n$.
T	Point at which the polynomial is to be evaluated.	
F	Value of the polynomial at the point T.	
IFLAG	=0:	Normal execution.
	=1:	N< 1 not permitted.

Figure 1.2. Subroutine NEWSOL and its description.

The *divided differences* $f[x_1, x_2, \cdots, x_n]$ may be calculated recursively from

$$f[x_1, x_2, \cdots, x_n] = \frac{f[x_2, \cdots, x_k] - f[x_1, \cdots, x_{k-1}]}{x_k - x_1}, \tag{1.8}$$

where $f[x_i] := y_i$, $i = 1, \cdots, n$. (A nice, expository derivation may be found, for example, in [8].) According to [164], it is recommended for numerical reasons that the x_i be renumbered so that

$$|v - x_1| \leq |v - x_2| \leq \cdots \leq |v - x_n|, \tag{1.9}$$

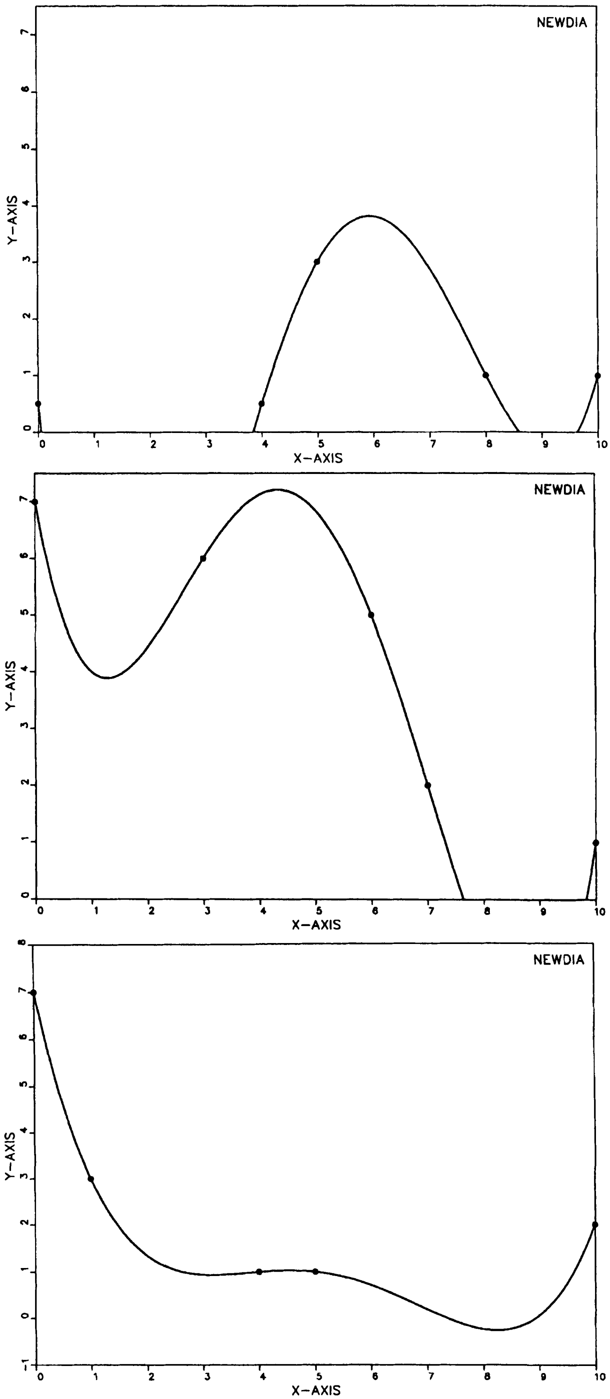


Figure 1.3. a-c.

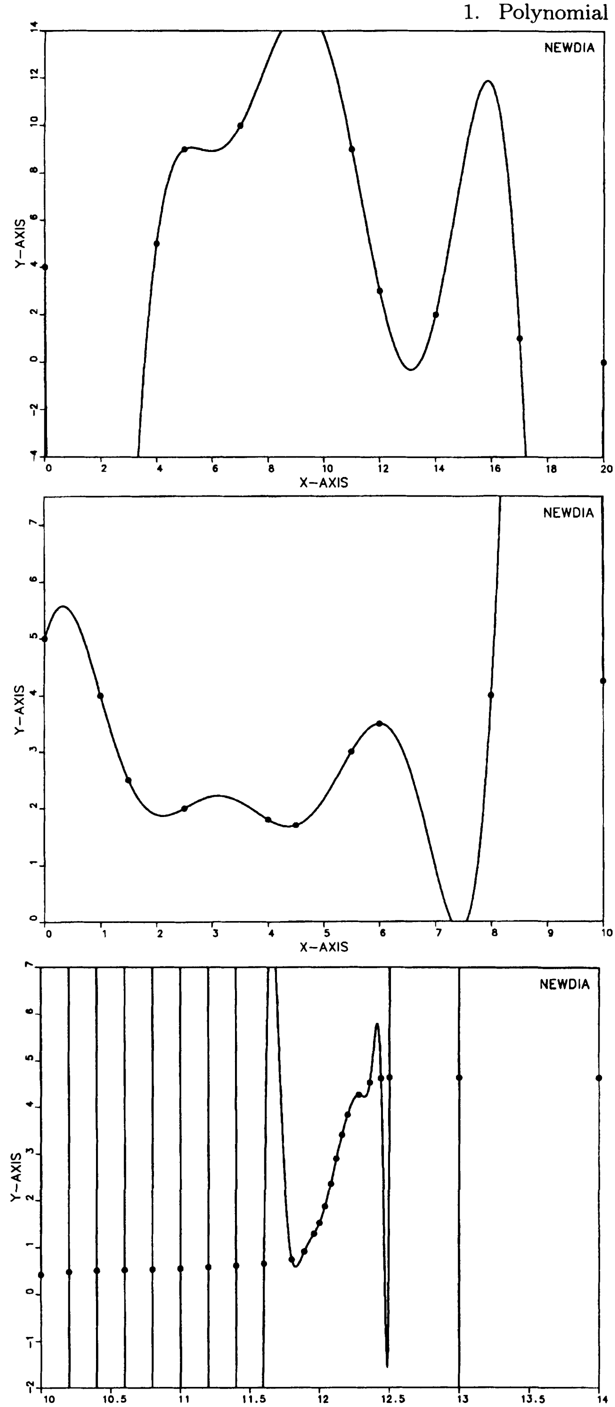

NEWDIA
Y-AXIS
X-AXIS
0 2 4 6 8 10 12 14 16 18 20
NEWDIA
Y-AXIS
X-AXIS
0 1 2 3 4 5 6 7 8 9 10
NEWDIA
Y-AXIS
X-AXIS
10 10.5 11 11.5 12 12.5 13 13.5 14

Figure 1.4. a-c.

where v is the point of evaluation. The Newton form of the interpolating polynomial may be efficiently evaluated by means of Horner's rule; i.e.,

$$\begin{aligned} p_{n-1}(x) &= a_1 + (x - x_1)(a_2 + (x - x_2)(a_3 + \cdots (x - x_{n-1})a_n) \cdots) \\ &= (\cdots((a_n(x - x_{n-1}) + a_{n-1})(x - x_{n-2}) + a_{n-2}) \cdots) \\ &\quad \times (x - x_1) + a_1. \end{aligned} \tag{1.10}$$

By this method, the Newton form without the renumbering is about half as expensive ([171]) as the Lagrange form without the renumbering (1.5).

Although the rearrangement of the given points for reasons of numerical stability, (1.5), is independent of the point of evaluation, while that for the Newton form, (1.9), is not, it is still in general preferable to use the representation (1.10) with the calculation of the coefficients by means of (1.8) (without renumbering). Therefore, we only give the subroutines NEWDIA (Fig. 1.1) for the calculation of divided differences and NEWSOL (Fig. 1.2) for the evaluation by a polynomial by Horner's rule, (1.10). NEWDIA uses (1.8) and the diagonal scheme of [121].

These routines do not involve the renumbering of (1.9). In general, this is not worthwhile, as interpolating polynomials of higher degree ($n \geq 4$), where it might be relevant, are not recommended for other reasons. This will be clear from the examples computed with NEWDIA and NEWSOL given in this section. In each of the first three, five points were given. These, together with the corresponding polynomial interpolants of fourth degree, are shown in Figs. 1.3a, b and c. This sequence of plots shows that polynomial interpolation preserves neither positivity nor monotonicity nor convexity of the data. In contrast, a simple *polygonal path* does possess these *shape-preserving* properties. The examples of Figs. 1.4a ($n = 9$) and b ($n = 10$) are taken from [154, pp. 31 and 105]. We will often encounter them later. Finally, the example of Fig. 1.4c ([121, p. 109]), involving 24 points, shows an interpolating polynomial of degree 23, which is completely ill-behaved. Although in special cases higher-degree polynomial interpolants can be useful, in general the results are such that this type of interpolation is not, in practice, applicable.

2

Polygonal Paths as Linear Spline Interpolants

2.1. General Spline Interpolants

For the given abscissas, we will now almost always suppose that

$$x_1 < x_2 < \cdots < x_n. \tag{2.1}$$

In general, by a spline interpolant $s \in C^m[x_1, x_n]$ with knots x_k, we mean a set of $n-1$ functions, s_k, defined on $[x_k, x_{k+1}]$, respectively, $k = 1, \cdots, n-1$, that are stitched together so as to be m-times ($m \geq 0$) continuously differentiable at the knots and that satisfy the interpolation conditions,

$$s_k(x_k) = y_k, \quad s_k(x_{k+1}) = y_{k+1}, \quad k = 1, \cdots, n-1. \tag{2.2}$$

For a polygonal path through the points (x_k, y_k), $k = 1, \ldots, n$, we have $m = 0$ and the s_k are all line segments with endpoints (x_i, y_i), $i = k, k+1$. For $m = 1$, we will be connecting parabolic segments, and for $m = 2$, cubic polynomials as well as other functions. As we shall see with polynomials of degree five and $m = 4$, $m > 2$ is in general unsuitable, since, as we saw in Chapter 1, the unacceptable properties of polynomials of higher degrees again take effect. One could, in principle, choose a different function type on each interval for s_k, but we avoid this for practical considerations.

2.2. Various Representations of a Polygonal Path

The line segments s_k can be represented in a number of ways. For example,

$$\begin{aligned} s_k(x) &= A_k + B_k x, && (2.3)\\ s_k(x) &= A_k + B_k(x - x_k), && (2.4)\\ s_k(x) &= A_k + B_k t, && (2.5)\\ s_k(x) &= A_k u + B_k t && (2.6) \end{aligned}$$

are all posssible. Here,

$$\begin{aligned} t &= \frac{x - x_k}{h_k}, \quad h_k = \Delta x_k = x_{k+1} - x_k,\\ u &= 1 - t = \frac{x_{k+1} - x}{h_k}. && (2.7) \end{aligned}$$

For each form, we may obtain the values of the corresponding $2(n-1)$ parameters A_k and B_k, $k = 1, \cdots, n-1$ from the interpolation conditions (2.2). The computational expense differs in each case. For the forms (2.4), (2.5), and (2.6), it follows immediately from $s_k(x_k) = y_k$ that $A_k = y_k$. For (2.4), it follows from $s_k(x_{k+1}) = y_{k+1}$ that the slope, B_k, of s_k is given by $B_k = \Delta y_k / \Delta x_k$.

The form (2.6) appears to be the most elegant, as then $B_k = y_{k+1}$ and thus,

$$s_k = y_k u + y_{k+1} t. \tag{2.8}$$

Hence, other than the given data (x_k, y_k), $k = 1, \cdots, n$, no new parameters are introduced and consequently no additional storage locations are required. (Note, however, that for (2.4), the y_k could be also be overwritten by the B_k.) The form (2.3) is unsuitable, as x does not vary intrinsically with respect to the interval $[x_k, x_{k+1}]$. In (2.4), $x - x_k$ varies from 0 to Δx_k, and in (2.5), t varies from 0 to 1 (standardized interval length).

Up till now, we have discussed the piecewise representation of the polygonal path, s. It is reasonable to ask if there is also a closed-form representation that holds on all of $[x_1, x_n]$? We introduce the notation,

$$(x - x_i)_+ = \begin{cases} x - x_i & \text{for } x \geq x_i \\ 0 & \text{otherwise} \end{cases}. \tag{2.9}$$

If we set

$$s(x) = \alpha + \beta_1(x - x_1) + \sum_{i=2}^{n-1} \beta_i (x - x_i)_+, \tag{2.10}$$

then clearly, s restricts to a linear polynomial on each of the intervals $[x_k, x_{k+1}]$, $k = 1, \ldots, n-1$, namely,

$$s_k(x) = \alpha + \sum_{i=1}^{k} \beta_i (x - x_i).$$

The parameters $\alpha, \beta_1, \cdots, \beta_{n-1}$ can be successively calculated from the interpolation conditions (2.2). The representation (2.10) has the advantage over the previous forms that it is not necessary to always first determine in which interval the point of evaluation, v, lies. The disadvantage is that the computation of the $\beta_2, \cdots, \beta_{n-1}$ and the evaluation itself are both expensive.

A representation analogous to the Lagrange form of the interpolating polynomial is obtained from the introduction of the so-called *B-splines.* (B stands for basis). Those of first order are given by

$$N_i(x) = \begin{cases} 0 & \text{for } x \leq x_{i-1} \\ \frac{x - x_{i-1}}{\Delta x_{i-1}} & \text{for } x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{\Delta x_i} & \text{for } x_i \leq x \leq x_{i+1} \\ 0 & \text{for } x \geq x_{i+1} \end{cases} . \qquad (2.11)$$

The points $x_0 < x_1$ and $x_{n+1} > x_n$ are otherwise arbitrary. Then clearly,

$$s(x) = \sum_{i=1}^{n} y_i N_i(x) \qquad (2.12)$$

is also a representation of the interpolating polygonal path, since s restricts to a linear on $[x_k, x_{k+1}]$ and it also satisfies the interpolation conditions. Further, from the definition (2.11), only N_k and N_{k+1} differ from zero on $[x_k, x_{k+1}]$. Hence,

$$\begin{aligned} s_k(x) &= y_k N_k(x) + y_{k+1} N_{k+1}(x) \\ &= y_k u + y_{k+1} t, \end{aligned}$$

and we recover the form (2.6). B-splines ([19,20]), for $m > 2$, are an important and indispensable tool for data smoothing ([67]) in one and two variables as well as for free-form curves and surfaces ([37]). For spline interpolation with $m \leq 2$, they are, however, in the case of polynomial segments, too complicated, and for non-polynomial segments, only explicitly available in exceptional cases. Hence, we will not pursue them further.

```
      SUBROUTINE INTONE(X,N,V,I,IFLAG)
      DIMENSION X(N)
      IFLAG=0
      IF (I.GE.N) I=1
      IF (V.LT.X(1).OR.V.GT.X(N)) THEN
         IFLAG=3
         RETURN
      END IF
      IF (V.LT.X(I)) GOTO 10
      IF (V.LE.X(I+1)) RETURN
      L=N
      GOTO 30
10    L=I
      I=1
20    K=(I+L)/2
      IF (V.LT.X(K)) THEN
         L=K
      ELSE
         I=K
      END IF
30    IF (L.GT.I+1) GOTO 20
      RETURN
      END
```

Calling sequence:

CALL INTONE(X,N,V,I,IFLAG)

Purpose:
Determination of an index I with X(I)≤V≤X(I+1). X(1)<X(2)< ⋯ <X(N) is required.

Description of the parameters:

X	ARRAY(N): Abscissas of the given points.	
N	Number of given X-values.	
V	Abscissa of the point at which the spline function is to be evaluated.	
I	Input:	Upon calling I must contain a value between 1 and $n-1$.
	Output:	I with X(I)≤V≤X(I+1).
IFLAG	=0:	Normal execution.
	=3:	V<X(1) and V>X(N) not allowed.

Figure 2.1. Subroutine INTONE and its description.

```
FUNCTION POLVAL(N,X,Y,V,IFLAG)
DIMENSION X(N),Y(N)
DATA I/1/
IFLAG=0
IF (N.LT.2) THEN
   IFLAG=1
   RETURN
END IF
CALL INTONE(X,N,V,I,IFLAG)
IF (IFLAG.NE.0) RETURN
XI=X(I)
T=(V-XI)/(X(I+1)-XI)
POLVAL=Y(I+1)*T+Y(I)*(1.-T)
RETURN
END
```

FUNCTION POLVAL(N,X,Y,V,IFLAG)

Purpose:
POLVAL is a FUNCTION subprogram for the calculation of a function value of a polygonal path at a point V$\in$[X(1),X(N)].

Description of the parameters:

N	Number of given points.	
X	ARRAY(N): Abscissas.	
Y	ARRAY(N): Ordinates.	
V	Point at which the function is to be evaluated.	
IFLAG	=0:	Normal execution.
	=1:	N$\geq$2 is required.
	=3:	Error in the interval determination (INTONE).

Required subroutines: INTONE.

Remark: The statement 'DATA I/1/' has the effect that I is set to 1 at the first call to POLVAL.

Figure 2.2. Function POLVAL and its description.

2.3. Evaluation by Searching an Ordered List

As for all spline interpolants, before (2.4) or (2.6) can be used to evaluate a polygonal path $s(v)$ at an abscissa $v \in [x_1, x_n]$, there first arises the problem of finding that index i for which $x_i \le v \le x_{i+1}$. For this there are several solutions of varying efficiency ([63,109]). Here, we proceed from the reasonable assumption that function values at a monotonically increasing sequence of absicissas $v_1 < v_2 < \cdots < v_{\tilde{n}}$ are to be calculated. If $\tilde{n}$ is substantially larger than n, then it will frequently be the case that in passing from v_j to v_{j+1}, the new abscissa will lie in the same interval, with index i, as did v_j. Thus, we initialize $i = 1$ and store the (possibly changed) index i of the interval in which the last v_j lay. If v_{j+1} is not in $[x_i, x_{i+1}]$, then the new i is found by a binary search on $[x_1, x_i]$ if $v_{j+1} < x_i$ and on $[x_i, x_n]$ if $v_{j+1} \ge x_i$ (the regular case). This procedure is implemented by the subroutine INTONE of Fig. 2.1; the program description is also found in Fig. 2.1. The function POLVAL (Fig. 2.2) evaluates a polygonal path by making use of INTONE. An example showing the polygonal path interpolant to the data of Fig. 1.4c is given in Fig. 2.3. Although the "curve" appears to be very smooth, this is deceiving as there are jumps in the first derivative at the nodes. If the abscissas can be chosen to be sufficiently close together, however, then these jumps become arbitrarily small. This fact is at the heart of any plotter software.

2.4. Properties of Polygonal Paths

As opposed to spline interpolants of higher degrees, the polygonal path s has notable shape-preserving properties. Positivity in the data is preserved: if $y_k > 0$, $k = 1, \cdots, n$, then $s > 0$ on $[x_1, x_n]$. Monotonicity is also preserved: if for instance $y_1 < y_2 < \cdots < y_n$, then since $s'_k = \frac{\Delta y_k}{\Delta x_k} > 0$, $s' > 0$ on the whole interval $[x_1, x_n]$. Denote by d_k the slopes

$$d_k = \frac{\Delta y_k}{\Delta x_k}, \quad k = 1, \cdots, n-1. \tag{2.13}$$

If the given data is convex in the sense that $d_1 < d_2 < \cdots < d_{n-1}$, then since $s'' = 0$, the polygonal path is also convex. Examples of these three properties are obtained from Figs. 1.3a-c by connecting the points by line segments. Now if $y_k = f(x_k)$, $k = 1, \cdots, n$, for $f \in C^1[x_1, x_n]$, then the

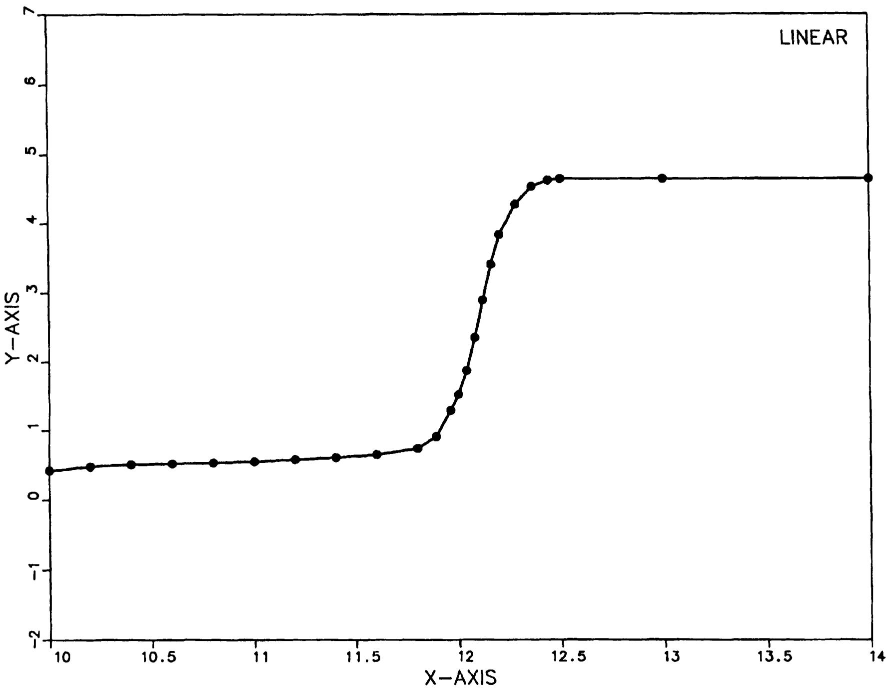


Figure 2.3.

interpolating polygonal path s has the property that

$$\int_{x_1}^{x_n} [s'(x)]^2 dx \leq \int_{x_1}^{x_n} [f'(x)]^2 dx, \tag{2.14}$$

i.e., among all interpolating functions, the polygonal path minimizes the aggregate slope squared. For the proof, we show that in

$$0 \leq \int_{x_1}^{x_n} (f' - s')^2 dx = \int_{x_1}^{x_n} (f')^2 dx - 2\int_{x_1}^{x_n} (f' - s')s' dx - \int_{x_1}^{x_n} (s')^2 dx,$$

the middle term on the right disappears. In fact,

$$\begin{aligned} \int_{x_1}^{x_n} (f' - s')s' dx &= \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} (f' - s')s' dx \\ &= \sum_{k=1}^{n-1} \left[(f - s)s' |_{x_k}^{x_{k+1}} - \int_{x_k}^{x_{k+1}} (f - s)s'' dx \right] \end{aligned}$$

by integration by parts. The first term is zero, since $f(x_k) = y_k = s(x_k)$, and the second, since $s'' = 0$.

2.5. When the Knots and Interpolation Nodes Are Different

The interpolating line segments need not necessarily be joined continuously just at the nodes x_k. We could also use parameters $\alpha_k \in (0,1)$ and define knots z_k by

$$z_k = \alpha_k x_k + (1-\alpha_k)x_{k+1}, \quad k = 1, \cdots, n-1.$$

Let

$$s_k(x) = y_k + B_k(x - x_k), \quad k = 1, \cdots, n,$$

then be n line segments passing respectively through (x_k, y_k) and corresponding to the intervals $[x_1, z_1]$, $[z_k, z_{k+1}]$, $k = 1, \cdots, n-2$, and $[z_{n-1}, x_n]$. Since $z_k - x_k = (1-\alpha_k)\Delta x_k$ and $z_k - x_{k+1} = -\alpha_k \Delta x_k$, the continuity conditions

$$s_k(z_k) = s_{k+1}(z_k), \quad k = 1, \cdots, n-1,$$

yield the linear system,

$$(1-\alpha_k)B_k + \alpha_k B_{k+1} = d_k, \quad k = 1, \cdots, n-1, \tag{2.15}$$

of $n-1$ equations in n unknowns $B_1, \cdots, B_n$. If, for example, B_1 is specified ahead of time, then it is uniquely solvable and the corresponding polygonal path is thus uniquely determined. From experience, the appearance of this interpolant depends strongly on the choice of B_1.

This dependency can be eliminated in the case of an odd number of points $n = 2m+1$ and symmetric data, i.e.,

$$y_k = y_{n+1-k}, \Delta x_k = \Delta x_{n-k}, \alpha_k = 1 - \alpha_{n-k}, \quad k = 1, \cdots, m,$$

by the reasonable requirement that $B_1 = -B_n$. The system (2.15) becomes

$$\begin{array}{llllll}
\alpha_1 B_2 & & & & -(1-\alpha_1)B_n & = d_1 \\
\cdot & \cdot & & & & \cdot \\
& (1-\alpha_m)B_m & +\alpha_m B_{m+1} & & & = d_m \quad . \\
& & \cdot & \cdot & & \cdot \\
& & & \alpha_1 B_{n-1} & +(1-\alpha_1)B_n & = -d_1
\end{array}$$

By adding the first and last equations, we see that $B_2 = -B_{n-1}$. Then, using this in the addition of the second and second from last equations, we obtain $B_3 = -B_{n-2}$. Continuing in this way, we finally obtain that $B_m = -B_{m+2}$ and from the addition of the mth and $(m+1)$st equations, that $B_{m+1} = 0$. Thus, starting with the mth equation, we may successively compute

$$B_k = -B_{n+1-k} = \frac{d_k - \alpha_k B_{k+1}}{1-\alpha_k}, \quad k = m, m-1, \cdots, 2.$$

```
      SUBROUTINE POLSYM(N,X,Y,W,B,IFLAG)
      DIMENSION X(N),Y(N),W(N),B(N)
      IFLAG=1
      IF (MOD(N,2).EQ.0.OR.N.LT.3) RETURN
      IFLAG=0
      M=(N-1)/2
      B(M+1)=0.
      DO 10 K=M,1,-1
         K1=K+1
         B(K)=((Y(K1)-Y(K))/(X(K1)-X(K))-W(K)*B(K1))/(1.-W(K))
         B(N-K+1)=-B(K)
10    CONTINUE
      RETURN
      END
```

Calling sequence:

CALL POLSYM(N,X,Y,W,B,IFLAG)

Purpose:

Calculation of a polygonal path with knots differing from interpolation nodes for data symmetric with respect to the y-axis.

Description of the parameters:

N	Number of given points n. N must be odd.	
X	ARRAY(N):	Vector of abscissas.
Y	ARRAY(N):	Vector of ordinates.
W	ARRAY(N):	Upon calling must contain the parameters α_k, $k = 1, \cdots, n-1$, with $0 < \alpha_k < 1$.
B	ARRAY(N):	Upon completion with IFLAG=0 contains the slopes of the required polygonal path.
IFLAG	=0:	Normal execution.
	=1:	N odd and N$\geq$3 are required.

Figure 2.4. Subroutine POLSYM and its description.

This procedure is implemented in the subroutine POLSYM (Fig. 2.4). An example with $\alpha_k = 1/2$, $k = 1, \cdots, 5$, is given in Fig. 2.5.

2.6. Parametric Polygonal Paths

Suppose for the moment that the general assumption (2.1) does not hold and that the numbering of the points is to correspond to the order in which

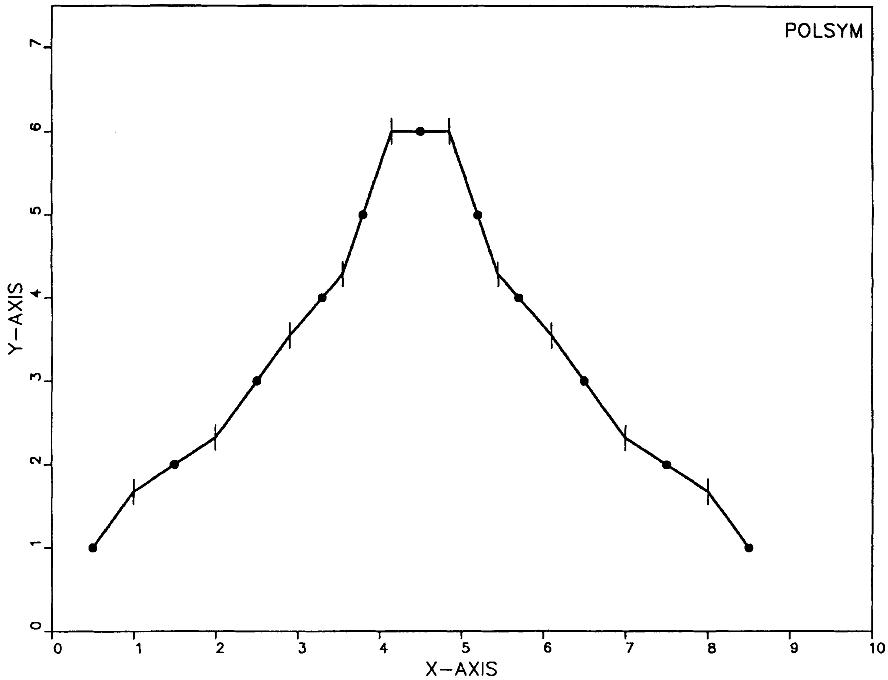


Figure 2.5.

the interpolant is to pass through them. Then, in general, the corresponding polygonal path in the plane can no longer be described by a function, and so, instead, we will make use of the parametric representation of a curve. Choose $v_1 < v_2 < \cdots < v_n$ arbitrarily and set

$$s_k(v) = \begin{cases} \xi(v) = x_k + \frac{\Delta x_k}{\Delta v_k}(v - v_k) \\ \eta(v) = y_k + \frac{\Delta y_k}{\Delta v_k}(v - v_k) \end{cases} \tag{2.16}$$

in the interval $[v_k, v_{k+1}]$, $k = 1, \cdots, n-1$. Then, as desired,

$$s_k(v_k) = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$$

and

$$s_k(v_{k+1}) = \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix}.$$

Moreover, s_k does represent the straight line between these two points, since we may eliminate $(v - v_k)/\Delta v_k$ from the equations,

$$\begin{aligned} \xi - x_k &= \frac{\Delta x_k}{\Delta v_k}(v - v_k), \\ \eta - y_k &= \frac{\Delta y_k}{\Delta v_k}(v - v_k), \end{aligned}$$

to obtain the usual equation of a line,

$$\frac{\eta - y_k}{\xi - x_k} = \frac{\Delta y_k}{\Delta x_k},$$

in case $\Delta x_k \neq 0$, or its reciprocal if $\Delta y_k \neq 0$. If $(x_1, y_1) = (x_n, y_n)$, then we obtain in this manner a closed polygonal path.

The magnitude of Δv_k has, in the case of the polgonal path, no effect on the appearance of the curve. It is suggested that one choose the v_k as the cumulative arclength along the curve, i.e,

$$v_1 = 0, \quad v_{k+1} = v_k + \sqrt{\Delta x_k^2 + \Delta y_k^2}, \quad k = 1, \cdots, n-1. \tag{2.17}$$

Here, as well as in the general case, this is called the canonical parameterization of the curve. If the curve segments are not straight lines, as we will be using later, then in general the arclength cannot be computed explicitly but (2.17) is the basis of a first approximation.

2.7. Smoothing with Polygonal Paths I

We now again assume that (2.1) holds. Further, we suppose that there are measurement errors in the y_k so that the desired

$$s_k(x) = A_k + B_k(x - x_k), \quad k = 1, \cdots, n-1, \tag{2.18}$$

is not to pass through the given points (x_k, y_k) themselves but through points (x_k, A_k) with yet to be determined "exact" ordinates A_k, $k = 1, \cdots, n$. In order to be as flexible as possible in the choice of these A_k, we introduce variable control parameters p_k and ask that the differences in the ordinates be proportional to the jumps in the first derivative of the polygonal path, i.e.,

$$p_k(A_k - y_k) = B_k - B_{k-1}, \quad k = 1, \cdots, n. \tag{2.19}$$

Here we have set $B_0 = B_n = 0$. (A similar requirement is made for cubic splines in [153]; we will return to this later. More precise reasons for this type of model can be motivated by the theory of nonlinear optimization ([163]).)

The interpolation conditions (2.2) yield

$$A_k + B_k \Delta x_k = A_{k+1}, \quad k = 1, \cdots, n-1.$$

Solving these for B_k and substituting in (2.19) with $B_0 = B_n = 0$ gives the linear system of equations,

$$M \cdot A = PY, \tag{2.20}$$

where

$$M = \begin{bmatrix} p_1 + \frac{1}{h_1} & -\frac{1}{h_1} & & & \\ -\frac{1}{h_1} & p_2 + \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} & & \\ & -\frac{1}{h_2} & p_3 + \frac{1}{h_2} + \frac{1}{h_3} & -\frac{1}{h_3} & \\ & & \cdot & \cdot & \cdot \\ & & & -\frac{1}{h_{n-1}} & p_n + \frac{1}{h_{n-1}} \end{bmatrix},$$

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \cdot \\ A_n \end{bmatrix}, \quad PY = \begin{bmatrix} p_1 y_1 \\ p_2 y_2 \\ p_3 y_3 \\ \cdot \\ p_n y_n \end{bmatrix}.$$

The $n \times n$ coefficient matrix M is symmetric and, for $p_k > 0$, strictly diagonally dominant. Hence, the system of equations is always uniquely solvable for arbitrary control parameters $p_k > 0$. In this case then, the corresponding smoothing polygonal path also exists and is unique. (We will often show the existence of spline interpolants by an argument of strict diagonal dominance. The reader not confident with this material should consult either the appendix or an appropriate textbook.)

The subroutine POLSM1 (Figs. 2.6 and 2.7) sets up the linear system (2.20) for given $p_k > 0$ and obtains the solution by calling the symmetric tridiagonal matrix solver TRIDIS (see the appendix). (TRIDIS does not use pivoting, as this is not necessary for strictly diagonally dominant matrices ([8]).) Examples computed with POLSM1 are illustrated in Figs. 2.8a and b. In Fig. 2.8a the control parameters p_k were chosen to be $p_k = 1$, $k = 1, \cdots, 7$, and in 2.8b, $p_1 = p_3 = 10$, $p_2 = p_4 = 1$, $p_5 = p_6 = p_7 = 5$. In the limit as $p_k \to \infty$, $A_k = y_k$, and thus we recover the interpolating polygonal path. This can be seen by dividing the kth row of (2.20) by p_k and then passing to the limit.

2.8. Smoothing with Polygonal Paths II

In statistics, there arises the problem ([39, 69]) of fitting a polygonal path with prescribed knots x_k, $k = 1, \cdots, n \geq 2$, to a set of points (u_i, v_i), $i = 1, \cdots, m \geq 3$, in the sense of least squares. Typically, m is substantially larger than n. In this, the assumptions that (2.1) holds and that $x_1 \leq u_i \leq x_n$, $i = 1, \cdots, m$, are also made. Abscissas u_i and x_k could be the same.

Using the B-spline representation (2.12), we wish to determine y_k corresponding to x_k, $k = 1, \cdots, n$, which minimize

$$S(y_1, \cdots, y_n) = \sum_{i=1}^{m} \left[\sum_{k=1}^{n} y_k N_k(u_i) - v_i \right]^2 \tag{2.21}$$

(see also [67], p. 71). The conditions necessary for a minimum of (2.21)

```
      SUBROUTINE POLSM1(N,X,Y,P,EPS,A,B,IFLAG,F,G)
      DIMENSION X(N),Y(N),P(N),A(N),B(N),F(N),G(N)
      IFLAG=0
      IF (N.LT.2) THEN
         IFLAG=1
         RETURN
      END IF
      H1=0.
      DO 10 K=1,N-1
         PK=P(K)
         IF (PK.LE.0.) THEN
            IFLAG=4
            RETURN
         END IF
         H2=1./(X(K+1)-X(K))
         B(K)=H2
         F(K)=PK+H1+H2
         G(K)=-H2
         A(K)=PK*Y(K)
         H1=H2
10    CONTINUE
      F(N)=P(N)+H1
      A(N)=P(N)*Y(N)
      CALL TRIDIS(N,F,G,A,EPS,IFLAG)
      IF (IFLAG.NE.0) RETURN
      DO 20 K=1,N-1
         B(K)=(A(K+1)-A(K))*B(K)
20    CONTINUE
      RETURN
      END
```

Figure 2.6. Program listing of POLSM1.

Calling sequence:

CALL POLSM1(N,X,Y,P,EPS,A,B,IFLAG,F,G)

Purpose:
Determination of the coefficients A_k and B_k of a smoothing linear spline function (knots same as nodes).

Description of the parameters:

N	Number of given points. N≥2 is required.	
X	ARRAY(N):	Upon calling must contain the abscissa values x_k, $k = 1, \cdots, n$, with $x_1 < x_2 < \cdots < x_n$.
Y	ARRAY(N):	Upon calling must contain the ordinate values y_k, $k = 1, \cdots, n$.
P	ARRAY(N):	Upon calling must contain the values of the weights p_k, $k = 1, \cdots, n$.
EPS	see TRIDIS.	
A,B	ARRAY(N):	Upon completion with IFLAG=0 contain the desired spline coefficients, $k = 1, \cdots, n$.
IFLAG	=0:	Normal execution.
	=1:	N≥2 is required.
	=2:	Error in solving the system (TRIDIS).
	=3:	$p_k > 0$ is required.
F,G	ARRAY(N):	Work space.

Required subroutine: TRIDIS.

Figure 2.7. Description of Subroutine POLSM1.

are

$$\frac{\partial S}{\partial y_j} = 2\sum_{i=1}^{m} N_j(u_i)\left[\sum_{k=1}^{n} N_k(u_i)y_k - v_i\right] = 0, \tag{2.22}$$

which yield the linear system of equations,

$$\sum_{i=1}^{m}\sum_{k=1}^{n} N_j(u_i)N_k(u_i)y_k = \sum_{i=1}^{m} N_j(u_i)v_i, \quad j = 1, \cdots, n. \tag{2.23}$$

Its coefficient matrix C can be written as

$$C = A^T A,$$

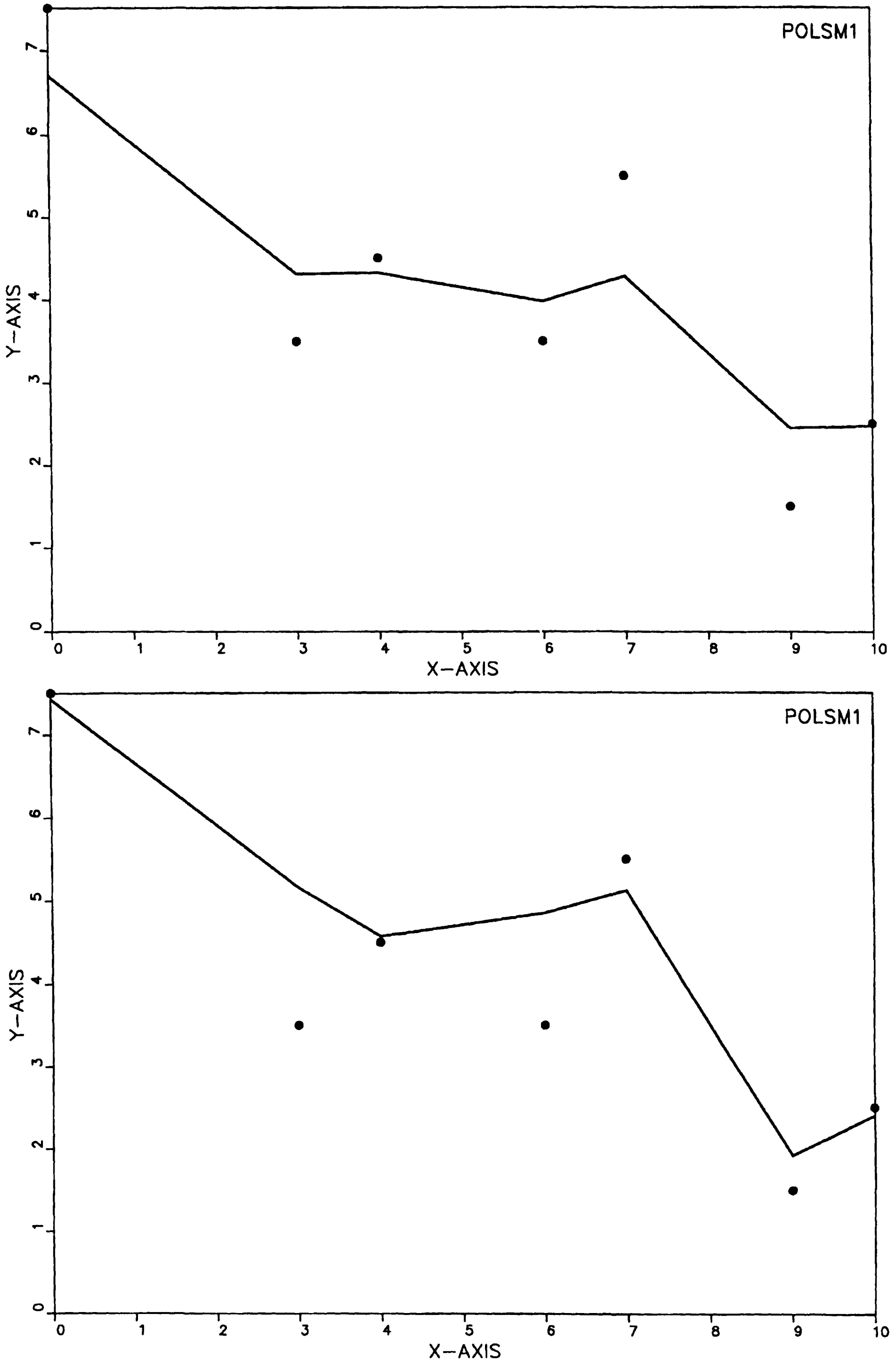


Figure 2.8. a, b.

with

$$A = \begin{bmatrix} N_1(u_1) & N_2(u_1) & \cdots & N_n(u_1) \\ N_1(u_2) & N_2(u_2) & \cdots & N_n(u_2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ N_1(u_m) & N_2(u_m) & \cdots & N_n(u_m) \end{bmatrix},$$

and so, in particular, it is symmetric. In order for C to be nonsingular, it is necessary that the rank of A be n, which necessarily presupposes $m \geq n$ and an appropriate distribution of the abscissas u_i and x_k. (Sufficient conditions do not seem to be known.) Further, since $N_k(u_i)N_{k+2}(u_i) = 0$ (see (2.11)), C is tridiagonal. The summands in the sums over i are determined by those elements of the diagonal and sub- and super-diagonals, which are in general nonzero. Specifically, these are

$$(N_k(u_i))^2 = \begin{cases} \left(\frac{u_i - x_{k-1}}{\Delta x_{k-1}}\right)^2 & \text{for } x_{k-1} \leq u_i \leq x_k \\ \left(\frac{x_{k+1} - u_i}{\Delta x_k}\right)^2 & \text{for } x_k \leq u_i \leq x_{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

and

$$N_k(u_i)N_{k+1}(u_i) = \begin{cases} 0 & \text{for } x_{k-1} \leq u_i \leq x_k \\ \frac{x_{k+1} - u_i}{\Delta x_k}\frac{u_i - x_k}{\Delta x_k} & \text{for } x_k \leq u_i \leq x_{k+1} \\ 0 & \text{for } x_{k+1} \leq u_i \leq x_{k+2} \end{cases}. \tag{2.25}$$

Evidently, the elements of C are also nonnegative. For arbitrary distributions of the u_i and x_k, this system of equations cannot, unfortunately, easily seen to be strictly diagonally dominant.

The subroutine POLSM2 (Figs. 2.9 and 2.10) forms the linear system (2.23) by means of (2.24) and (2.25) and attempts to solve it with TRIDIS (thereby assuming that no pivoting is necessary). If TRIDIS is not able to run till completion (see its description), execution is terminated. This is never the case in examples of practical importance. The resulting $A_k = y_k$, $k = 1, \cdots, n$, and B_k, $k = 1, \cdots, n-1$, are the coefficients of the representation (2.4), and so B-splines need not be involved in evaluation of the polygonal path.

Four examples with the same initial data are given in Figs. 2.11a–2.12b. The following choices of knots were made. For all of them, $x_1 = u_1$, and then for 2.11a, $x_2 = u_9$, for 2.11b, $x_2 = u_4$, $x_3 = u_9$, for 2.12a, $x_2 = (u_6 + u_7)/2$, $x_3 = u_9$, and finally for 2.12b, $x_2 = u_4$, $x_3 = (u_6 + u_7)/2$, and $x_4 = u_9$. Figure 2.11a shows the smoothing straight line.

For the practical determination of the knots x_k, the number of which should be kept as small as possible for practical reasons, one can proceed

```
      SUBROUTINE POLSM2(M,N,U,V,X,EPS,A,B,IFLAG,F,G)
      DIMENSION U(M),V(M),X(N),A(N),B(N),F(N),G(N)
      IFLAG=0
      IF (M.LT.3.OR.N.LT.2.OR.M.LT.N) THEN
         IFLAG=1
         RETURN
      END IF
      DO 10 K=1,N-1
         B(K)=X(K+1)-X(K)
10    CONTINUE
      DO 30 K=1,N
         K1=K+1
         K2=K-1
         T=0.
         B2=0.
         RS=0.
         DO 20 I=1,M
            R1=0.
            R2=0.
            UH=U(I)
            IF (K.GT.1) THEN
               IF (UH.GE.X(K2).AND.UH.LE.X(K)) THEN
                  R1=(UH-X(K2))/B(K2)
               END IF
            END IF
            IF (K.LT.N) THEN
               IF (UH.GE.X(K).AND.UH.LE.X(K1)) THEN
                  R1=(X(K1)-UH)/B(K)
                  R2=(UH-X(K))/B(K)
               END IF
            END IF
            T=T+R1*R2
            B2=B2+R1*R2
            RS=RS+R1*V(I)
20       CONTINUE
         F(K)=T
         A(K)=RS
         IF (K.LT.N) G(K)=B2
30    CONTINUE
      CALL TRIDIS(N,F,G,A,EPS,IFLAG)
      IF (IFLAG.NE.0) RETURN
      DO 40 K=1,N-1
         B(K)=(A(K+1)-A(K))/B(K)
40    CONTINUE
      RETURN
      END
```

Figure 2.9. Program listing of POLSM2.

Calling sequence:

CALL POLSM2(M,N,U,V,X,EPS,A,B,IFLAG,F,G)

Purpose: Determination of a smoothing polygonal path with fewer knots than interpolation points.

Description of the parameters:

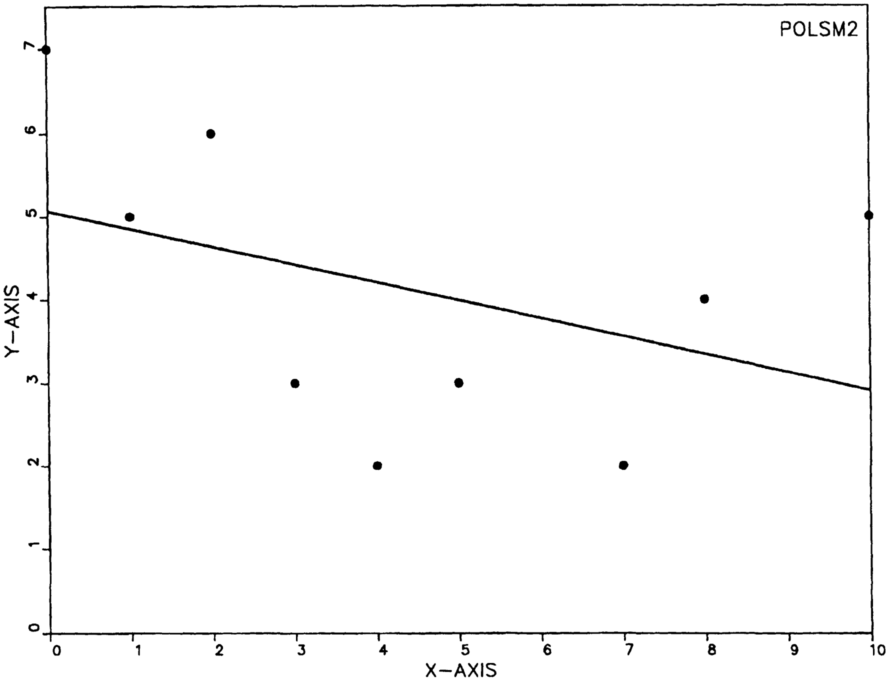
M	Number of given points.	
N	Number of knots.	
U	ARRAY(M):	Upon calling must contain the abscissa values u_k, $k = 1, \cdots, m$, with $u_1 \leq u_2 \leq \cdots \leq u_m$.
V	ARRAY(M):	Upon calling must contain the ordinate values v_k, $k = 1, \cdots, m$.
X	ARRAY(N):	Upon calling must contain the values x_k, $k = 1, \cdots, n$.
EPS	see TRIDIS.	
A,B	ARRAY(N):	Upon execution with IFLAG=0 contain the desired spline coefficients, $k = 1, \cdots, n-1$.
IFLAG	=0:	Normal execution.
	=1:	M≥3 and N≥2 and M≥N are required.
	=2:	Error in solving the linear system (TRIDIS).
F,G	ARRAY(N):	Work space.

Required subroutines: TRIDIS.

Figure 2.10. Description of Subroutine POLSM2.

in a manner analogous to that for cubic splines ([58]). Initially, choose $n = 2$, and $x_1 = u_1$ and $x_2 = u_m$, then $n = 3$, with $x_1 = u_1$, $x_3 = u_m$, and x_2 chosen so that about the same number of abscissas u_i lie on either side of it. The interval $[x_1, x_2]$ or $[x_2, x_3]$ for which the sum of the squares of the errors is largest is then again so subdivided and so on until a prescribed maximum value of $n \leq m$ is attained.

It would be very difficult to fix n and determine the x_k so as to also minimize (2.20) in these variables. We will not consider such so-called *free knot* problems; such do not arise in the interpolation problems constituting our main object of study.



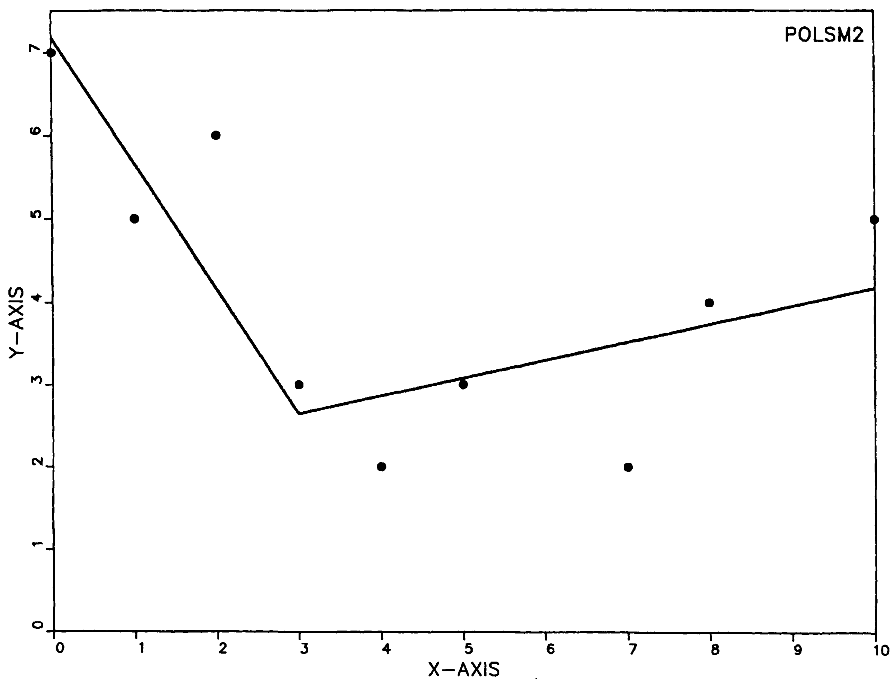


Figure 2.11. a, b.

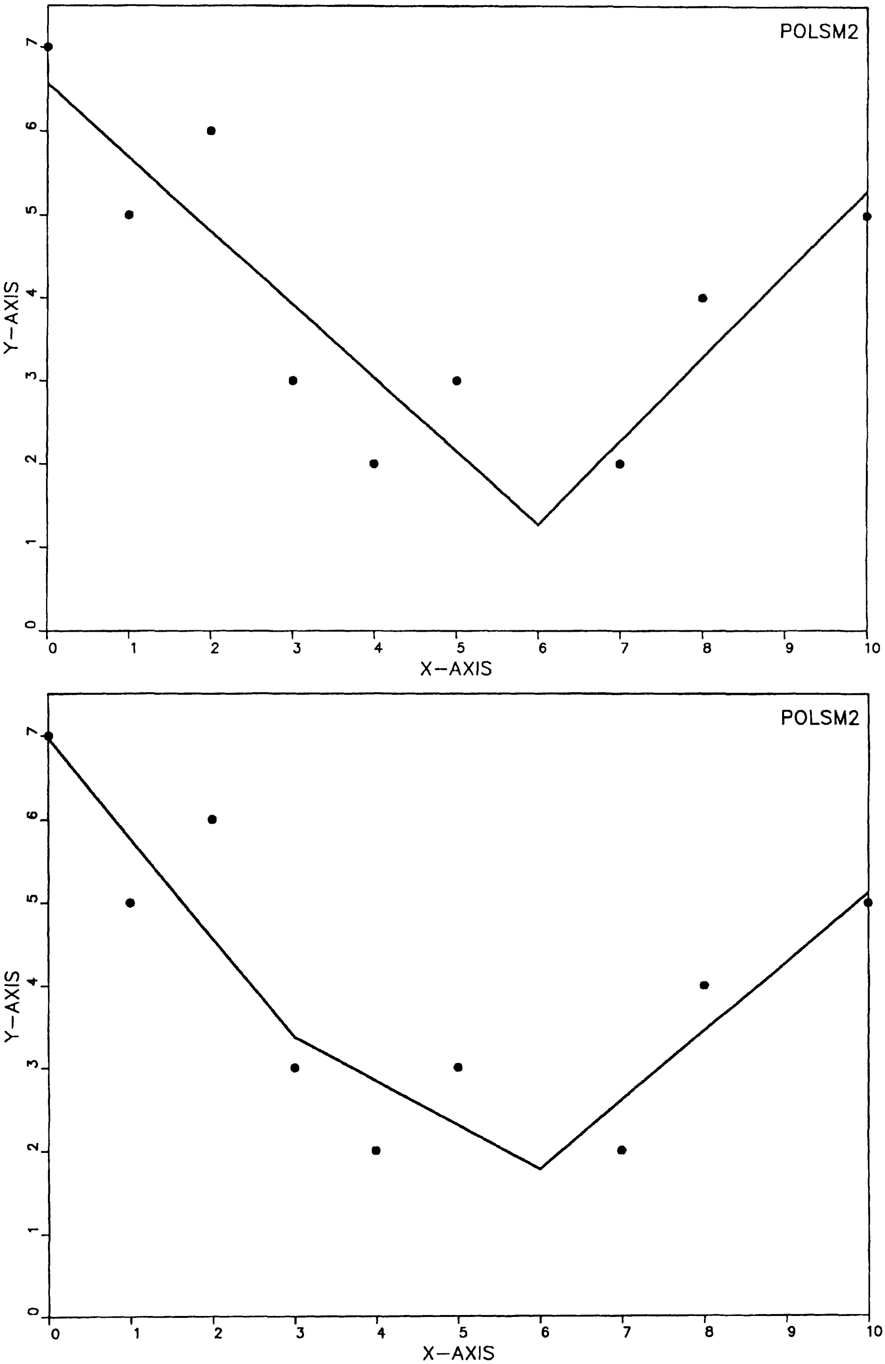


Figure 2.12. a, b.

3

Quadratic Spline Interpolants

3.1. Knots the Same as Nodes

Suppose once more that (2.1) holds. We wish now to join together parabolic segments,

$$s_k(x) = A_k + B_k(x - x_k) + C_k(x - x_k)^2, \quad k = 1, \cdots, n-1, \tag{3.1}$$

at the nodes x_k so as to form a *once* continuously differentiable quadratic spline interpolant s. The interpolation conditions (2.2) become

$$\begin{aligned} A_k &= y_k, \\ A_k + h_k B_k + h_k^2 C_k &= y_{k+1}, \end{aligned} \tag{3.2}$$

from which we obtain the relation,

$$B_k + h_k C_k = d_k, \quad k = 1, \cdots, n-1. \tag{3.3}$$

Since

$$s_k'(x) = B_k + 2\,C_k(x - x_k), \tag{3.4}$$

the C^1 conditions yield the equations,

$$B_{k-1} + 2\,h_{k-1}C_{k-1} = B_k, \quad k = 2, \cdots, n-1. \tag{3.5}$$

Now solve (3.3) for C_k to obtain

$$C_k = \frac{1}{h_k}(d_k - B_k), \quad k = 1, \cdots, n-1. \tag{3.6}$$

Moreover, if we introduce the additional unknown B_n, then (3.5) holds also for $k = n$, and so substituting (3.6) into (3.5), we obtain ([87]) the linear system of equations,

$$B_{k-1} + B_k = 2\,d_{k-1}, \quad k = 2, \cdots, n, \tag{3.7}$$

for the determination of $B_1, \cdots, B_n$. Unfortunately, this system consists of n unknowns but only $n-1$ equations, and thus we require one extra condition. If, for example, we fix a value for B_1, then (3.7) can easily be solved recursively. It can be shown by complete induction that

$$B_k = (-1)^{k+1} B_1 + 2 \sum_{j=1}^{k-1} (-1)^{k+j+1} d_j, \quad k = 2, \cdots, n. \tag{3.8}$$

The coefficients of (3.1) are then uniquely determined by (3.8), (3.2), and (3.6).

From (3.8), one readily sees a *shape-preserving* property of s ([87]). Suppose that $y_k \geq y_{k-1}$, $k = 2, \cdots, n$, and $d_k \geq d_{k-1}$, $k = 2, \cdots, n-1$, as well as that $0 \leq B_1 \leq 2\,d_1$. Then it follows that $B_k \geq 0$ for $k = 1, \cdots, n$. But then $s'(x)$ is continuous piecewise linear and, by (3.4), nonnegative at the x_k. Hence, $s'(x) \geq 0$ and we see that a certain kind of *monotonicity* is preserved. By substituting (3.8) in (3.6) and using the fact that $s_k''(x) = 2\,C_k$, we can obtain similar conditions for *convexity* preservation ([87]).

3.2. Optimal Initial Slope

For the choice of the value B_1 of the slope at x_1, $B_1 = d_1$, for example, suggests itself. But we could also ask that B_1 be, in a certain sense, optimal. For example, the minimality property (2.14) of polygonal paths suggests that we distinguish an s among all quadratic spline interplolants by choosing B_1 to minimize

$$\tilde{F}_1 = \int_{x_1}^{x_n} [s'(x)]^2 dx.$$

We may calculate

$$\tilde{F}_1 \;=\; \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} [s_k'(x)]^2 dx$$

$$
\begin{aligned}
&= \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} [B_k + 2\,C_k(x - x_k)]^2 dx \\
&= \sum_{k=1}^{n-1} (B_k^2 h_k + 2\,B_k C_k h_k^2 + \frac{4}{3} C_k^2 h_k^3) \\
&= \sum_{k=1}^{n-1} h_k (B_k^2 + 2\,B_k(d_k - B_k) + \frac{4}{3}(d_k - B_k)^2) \\
&= \sum_{k=1}^{n-1} h_k (\frac{1}{3} B_k^2 - \frac{2}{3} B_k d_k + \frac{4}{3} d_k^2) \\
&= \frac{1}{3} \sum_{k=1}^{n-1} h_k (B_k - d_k)^2 + \sum_{k=1}^{n-1} h_k d_k^2 .
\end{aligned}
$$

As the second term in the last line is independent of the B_k, it suffices to minimize

$$
F_1(B_1, \cdots, B_{n-1}) = \frac{1}{3} \sum_{k=1}^{n-1} h_k (B_k - d_k)^2 . \tag{3.9}
$$

A second possibility that offers itself is ([75])

$$
F_2 = \int_{x_1}^{x_n} [s''(x)]^2 dx ,
$$

as the integral on the right side, as we shall see later, is minimized by certain cubic spline interpolants. It is easily seen that

$$
\begin{aligned}
F_2 &= \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} [s_k''(x)]^2 dx \\
&= 4 \sum_{k=1}^{n-1} h_k C_k^2 ,
\end{aligned}
$$

and thus by (3.6),

$$
F_2(B_1, \cdots, B_{n-1}) = 4 \sum_{k=1}^{n-1} \frac{1}{h_k} (B_k - d_k)^2 . \tag{3.10}
$$

A third choice is to take a *convex combination*,

$$
\begin{aligned}
F_3(B_1, \cdots, B_{n-1}) &= \lambda F_1(B_1, \cdots, B_{n-1}) + (1 - \lambda) F_2(B_1, \ldots, B_{n-1}) \\
&= \sum_{k=1}^{n-1} \left(\frac{\lambda}{3} h_k + \frac{4(1 - \lambda)}{h_k} \right) (B_k - d_k)^2 , \qquad (3.11)
\end{aligned}
$$