Old Dominion University

ODU Digital Commons

Mathematics & Statistics Theses & Dissertations

Mathematics & Statistics

Summer 1985

Minimal Norm Constrained Interpolation

Larry Dean Irvine
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/mathstat_etds

 Part of the Fluid Dynamics Commons, and the Mathematics Commons

Recommended Citation

Irvine, Larry D.. "Minimal Norm Constrained Interpolation" (1985). Doctor of Philosophy (PhD), Dissertation, Mathematics & Statistics, Old Dominion University, DOI: 10.25777/k9fd-rm76
https://digitalcommons.odu.edu/mathstat_etds/100

This Dissertation is brought to you for free and open access by the Mathematics & Statistics at ODU Digital Commons. It has been accepted for inclusion in Mathematics & Statistics Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

Minimal Norm

Constrained Interpolation

by

Larry Dean Irvine
B.S. May 1981, Georgetown College
M.S. December 1982, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

Computational and Applied Mathematics

Old Dominion University
August, 1985

Approved by:

Philip W. Smith (Director)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

MINIMAL NORM CONSTRAINED INTERPOLATION

Larry Dean Irvine
Old Dominion University, 1985
Director: Dr. Philip W. Smith

In computatonal fluid dynamics and in CAD/CAM a physical boundary, usually known only discreetly (say, from a set of measurements), must often be approximated. An acceptable approximation must, of course, preserve the salient features of the data (convexity, concavity, etc.) In this dissertation we compute a smooth interpolant which is locally convex where the data are locally convex and is locally concave where the data are locally concave.

Such an interpolant is found by posing and solving a minimization problem. The solution is a piecewise cubic polynomial. We actually solve this problem indirectly by using the Peano kernel theorem to recast this problem into an equivalent minimization problem having the second derivative of the interpolant as the solution.

We are then led to solve a nonlinear system of equations. We show that with Newton's method we have an exceptionally attractive and efficient method for solving this nonlinear system of equations.

We display examples of such interpolants as well as convergence results obtained by using Newton's method. We list a FORTRAN program to compute these shape-preserving interpolants.

Next we consider the problem of computing the interpolant of minimal norm from a convex cone in a normed dual space. This is an extension of de Boor's work on minimal norm unconstrained interpolation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Dedication

To Dr. Philip Smith for his help and guidance.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Acknowledgements

I wish to express my appreciation to Dr. Philip Smith, Dr. Robert Smith, and Dr. John Swetits for their advice, assistance, and encouragement during my doctoral work and research. I wish to thank Dr. Surendra N. Tiwari for the financial support which I received from the Institute for Computational and Applied Mechanics.

I also wish to thank Barbara Jeffrey for typing the dissertation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1. The Natural Spline Interpolant

We consider the problem of computing an interpolant to given data. Throughout our discussion we shall denote the data

$$(t_i, y_i) \qquad i = 1,2,\ldots,n$$

where $a = t_1 < t_2 < \ldots < t_n = b$ and in this chapter we place no restrictions on the numbers y_i. There are, of course, many such interpolants which we can form. For example, we can calculate the unique polynomial p of order n (degree n-1 or less) which interpolates the data. However, as pointed out in [deB(1), chapter 2], for large n (and especially for equally spaced points t_i) the polynomial interpolant is notorious for large changes in its first derivative near the endpoints. Figure (1.1) displays the polynomial interpolant to the function

$$f(t) = \frac{1 - \sin(7\,\pi\, t)}{2}$$

at the points $t_i = (i-1)/10$ for $i = 1,2,\ldots,11$. Since $0 \leq y_i \leq 1$ for each i, we expect a good interpolant to remain reasonably close to these bounds. However, because of its behavior near the endpoints, the polynomial interpolant fails to model the data well. This behavior is typical of high-order polynomial interpolants.

In order to decrease the unnaturally large changes in the first derivative characteristic of the polynomial interpolant, we wish to calculate the interpolant which "bends" the least over all suitable interpolants. The norm of the second derivative of an interpolant will furnish a measure of the bending of the interpolant so we pose a minimization problem on $L_2^{(2)}[a,b]$, the Sobolev space of functions with

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

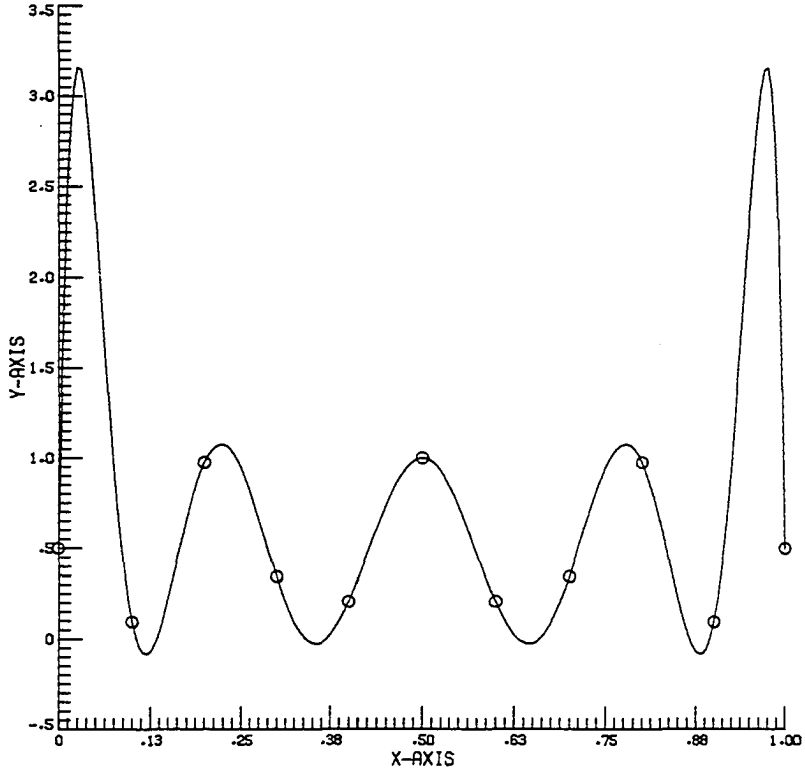

Figure (1.1): The Polynomial Interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

second derivatives in the normed linear space $L_2[a,b]$. Let A denote the set of all interpolants in the Sobolev space. We consider the minimization problem

$$\text{Find } f_* \in A \text{ such that } \|f_*^{(2)}\|_2 \leq \|f^{(2)}\|_2 \text{ for all } f \in A. \quad (1.1)$$

We shall see that the solution to (1.1) is piecewise cubic with two continuous derivatives; that is

$$f_*(t) = p_i(t) \quad \text{if} \quad t_i \leq t \leq t_{i+1}$$

for $i = 1,2,\ldots,n-1$ where p_i is a cubic polynomial and f_* is in $C^2[a,b]$. We follow the pattern in [deB(1), chapter 5], taking advantage of the fact that $L_2[a,b]$ is not only a normed linear space, but also a Hilbert space with an inner product defined by

$$(f,g) = \int_a^b f(t)g(t)dt$$

for any two elements f and g in $L_2[a,b]$.

Assume f is an element of A. (The set A is nonempty since it contains the polynomial interpolant.) We shall use the Peano kernel theorem to obtain a set of equations for $f^{(2)}$. By the Fundamental Theorem of Calculus we have

$$f(t) = f(a) + \int_a^t f^{(1)}(s)ds \quad (1.2)$$

We integrate $\int_a^t f^{(1)}(s)ds$ by parts noting that $\int udv = uv - \int vdu$. Let

$$u(s) = f^{(1)}(s) \text{ and } dv(s) = ds$$

so that

$$du(s) = f^{(2)}(s)ds \text{ and } v(s) = -(t-s)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where t is a constant. Hence

$$\int_a^t f^{(1)}(s)ds = (t-a)f^{(1)}(a) + \int_a^t (t-s)f^{(2)}(s)ds$$

and so (1.2) becomes

$$f(t) = q_1(t) + \int_a^t (t-s)f^{(2)}(s)ds \tag{1.3}$$

where $q_1(t) = f(a) + f^{(1)}(a)(t-a)$. (This is actually a Taylor's series with integral remainder.)

To acquire constant limit of integration we can write (1.3) as

$$f(t) = q_1(t) + \int_a^b (t-s)_+^{1} f^{(2)}(s)ds \tag{1.4}$$

where $(h)_+$, the positive part of the function h, is defined by

$$(h)_+(t) = \begin{cases} h(t) & \text{if} \quad h(t) \geq 0 \\ 0 & \text{if} \quad h(t) \leq 0. \end{cases}$$

Now we consider the divided difference operator. Given a function g and a set of points $\{\tau_i, \tau_{i+1}, \ldots, \tau_{i+m}\}$, the m-th divided difference of g - denoted by $[\tau_i, \tau_{i+1}, \ldots, \tau_{i+m}]g(\cdot)$ - is the leading coefficient of the polynomial of order m+1 which interpolates g at $\tau_i, \tau_{i+1}, \ldots \tau_{i+m}$ (and hence is a function of $\tau_i, \tau_{i+1}, \ldots, \tau_{i+m}$). The recursive relations

$$[\tau_p]g(\cdot) = g(\tau_p)$$

$$[\tau_i, \tau_{i+1}, \ldots, \tau_{i+m}]g(\cdot) = \frac{[\tau_{i+1}, \ldots, \tau_{i+m}]g(\cdot) - [\tau_i, \ldots, \tau_{i+m-1}]g(\cdot)}{\tau_{i+m} - \tau_i} \tag{1.5}$$

hold if $\tau_{i+m} \neq \tau_i$ (which we assume for our data). Presently we are interested in the case m=2. Equation (1.5) becomes (with $\tau_i = t_i$)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$(t_{i+2} - t_i)[t_i,t_{i+1},t_{i+2}]g(\cdot) = \frac{g(t_{i+2})-g(t_{i+1})}{t_{I+2} - t_{i+1}} - \frac{g(t_{i+1})-g(t_i)}{t_{i+1} - t_i} \tag{1.6}$$

which is computable for i=1,2,...,n-2.

Notice that $[\tau_i,\tau_{i+1},\ldots,\tau_{i+m}]p(\cdot) = 0$ if p is a polynomial of order m or less (degree m-1 or less). (From equation (1.6) we see that $(t_{i+2} - t_i)[t_i,t_{i+1},t_{i+2}]g(\cdot)$ measures a difference in slopes; the difference in slopes being zero if g is linear.)

Now we apply the (scaled) second-divided difference operator $(t_{i+2} - t_i)[t_i,t_{i+1},t_{i+2}]$ to (1.4) and interchange the order of the integral and divided difference operators to obtain

$$d_{i,2} = \int_a^b g(s)N_i(s)ds \quad i=1,2,\ldots,n-2 \tag{1.7}$$

where

$$d_{i,2} = (t_{i+2} - t_i)[t_i,t_{i+1},t_{i+2}]f(\cdot)$$

$$= \frac{y_{i+2} - y_{i+1}}{t_{i+2} - t_{i+1}} - \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \quad , \tag{1.8}$$

$$N_{i,2}(\cdot) = (t_{i,2} - t_i)[t_i,t_{i+1},t_{i+2}](\cdot -s)_+$$

$$= \frac{(t_{i+2} - s)_+ - (t_{i+1} - s)_+}{t_{i+2} - t_{i+1}} - \frac{(t_{i+1} - s)_+ - (t_i - s)_+}{t_{i+1} - t_i} \tag{1.9}$$

and $g = f^{(2)}$. We call $N_{i,2}$ the (normalized) linear B-spline (or B-spline of order 2) with knots t_i, t_{i+1} and t_{i+2}. The graph of $N_{i,2}$ is displayed in figure (1.2).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

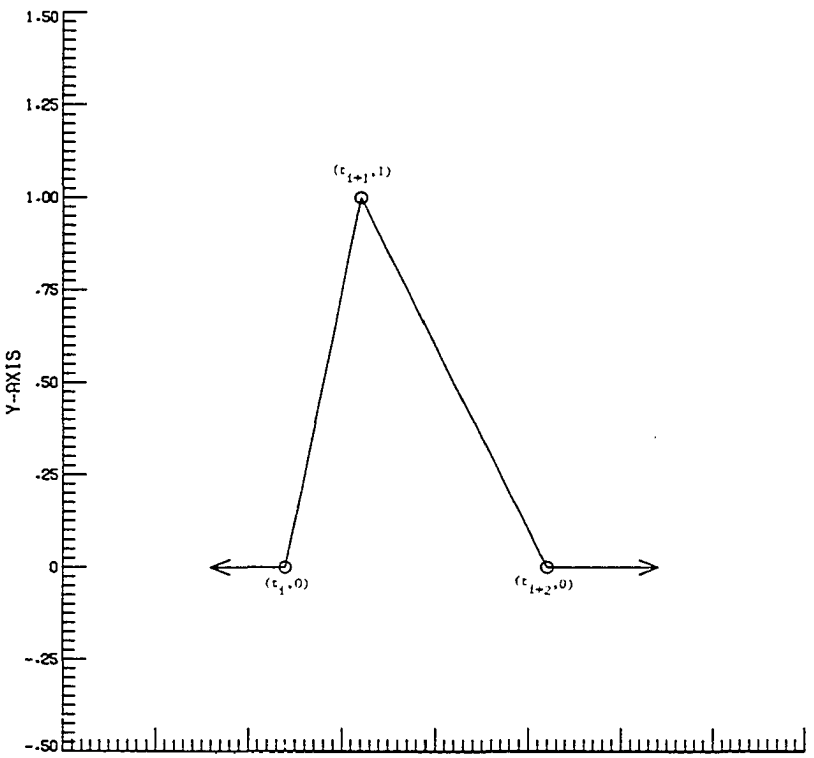

Figure (1.2): The Normalized Linear B-spline.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We have shown that if f is an interpolant in the Sobolev space ($f \in A$), then $g = f^{(2)}$ satisfies (1.7). Let the set B consist of all functions which are in $L_2[a,b]$ and which satisfy (1.7).

Now consider the problem

$$\text{Find } g_* \in B \text{ such that } \|g_*\|_2 \leq \|g\|_2 \text{ for all } g \in B \qquad (1.10)$$

A unique solution exists since (1.10) is a minimal norm problem over a nonempty closed convex set in a Hilbert space. Furthermore, the solutions of problems (1.1) and (1.10) are related via $g_* = f_*^{(2)}$. Hence, to compute f_* we can first calculate g_* and then integrate g_* twice. Since much of our emphasis will be on g_*, rather than f_*, we shall call g_* the interpolant of minimal norm.

For brevity we denote the index $m = n-2$, the B-spline $N_i = N_{i,2}$, and the divided difference $d_i = d_{i,2}$. We also define the vector-valued function $T: L_2[a,b] \to R^m$ by

$$(Tx)_i = \int_a^b x(t)N_i(t)dt \qquad i=1,2,\ldots,m.$$

To solve problem (1.10) we shall show that g_*, the interpolant of minimal norm, is the intersection of two specific sets--one an orthogonal complement of a subspace and the other a translate of a subspace --in $L_2[a,b]$ via a variation of the Projection Theorem. If W is a closed subspace of a Hilbert space H and if x is an arbitrary element of H, then the Projection Theorem states that there exists a unique element w_o in W satisfying

$$\|x - w_o\| \leq \|x - w\| \quad \text{for all } w \in W \qquad (1.11)$$

and characterized by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$(x - w_o, w) = 0 \quad \text{for all } w \in W.$$

Hence $x - w_o$ is in $W^{\perp}$, the orthogonal complement of W. The proof of the Projection Theorem can be found in any book dealing with Hilbert spaces (for example, [L, page 517]). The next proposition will serve as the actual form of the Projection theorem which we shall use.

Proposition ([L, page 64]): Let W be a closed subspace in a Hilbert space H. For a fixed element x in H define V: = x + W. Then there exists a unique element x_o in V of minimal norm. Furthermore, x_o is in $W^{\perp}$.

(The translate V is called an affine set or linear variety.) Notice that x_o is the intersection of the orthogonal complement of W and the translate V of W. In fact, (1.11) reveals that $x_o = x - w_o$.

Define

$$W: = \{z \in L_2[a,b] : Tz = \theta\}$$

which is a closed subspace in $L_2[a,b]$. Let $g \in L_2[a,b]$ be any element such that $Tg = \underline{d}$. (Equivalently, let g be any element of B.) Then $B = g + W$ and B corresponds to the linear variety in the proposition. Hence g_* is the unique element in $W^{\perp}$ satisfying $Tg_* = \underline{d}$.

We consider the contents of $W^{\perp}$. Any element which is orthogonal to each N_i is also orthogonal to any linear combination of the B-splines. Hence S: = span$(N_1, N_2, \ldots, N_m)$ is a subset of $W^{\perp}$. We now show that $W^{\perp}$ is a subset of S (and hence $S = W^{\perp}$) by contradiction. Assume that there exists an element y which is in $W^{\perp}$ but not in S. Since S is a closed subspace there exists (by the Projection Theorem)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

an element s_o in S such that

$$\| y - s_o \| \leq \| y - s \| \quad \text{for all } s \in S$$

with $y - s_o$ in the orthogonal complement of S. This implies $T(y - s_o) = \theta$ or $(y - s_o) \in W$. However $y - s_o$ is also in $W^{\perp}$ since both y and s_o are in $W^{\perp}$. Therefore $(y - s_o) = \theta$ and $S = W^{\perp}$.

In summary, g_* is characterized by

$$g_* = \sum_{i=1}^{m} \alpha_i N_i$$

(since g_* is in the span of the B-splines) where the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_m$ are chosen to satisfy

$$\left(\sum_{j=1}^{m} \alpha_j N_j, N_i \right) = d_i \quad i=1,2,\ldots,m \tag{1.12}$$

(since $Tg_* = \underline{d}$). Equation (1.13), a system of m linear equations in m unknowns, can be written in matrix notation as

$$A\underline{\alpha} = \underline{d} \tag{1.13}$$

where the symmetric matrix A has entries $A_{ij} = (N_i, N_j)$.

Because the B-splines are linearly independent, the matrix A, a Grahm matrix, is nonsingular and hence a unique solution exists for any given $\underline{d}$. Furthermore, since N_i has support $[t_i, t_{i+2}]$, the matrix A is tridiagonal. For any $\underline{x} \in R^m$ we have

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\underline{x}^T A\underline{x} = \sum_{i=1}^{m} x_i (A\underline{x})_i$$

$$= \sum_{i=1}^{m} x_i (N_i, \sum_{j=1}^{m} x_j N_j)$$

$$= (\sum_{i=1}^{m} x_i N_i, \sum_{j=1}^{m} x_j N_j)$$

$$= \| \sum_{i=1}^{m} x_i N_i \|_2^2$$

$$\geq 0$$

with equality holding if and only if $\underline{x} = \theta$. The matrix A is hence positive definite and (1.13) can be solved by Gauss elimination without pivoting, or, better still, by Cholesky decomposition.

We note also that

$$\| g_* \| = \underline{\alpha}^T A \underline{\alpha} = \underline{\alpha}^T \underline{d}.$$

The entry A_{ij}, the integral of the product of two piecewise linear polynomials, can be computed exactly by Simpson's rule applied on each subinterval $[t_k, t_{k+1}]$. Denoting $\Delta t_k := t_{k+1} - t_k$ and z_k the midpoint of the interval $[t_k, t_{k+1}]$ we have for i=1,2,...,m

$$A_{ii} = \int_{t_i}^{t_{i+1}} N_i(t)^2 dt + \int_{t_{i+1}}^{t_{i+2}} N_i(t)^2 dt$$

$$= (\Delta t_{i+1}/6)[N_i(t_i)^2 + 4N_i(z_i)^2 + N_i(t_{i+1})^2]$$

$$+ (\Delta t_{i+2}/6)[N_i(t_{i+1})^2 + 4N_i(z_{i+1})^2 + N_i(t_{i+2})^2]$$

$$= (t_{i+2} - t_i)/3.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We also compute for i=1,2,...,m-1

$$A_{i,i+1} = A_{i+1,i}$$

$$= \int_{t_{i+1}}^{t_{i+2}} N_i N_{i+1}(t)dt$$

$$= (t_{i+2} - t_{i+1})/6.$$

The solution g_*, being a linear combination of linear B-splines, is piecewise linear (and continuous) with knots t_i. After integrating g_* twice and applying the interpolation conditions, we obtain f_* which is piecewise cubic (with knots t_i) with two continuous derivatives.

Define $\underline{\beta} \in R^n$ via

$$\beta_i = \begin{cases} 0 & i=1 \\ \alpha_{i-1} & i=2,3,\ldots,n-1 \\ 0 & i=n \end{cases}$$

and $\Delta\beta = \beta_{i+1} - \beta_i$. On $[t_i,t_{i+1}]$ f_* is defined by a unique cubic polynomial p_* and hence f_* can be determined by specifying the numbers $p_{*i}^{(j)}(t_i)$ for i=1,2,...,n-1 and j=0,1,2,3. Then

$$f_*(t) = \frac{p_{*i}^{(2)}(t_i)}{0!} + \frac{p_{*i}(t_i)}{1!}(t-t_i)$$

$$+ \frac{p_{*i}^{(2)}(t_i)}{2!}(t-t_i)^2 + \frac{p_{*i}^{(3)}(t_i)}{3!}(t-t_i)^3 \qquad (1.14)$$

for $t \in [t_i, t_{i+1}]$. Of course, (1.14) can be more efficiently evaluated by using nested multiplication.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The polynomial p_{*i} solves the differential equation

$$p_{*i}^{(2)}(t) = \beta_i + (\Delta\beta_i/\Delta t_i)(t-t_i) \tag{1.15}$$

on the interval $[t_i,t_{i+1}]$ with boundary conditions $p_{*i}(t_i) = y_i$ and $p_{*i}(t_{i+1}) = y_{i+1}$. Therefore

$$p_{*i}(t) = \frac{\beta_i}{2}(t-t_i)^2 + \frac{\Delta\beta_i}{6\Delta t_i}(t-t_i)^3 + c_i(t-t_i) + e_i \tag{1.16}$$

where the constants c_i and e_i are evaluated as $e_i = y_i$ and

$$c_i = \frac{\Delta y_i}{\Delta t_i} - \left(\frac{\beta_{i+1}}{2} + \frac{\Delta\beta_i}{6}\right)\Delta t_i \tag{1.17}$$

with $\Delta y_i = y_{i+1} - y_i$. From (1.17) we obtain

$$\begin{aligned} p_{*i}^{(0)}(t_i) &= y_i \\ p_{*i}^{(1)}(t_i) &= c_i \\ p_{*i}^{(2)}(t_i) &= \beta_i \\ p_{*i}^{(3)}(t_i) &= \Delta\beta_i/\Delta t_i \end{aligned} \tag{1.18}$$

where c_i is given by (1.17). A complete FORTRAN program for computing the natural cubic spline interpolant is given in Appendix A.

Figure (1.3) displays the natural cubic spline interpolant that is in contrast to the polynomial interpolant of figure (1.1).

We complete this chapter by posing (and solving) a generalization of problem (1.1). For k fixed satisfying $2 \leq k \leq n$, let A(k) be the be the set of interpolants (to the data) which are in $L_2^{(k)}[a,b]$. We

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

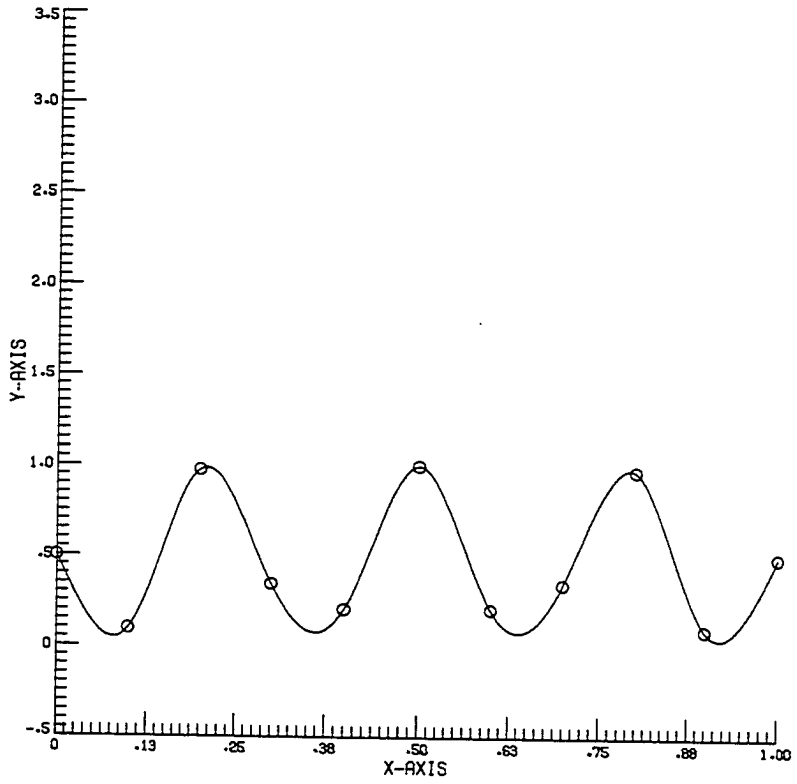

Figure (1.3): The Natural Cubic Spline Interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

consider the problem

$$\text{Find } f_* \in A(k) \text{ such that } \|f_*^{(k)}\|_2 \leq \|f^{(k)}\|_2 \text{ for all } f \in A(k) \tag{1.19}$$

Let f be an element of A(k). Since (1.3) is valid for f, we can integrate by parts again (assuming $k > 2$) to obtain

$$f(t) = q_2(t) + \int_a^t \frac{(t-s)^2}{2!} f^{(3)}(s)ds \tag{1.20}$$

where

$$q_2(t) = f(a) + f^{(1)}(a)(t-a) + \frac{f^{(2)}(a)}{2!}(t-a)^2.$$

In general, after integrating by parts k-1 times we obtain

$$f(t) = q_{k-1}(t) + \int_a^b \frac{(t-s)^{k-1}}{(k-1)!} f^{(k)}(s)ds \tag{1.21}$$

or

$$f(t) = q_{k-1}(t) + \int_a^b \frac{(t-s)_+^{k-1}}{(k-1)!} f^{(k)}(s)ds. \tag{1.22}$$

Now we take the (scaled) k-th divided difference of (1.22) to obtain

$$d_{i,k} = \int_a^b g(s)N_{i,k}(s)ds \qquad i=1,2,\ldots,n-1 \tag{1.23}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where

$$d_{i,k} = (k-1)!(t_{i+k}-t_i)[t_i,\ldots,t_{i+k}]f(\cdot) , \qquad (1.24)$$

$$N_{i,k}(s) = (t_{i+k}-t_i)[t_i,\ldots,t_{i+k}](\cdot-s)_+^{k-1} \qquad (1.25)$$

(the normalized B-spline of order k), and $g = f^{(2)}$.

Let B(k) denote the set of elements (in $L_2[a,b]$) wich satisfy (1.23). Then the solution f_* to (1.20) is related to the solution to the problem

$$\text{Find } g_* \in B(k) \text{ such that } \|g_*^{(k)}\|_2 \leq \|g^{(k)}\|_2 \text{ for all } g \in B(k) \qquad (1.26)$$

via $g_* = f_*^{(k)}$. Furthermore, for some $\underline{\alpha} \in R^{n-k}$ we have

$$g_* = \sum_{j=1}^{n-k} \alpha_j N_{j,k}.$$

The coefficients $\alpha_1, \alpha_2, \ldots, \alpha_{n-k}$ are chosen to solve the linear system of n-k equations in n-k unknowns represented by the matrix equation $A\,\underline{\alpha} = \underline{d}$ where A is symmetric and positive definite with entries

$$A_{ij} = (N_{i,k}, N_{j,k}).$$

Since g_* is a linear combination of piecewise polynomials of order k, f_* will be a piecewise polynomial of order 2k. We call f_* the natural spline interpolant of order 2k.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2. A Minimal Norm Interpolation Problem in the L_p [a,b] Spaces

For p such that $1 < p \leq \infty$ we define the set

$$G(p) := \left\{ g \in L_p[a,b] : \int_a^b g(t)\phi_i(t)dt = \int_a^b g_o(t)\phi_i(t)dt \right. \\ \left. \text{for } i=1,2,\ldots,n \right\} \tag{2.1}$$

where $\{\phi_i\}_{i=1}^n$ is a set of elements in $L_q[a,b]$, q is conjugate to p ($p+q = pq$ if $p \neq \infty$ and $q=1$ if $p = \infty$), and g_o is a fixed element of $L_p[a,b]$. Consider the problem

$$\text{Find } g_* \in G(p) \text{ such that } \|g_*\|_p \leq \|g\|_p \text{ for all } g \in G(p). \tag{2.2}$$

In chapter 1 we solved (2.2) for the special case p=2; finding from a linear variety in a Hilbert space the element of minimal norm. The Projection Theorem came in handy to characterize g_* as well as to guarantee uniqueness. However, for $p \neq 2$ $L_p[a,b]$ does not have the orthogonality properties of a Hilbert space and hence, we cannot use the Projection Theorem to solve (2.2). Instead we solve (2.2) in this chapter by utilizing the Hahn-Banach theorem to characterize g_*. Uniqueness follows in the case $1 < p < \infty$ by the strict convexity of the norm. This chapter, modeled after [deB(2)], motivates the use of the Hahn-Banach theorem in chapter 5.

Let λ be the linear functional defined on the subspace

$$S := \text{span}(\phi_1,\ldots,\phi_n)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

via

$$\lambda(\sum_{i=1}^{n}\alpha_i\phi_i) = \int_a^b (\sum_{i=1}^{n}\alpha_i\phi_i)(t)g_o(t)dt. \qquad (2.3)$$

Any element of G(p) (including g_o) will serve as an extension of λ to a bounded linear functional defined on all of $L_q[a,b]$. Hence,

$$\|\lambda\|_s \leq \|g\|_p \text{ for all } g\in G(p). \qquad (2.4)$$

Conversely, any extension of λ to a bounded linear functional defined on all of $L_q[a,b]$, being identical to λ on S, is represented by an element of G(p).

The Hahn-Banach theorem guarantees the existence of an element $\hat{g}\in G(p)$ such that

$$\int_a^b f(t)\hat{g}(t)dt \leq \|\lambda\|_s \cdot \|f\|_q \text{ for all } f\in L_q[a,b].$$

This implies that $\|\hat{g}\| \leq \|\lambda\|_s$ which, taken along with (2.4), gives us $\|\hat{g}\| = \|\lambda\|_s$ and, therefore, a solution to (2.2). Now we characterize $\hat{g}$.

Let $\sum_{i=1}^{n}\alpha_i^*\phi_i$ be an element such that

$$\|\sum_{i=1}^{n}\alpha_i\phi_i\|_q = 1 \quad \text{and} \quad \lambda(\sum_{i=1}^{n}\alpha_i\phi_i) = \|\lambda\|_s.$$

(This element is unique if $1 < p < \infty$ since the norm is strictly convex.) Then

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\|\hat{g}\|_p = \|\lambda\|_s$$

$$= \lambda\left(\sum_{i=1}^{n} \alpha_i \phi_i\right)$$

$$= \int_a^b \left(\sum_{i=1}^{n} \alpha_i \phi_i\right)(t)\hat{g}(t)dt$$

$$\leq \left\|\sum_{n=i}^{n} \alpha_i \phi_i\right\| \cdot \|\hat{g}\|_p$$

$$= \|\hat{g}\|_p .$$

Therefore, equality holds throughout and we have

$$\int_a^b \left(\sum_{i=1}^{n} \alpha_i \phi_i\right)(t)\hat{g}(t)dt = \left\|\sum_{i=1}^{n} \alpha_i \phi_i\right\|_q \cdot \|\hat{g}\|_p .$$

Since $\hat{g}$ and $\sum_{i=1}^{n} \alpha_i \phi_i$ are aligned, we must have

$$\hat{g}(t) = \|\lambda\|_s \cdot \left\|\sum_{i=1}^{n} \alpha_i \phi_i\right\|^{q-1} \text{ signum } \left(\sum_{i=1}^{n} \alpha_i \phi_i\right)(t).$$

We close this chapter by stating the interpolation problem that goes along with solving (2.2). Let p be a number such that $1 < p < \infty$, let k be an integer such that $k \geq 2$, and let $f_o \in L_p^{(k)}[a,b]$. Define the sets

$$F := \{f \in L_p^{(k)}[a,b]: f(t_i) = f_o(t_i) \quad i=1,2,\ldots,n\}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and

$$G: = \{g \in L_p[a,b]: \int_a^b g(t)N_{i,k}(t)dt = d_{i,k} \quad i=1,2,\ldots,n-k\}$$

Then the problems

$$\text{Find } f_* \in F \text{ such that } \|f_*^{(k)}\|_p \leq \|f^{(k)}\|_p \text{ for all } f \in F$$

and

$$\text{Find } f_* \in G \text{ such that } \|g_*\|_p \leq \|g\|_p \text{ for all } g \in G$$

are equivalent and

$$g_*(t) = f_*^{(k)}(t) = \left|\sum_{i=1}^{n-k} \beta_i N_{i,k}\right|^{q-1} \text{signum } \left(\sum_{i=1}^{n-k} \beta_i N_{i,k}\right)(t).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3. The Convex Spline Interpolant

The data $\{(t_i,y_i)\}_{i=1}^n$ are called convex if the point (t_{i_2},y_{i_2}) lies on or beneath the line joining the points (t_{i_1},y_{i_1}) and (t_{i_3},y_{i_3}) whenever $1 \leqq i_1 < i_2 < i_3 \leqq n$. Equivalently,

$$[t_{i_1},t_{i_2},t_{i_3}]f(\cdot) = 0$$

(where f is any interpolant to the data) or

$$d_i = \frac{y_{i+2} - y_{i+1}}{t_{i+2} - t_{i+1}} - \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \geqq 0$$

for $i = 1,2,\ldots,m(= n-2)$.

In this chapter we address the problem of finding, for convex data, the smoothest convex interpolant; that is, the convex interpolant having second derivative of minimal norm over all smooth convex interpolants. The natural cubic spline interpolant, the smoothest of all interpolants, regrettably does not always preserve the convexity of the data. In chapter 1 we showed that f_*, the natural cubic spline interpolant, has second derivative

$$f_*^{(2)} = \sum_{j=1}^{m} \alpha_j N_j$$

where the coefficients $\alpha_1,\alpha_2,\ldots,\alpha_m$ satisfy (1.13). If any α_i is negative, then f_* is actually concave on a subset of [a,b].

Let $\{(t_i,y_i)\}_{i=1}^n$ denote convex data and let A denote the set of convex interpolants in $L_2^{(2)}[a,b]$. We assume that A is nonempty.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(There are convex data for which A is empty. For example, let $y_i = |t_i|$ and $t_1 = -2$, $t_2 = -1$, $t_3 = 0$, $t_4 = 1$, and $t_5 = 2$. The only convex interpolant is $f(x) = |x|$, which is not in $L_2^{(2)}[-2,2]$.)

Using the Peano kernel theorem as we did in chapter 1 we can show that if $f \in A$ then $T(f^{(2)}) = \underline{d}$ where $T:L_2[a,b] \to R^m$ is given by $(Tg_i) := (g,N_i)$. Hence if

$$B = \{g \in L_2[a,b]: \quad g \geq 0 \text{ and } Tg = \underline{d}\},$$

then problems

$$\text{Find } f_* \in A \text{ such that } \|f_*^{(2)}\|_2 \leq \|f^{(2)}\|_2 \text{ for all } f \in A \qquad (3.1)$$

and

$$\text{Find } g_* \in B \text{ such that } \|g_*\|_2 \leq \|g\|_2 \text{ for all } g \in B \qquad (3.2)$$

are equivalent and the solutions are related via $g_* = f_*^{(2)}$. Since B is a nonempty closed convex set, we consider (3.2) as finding the distance from a point to a closed convex set in a Hilbert space.

Proposition ([L, page 69]): Let x be an element of a Hilbert space H and let K be a nonempty closed convex subset of H. Then there exists a unique element $k_o \in K$ such that

$$\|x - k_o\| \leq \|x - k\| \quad \text{for all} \quad k \in K$$

Furthermore, k_o is characterized by

$$(x - k_o, k - k_o) \leq 0 \quad \text{for all} \quad k \in K.$$

Since we wish to find the element of minimal norm in B, X corresponds to the zero function and hence g_* is characterized by

$$(g_*, g - g_*) \geq 0 \quad \text{for all} \quad g \in B. \qquad (3.3)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Propostion ([MSSW, proposition 2.1]): If there exist coefficients $\alpha_1,\alpha_2,\ldots,\alpha_m$ satisfying

$$\int_a^b (\sum_{j=1}^m \alpha_j N_j)_+ N_i(t)dt = d_i \quad i=1,2,\ldots,m, \tag{3.4}$$

then $g_* = (\sum_{j=1}^m \alpha_j N_j)_+$. Furthermore, such coefficients exist if there exists $\hat{g} \in B$ such that $\{N_i\}_{i=1}^m$ are linearly independent over the support of $\hat{g}$.

Proof: Assume $\alpha_1,\alpha_2,\ldots,\alpha_m$ satisfy (3.4). Denote $s = \sum_{j=1}^m \alpha_j N_j$ and assume $g \in B$. Define $(h)_- = (-h)_+$ so that

$$h = (h)_+ - (h)_-.$$

Then

$$\begin{aligned}
&((s)_+,\ g-(s)_+) \\
&\quad = (s + (s)_-,\ g - (s)_+) \\
&\quad = (s, g - (s)_+) + ((s)_-, g - (s)_+) \\
&\quad = ((s)_-, g) - ((s)_-, (s)_+) \\
&\quad = ((s)_-, g) \\
&\quad \geqq 0.
\end{aligned}$$

The last inequality is valid since both $(s)_-$ and g are nonnegative functions. Hence $(s)_+$ satisfies (3.3).

We now show that we can find coefficients $\alpha_1,\alpha_2,\ldots,\alpha_m$ so that (3.4) holds by following the procedure employed in [MSSW].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We begin by considering the problem

$$\inf \int_a^b \Big(\sum_{j=1}^m \alpha_j N_j\Big)_+^2(t)dt : \sum \alpha_i d_i = 1 \quad . \tag{3.5}$$

and showing that if the infimum is attained at some $\underline{\alpha}$, then for some positive constant C the coefficients $C\alpha_1$, $C\alpha_2$,..., $C\alpha_m$ satisfy (3.4).

If the infimum of (3.5) is attained at $\underline{\alpha}^*$, then $\underline{\alpha}^*$ is a critical point of the Largrangian

$$L(\underline{\alpha},\lambda) = \int_a^b \Big(\sum_{j=1}^m \alpha_j N_j\Big)_+(t)dt + \lambda\Big(1 - \sum_{j=1}^m \alpha_j d_j\Big). \tag{3.6}$$

At a minimum of L we must have

$$0 = 2\int_a^b \Big(\sum_{j=1}^m \alpha_j N_j\Big)_+ N_i(t)dt - \lambda d_i \quad i=1,2,\ldots,m \tag{3.7}$$

and $\underline{\alpha} \cdot \underline{d} = 1$ for some λ.

Now multiply (3.7) by α_i and sum over i=1,2,...,m to obtain

$$2\int_a^b \Big(\sum_{j=1}^m \alpha_j N_j\Big)_+\Big(\sum_{i=1}^m \alpha_i N_i\Big)(t)dt - \lambda \sum_{i=1}^m \alpha_i d_i = 0$$

or

$$\lambda = 2\int_a^b \Big(\sum_{j=1}^m \alpha_j N_j\Big)_+^2(t)dt \geq 0. \tag{3.8}$$

If $\lambda > 0$, then (3.7) reveals that

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\int_a^b \left(\sum_{j=1}^m \alpha_j^* N_j \right)_+ N_i(t)dt = d_i \quad i=1,2,\ldots,m \tag{3.9}$$

where $\alpha_j^* = 2\alpha_j/\lambda$. If $\lambda = 0$, then (3.8) reveals that

$$\int_a^b \left(\sum_{j=1}^m \alpha_j N_j \right)_+ (t)dt = 0$$

where $\underline{\alpha} \cdot \underline{d} = 1$. This implies that $\left(\sum_{j=1}^m \alpha_j N_j \right) \leq 0$. However, for any $g \in B$ we have

$$\int_a^b \left(\sum_{j=1}^m \alpha_j N_j \right) g(t)dt = \sum_{j=1}^m \alpha_j (N_j, g)$$

$$= \sum_{j=1}^m \alpha_j d_j$$

$$= 1$$

which is impossible because g is nonnegative on [a,b]. We conclude that λ is strictly positive and, if the infimum in (3.5) is attained by some $\underline{\alpha}$, that (3.4) is solvable. We now show that the infimum is attained.

Let $\{\underline{\alpha}^{(k)}\}_{k=1}^{\infty}$ be a minimizing sequence. If $\{\|\underline{\alpha}^{(k)}\|\}_{k=1}^{\infty}$ is unbounded, then divide the objective function of (3.6) by $\|\underline{\alpha}\|_\infty^2$ and the constraint by $\|\underline{\alpha}\|_\infty$. There then exists $\underline{\alpha}$ such that

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\| \underline{\alpha} \|_\infty = 1,$$

$$\underline{\alpha} \cdot \underline{d} = 0, \text{ and}$$

$$\int_a^b \Big(\sum_{j=1}^m \alpha_j N_j \Big)_+^2 (t)dt = 0.$$

We conclude that $\sum_{j=1}^m \alpha_j N_j$ is nonpositive, but not identically zero. Since we have assumed there exists $\hat{g} \in B$ such that the B-splines are linearly independent on the support of $\hat{g}$,

$$0 = \sum_{j=1}^m \alpha_j d_j = \sum_{j=1}^m \alpha_j (\hat{g}, N_j)$$

$$= (\hat{g}, \sum_{j=1}^m \alpha_j N_j)$$

$$< 0$$

which is a contradiction. Hence a minimizing sequence must be bounded and the infimum is attained via a convergent subsequence. This completes the proof of the proposition.

We note that the existence of $\hat{g} \in B$, such that $\{N_i\}_{i=1}^m$ are linearly independent over the support of $\hat{g}$, in the previous proposition is guaranteed if $d_i > 0$ for each i. Then each $g \in B$ must be positive on some subinterval of $[t_i, t_{i+2}]$, the support of N_i, for each i.

Now we consider the implication of allowing $d_k = 0$ for some k. As a specific example let $t_i = (i-1)$ for i=1,2,3,4 and let $\underline{d} = (1,0)^T$. If g_* is the positive part of a linear combination of B-splines, then there must exist numbers α_1 and α_2 satisfying

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\int_0^3 (\alpha_1 N_1 + \alpha_2 N_2)_+ N_i(t)dt = \begin{cases} 1 & \text{if } i=1 \\ 0 & \text{if } i=2 \end{cases} \qquad (3.10)$$

which implies that $\alpha_2 = -\infty$. This is equivalent to the solution being identically zero on [1,3]. In fact, any $g \in B$ must be of the form $g = g\,\chi_{[0,1]}$. It is shown in [MSSW, theorem 3.1] that the solution to (3.2) is

$$g_* = \left(\sum_{j=1}^{m} \alpha_j N_j\right)_+ \chi_\Gamma$$

for appropriate coefficients $\alpha_1, \alpha_2, \ldots, \alpha_m$ where

$$\Gamma: = [a,b]/\{\bigcup_{j=1}^{m} (t_j, t_{j+2}) : d_j = 0\}.$$

Hence the solution to (3.2) with $t_i = (i-1)$ for i=1,2,3,4 and $\underline{d} = (1,0)^T$ is

$$g_* = 3N_1\chi_{[0,1]}.$$

Unless otherwise stated we assume $d_i > 0$ for each i for the remainder of this chapter.

Before we consider how to compute the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_m$ which satisfy (3.4), we give a procedure for integrating g_*. Define β_i, $\Delta\beta_i$, Δt_i, and Δy_i as in chapter 1. We integrate g_* on each subinterval $[t_i, t_{i+1}]$ separately, forming a piecewise polynomial, by solving the differential equation

$$p_{*i}^{(2)}(t) = \left(\beta_i + \frac{\Delta\beta_i}{\Delta t_i}(t-t_i)\right)_+ \qquad (3.11)$$

for $t_i \leq t \leq t_{i+1}$ with boundary conditions $p_*(t_i) = y_i$ and $p_*(t_{i+1}) = y_{i+1}$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Two integrations gives us

$$p_{*i}^{(1)}(t) = \frac{\Delta t_i}{2\Delta\beta_i}\left(\beta_i + \frac{\Delta\beta_i}{\Delta t_i}(t-t_i)\right)_+^2 + c_i \qquad (3.12)$$

and

$$p_{*i}(t) = \frac{\Delta t_i}{6(\Delta\beta_i)^2}\left(\beta_i + \Delta\beta_i(t-t_i)\right)_+^3 + c_i(t-t_i) + e_i \qquad (3.13)$$

for constants c_i and e_i. We proceed by cases.

Case 1 occurs when both β_i and β_{i+1} are nonnegative. The nonnegativity constraint is not active in this case and so (3.13) is equivalent to (1.16), although with modified constants c_i and e_i. The values $p_{*i}^{(j)}(t_i)$ for $j = 0,1,2,3$, are given by (1.18).

Case 2 occurs when $\beta_i < 0$ and $\beta_{i+1} > 0$. In this case p_{*i} can be defined by two polynomials: a linear polynomial q_{i1} defined on $[t_i,\tau_i]$ - where the nonnegativity constraint is active and hence the second derivative is zero - and a cubic polynomial defined on $[\tau_i,t_{i+1}]$ where

$$\tau_i = t_i - \beta_i\Delta t_i/\Delta\beta_i \qquad (3.14)$$

Applying the boundary condition $p_{*i}(t_i) = y_i$ we obtain $e_i = y_i$. Applying $p_{*i}(t_{i+1}) = y_{i+1}$ we get an equation for c_i:

$$\frac{(\Delta t_i)^2}{6(\Delta\beta_i)^2}(\beta_{i+1})^3 + c_i\Delta t_i + y_i = y_{i+1}.$$

Solving for c_i we have

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$c_i = \frac{\Delta y_i}{\Delta t_i} - \frac{(\beta_{i+1})^3 \Delta t_i}{2(\Delta \beta_i)^2} \tag{3.15}$$

From (3.11), (3.12), and (3.13) we obtain

$$\begin{aligned}
&q_{i1}(t_i) = y_i \\
&q_{i1}^{(1)}(t_i) = c_i \\
&q_{i1}^{(2)}(t_i) = 0 \\
&q_{i1}^{(3)}(t_i) = 0 \\
&q_{i2}(\tau_i) = c_i(\tau_i - t_i) + y_i \\
&q_{i2}^{(1)}(\tau_i) = c_i \\
&q_{i2}^{(2)}(\tau_i) = 0 \\
&q_{i2}^{(3)}(\tau_i) = \Delta\beta_i / \Delta t_i
\end{aligned} \tag{3.16}$$

where τ_i and c_i are given by (3.14) and (3.15) respectively.

Case 3 occurs when $\beta_i > 0$ and $\beta_{i+1} < 0$. In this case p_{*i} is defined by a cubic polynomial q_{i1} on $[t_i, \tau_i]$ and by a linear polynomial q_{i2} on $[\tau_i, t_{i+1}]$ with τ_i defined by (3.14). These polynomials are determined by the values

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$q_{i1}(t_i) = y_i$$

$$q_{i1}^{(1)}(t_i) = c_i + (\beta_i)^2\Delta t_i/(2\Delta\beta_i)$$

$$q_{i1}^{(2)}(t_i) = \beta_i \qquad (3.17)$$

$$q_{i1}^{(3)}(t_i) = \Delta\beta_i/\Delta t_i$$

$$q_{i2}(\tau_i) = c_i(\tau_i - t_i) + e_i$$

$$q_{i2}^{(1)}(\tau_i) = c_i$$

$$q_{i2}^{(2)}(\tau_i) = 0$$

$$q_{i2}^{(3)}(\tau_i) = 0$$

where c_i and e_i are given by

$$c_i = \frac{\Delta y_i}{\Delta t_i} - \frac{(\beta_i)^3\Delta t_i}{2(\Delta\beta_i)^2} .$$

and

$$e_i = y_i - \frac{(\beta_i)^3(\Delta t_i)^2}{6(\Delta\beta_i)^2} .$$

Case 4 occurs when β_i and β_{i+1} are both nonpositive. In this case we obtain a linear polynomial defined on $[t_i, t_{i+1}]$ and determined by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$p_{*i}(t_i) = y_i$$

$$p_{*i}^{(1)}(t_i) = \Delta y_i/\Delta t_i \qquad (3.18)$$

$$p_{*i}^{(2)}(t_i) = 0$$

$$p_{*i}^{(3)}(t_i) = 0.$$

Since g_* is piecewise linear and continuous (with knots at the t_i's and τ_i's), f_* will be piecewise cubic with two continuous derivatives (if $d_i > 0$ for each i). We call f_* the convex cubic spline interpolant.

Now we turn our attention to the task of numerically calculating the coefficients $\alpha_1,\alpha_2,\ldots,\alpha_m$ which satisfy (3.4). We continue to assume that $d_i > 0$ for each i. Define $F:R^m \to R^m$ by $F = (F_1,F_2,\ldots,F_m)^T$ where

$$F_i(\underline{\alpha}) = \int_a^b \Big(\sum_{j=1}^{m} \alpha_j N_j\Big)_+ N_i(t)dt \quad i=1,2,\ldots,m. \qquad (3.19)$$

We wish to solve $F(\underline{x}) = \underline{d}$.

One method is to use Jacobi iteration. An initial guess $\underline{x}^{(o)} = (x_1^{(o)}, x_2^{(o)},\ldots, x_m^{(o)})^T$ is chosen and a sequence $\{\underline{x}^{(k)}\}_{k=o}^{\infty}$ is generated by calculating $\underline{x}^{(k+1)}$, once $\underline{x}^{(k)}$ is known, by solving

$$F_i(x_1^{(k)},\ldots,x_{i-1}^{(k)},x_i^{(k+1)},x_{i+1}^{(k)},\ldots,x_m^{(k)}) = d_i$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

for $x_i^{(k+1)}$ for each i. A modification, the Gauss-Seidel iteration, involves calculating $\underline{x}^{(k+1)}$, once $\underline{x}^{(k)}$ is known, by solving

$$F_i(x_1^{(k+1)},\ldots,x_{i-1}^{(k+1)},x_i^{(k+1)},x_{i+1}^{(k)},\ldots,x_m^{(k)}) = d_i$$

for $x_i^{(k+1)}$ for i=1,2,...,m in succession. Both Jacobi and Gauss-Siedel iterations converge globally as proved in [IMS]

Now we consider Newton's method to solve $G(\underline{x}) = F(\underline{x}) - \underline{d} = \theta$. We pick a suitable initial guess $\underline{x}^{(0)}$ and form a sequence $\{\underline{x}^{(k)}\}_{k=0}^{\infty}$ by solving

$$(\nabla G)(\underline{x}^{(k)})(\underline{x}^{(k+1)} - \underline{x}^{(k)}) = -G(\underline{x}^{(k)}) \tag{3.20}$$

for $\underline{x}^{(k+1)}$ once $\underline{x}^{(k)}$ is known. Since $\nabla G = \nabla F$, we can express (3.20) alternately as

$$(\nabla F)(\underline{x}^{(k)})(\underline{x}^{(k+1)}) - \underline{x}^{(k)}) = \underline{d} - F(\underline{x}^{(k)}). \tag{3.21}$$

The entries of the Jabocian matrix F are

$$(\nabla F)_{ij}(\underline{\alpha}) = \int_a^b (\sum_{k=1}^{m} \alpha_k N_k)_+^o N_j N_i(t)dt \tag{3.22}$$

where $(\sum_{k=1}^{m} \alpha_k N_k)_+^o$ is the characteristic function for the support of $(\sum_{j=1}^{m} \alpha_j N_j)_+$. We see that ∇F is symmetric and tridiagonal at each $\underline{\alpha}$. We now characterize those $\underline{\alpha}$ for which $(\nabla F)(\underline{\alpha})$ is positive definite.

Lemma (3.1): The Jacobian $(\nabla f)(\underline{\alpha})$ is positive definite if and only if

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$(\sum_{k=1}^{m} \alpha_k N_k)_+$ does not vanish identically on any of the subintervals $[t_i, t_{i+2}]$ for $i=1,2,\ldots,m$.

Proof: For any $\underline{x} \in K^m$ we have

$$\underline{x}^T(\nabla F)(\underline{\alpha})\underline{x} = \sum_{i=1}^{m} x_i \sum_{j=1}^{m} (\nabla F)_{ij}(\underline{\alpha})x_j$$

$$= \int_a^b (\sum_{k=1}^{m} \alpha_k N_k)_+^o (\sum_{j=1}^{m} x_j N_j)(\sum_{i=1}^{m} x_i N_i)(t)dt$$

$$= \int_a^b (\sum_{k=1}^{m} \alpha_k N_k)_+^o (\sum_{i=1}^{m} x_i N_i)^2(t)dt$$

$$\geq 0$$

If $(\sum_{j=1} \alpha_j N_j)_+$ does not vanish identically on $[t_i, t_{i+2}]$ for each i, then equality holds if and only if $x_i = 0$ for each i. If there exists some k such that $(\sum_{j=1}^{m} \alpha_j N_j)_+$ is identically zero on $[t_k, t_{k+2}]$, then equality does hold for the nonzero vector $\underline{x}$ defined by $x_i = \delta_{ik}$ for each i. This completes the proof of the lemma.

From (3.20) we see that

$$F_i(\underline{\alpha}) = \sum_{j=1}^{m} \alpha_j \int_a^b (\sum_{k=1}^{m} \alpha_k N_k)_+^o N_j N_i(t)dt$$

$$= \sum_{j=1}^{m} \alpha_j (\nabla F)_{ij}(\underline{a})$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

so that $F(\underline{\alpha}) = (\nabla F)(\underline{\alpha})\,\underline{\alpha}$. Newton's method - equation (3.22) - takes the form

$$(\nabla F)(\underline{x}^{(k)})\underline{x}^{(k+1)} = \underline{d}. \tag{3.23}$$

Theorem (3.2): If $(\nabla F)(\underline{x}^{(k)})$ is positive definite, then $(\nabla F)(\underline{x}^{(k+1)})$ is positive definite for each k and, hence, Newton's method - equation 3.23) - is always well-defined.

Proof: Having the known values $x_i^{(k)}$, we wish to determine the values $x_i^{(k+1)}$ satisfying

$$\int_{S(k)} \Big(\sum_{j=1}^{m} x_j^{(k+1)} N_j\Big) N_i(t)\,dt = d_i \qquad i=1,2,\ldots,m \tag{3.24}$$

where S(k) is the support of $\Big(\sum_{j=1}^{m} x_j^{(k)} N_j\Big)_+$. Since $(\nabla F)(\underline{x}^{(k)})$ is positive definite, then $S(k) \cup [t_i, t_{i+2}]$ contains an interval for each i. Since $d_i > 0$, then $\Big(\sum_{j=1}^{m} x_j^{(k+1)} N_j\Big)_+$ is positive on some subinterval of $[t_i, t_{i+2}]$. Hence, $(\nabla F)(\underline{x}^{(k+1)})$ is positive definite. This completes the proof of the Theorem.

Note that if $\underline{x}^{(0)}$ has all positive components (for example, if $x_i^{(0)} = 1$ for each i, then $S(0) = [a,b]$ and $\sum_{j=1}^{m} x_j^{(1)} N_j$ is the second derivative of the natural cubic spline interpolant.

Now we assume that $d_k = 0$ for some k. In this case special care must be exercised since $\{x_k^{(j)}\}_{j=0}^{\infty}$ may diverge to $-\infty$, preventing any numerical convergence. We already know that $d_k = 0$ implies that the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

data points (t_k,y_k), (t_{k+1},y_{k+1}), and (t_{k+2},y_{k+2}) are collinear and, hence, any convex interpolant must be linear on $[t_k,t_{k+2}]$. Equivalently, the second derivative of any convex interpolant must be zero on $[t_k,t_{k+2}]$. Hence g_* is of the form

$$\left(\sum_{j=1}^{m} x_j N_j\right)_+ \{\chi_{[a,t_k]} + \chi_{[t_{k+2},b]}\}.$$

Since the value of x_k is immaterial and the k-th equation is automatically satisfied, the number of equations and unknowns each reduce by one. For computational convencience (3.23) can still be used with the following modifications: $(\nabla F)_{kk} = 1$, $(\nabla F)_{k,k+1} = 0$, and $(\nabla F)_{k,k-1} = 0$.

If $d_k = 0$, then the solution is discontinuous at t_k if $x_{k-1} > 0$ and is discontinuous at t_{k+2} if $x_{k+1} > 0$. If the solution is discontinuous, then f_* will have only one continuous derivative.

A further problem is encountered when d_{k-1} and d_{k+1} are both zero, but d_k is nonzero for some k. Any nonnegative function g which satisfies the (k-1)-st and (k+1)-st equations can not satisfy the k-th equation since g is identically zero on $[t_{k-1},t_{k+1}]$ and on $(t_{k+1},t_{k+2}]$. We conclude that there does not exist any convex interpolant in $L_2^{(2)}[a,b]$ (and no solution to the problem as posed). However, we can find a convex interpolant whose second derivative is of the form

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\left(\sum_{j=1}^{m} x_j N_j\right)_+ \{ \chi_{[a,t_{k-1}]} + \chi_{[t_{k+3},b]} \}$$

satisfying all but the k-th equation. We already know that this convex interpolant must be linear on $[t_{k-1},t_{k+1}]$ and on $[t_{k+1},t_{k+3}]$ and, hence, piecewise linear on $[t_{k-1},t_{k+3}]$. If d_k is nonzero, then there will be a discontinuity in slope at t_{k+1}. For the convenience of utilizing (3.23) we can set d_k to be zero to satisfy the k-th equation. The discontinuity in slope will show up after we integrate the solution to obtain the interpolant.

Figure (3.1) displays the natural cubic spline interpolant to the function

$$f(t) = \frac{1}{(0.05+t)(1.05-t)}$$

at the knots $t_1 = 0$, $t_2 = 0.1$, $t_3 = 0.4$, $t_4 = 0.7$, $t_5 = 0.8$, and $t_6 = 1.0$. Figure (3.2) displays the convex spline interpolant to this function. Table (3.1) shows the convergence results for Jacobi, Gauss-Seidel, and Newton's method iterations taken from [IMS]. Note the quadratic convergence characteristic of Newton's method. These convergence results are typical.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

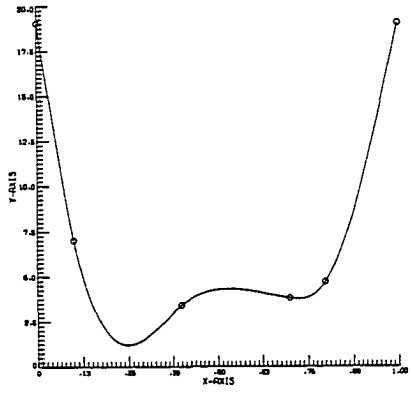

Figure (3.1): The Natural Cubic Spline Interpolant.

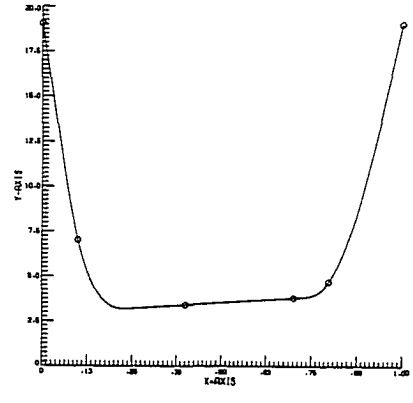

Figure (3.2): The Convex Cubic Spline Interpolant

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE 3.1

$$\| F(x^{(n)}) - d \|_2$$

Iteration Number	Jacobi	Gauss Seidel	Newton
1	$.46 \times 10^{2}$	$.27 \times 10^{2}$	$.19 \times 10^{2}$
2	$.28 \times 10^{2}$	$.11 \times 10^{2}$	$.85 \times 60^{1}$
3	$.75 \times 10^{1}$	$.42 \times 10^{1}$	$,29 \times 10^{1}$
4	$.12 \times 10^{2}$	$.18 \times 10^{1}$	$.49 \times 10^{0}$
5	$.26 \times 10^{1}$	$.75 \times 10^{0}$	$.14 \times 10^{-1}$
6	$.49 \times 10^{1}$	$.31 \times 10^{0}$	$.11 \times 10^{-4}$
7	$.10 \times 10^{1}$	$.13 \times 10^{0}$	$.71 \times 10^{-11}$
8	$.21 \times 10^{1}$	$.55 \times 10^{-1}$	$.49 \times 10^{-12}$
9	$.43 \times 10^{0}$	$.23 \times 10^{-1}$	-----
10	$.86 \times 10^{0}$	$.96 \times 10^{-2}$	-----
20	$.11 \times 10^{-1}$	$.16 \times 10^{-5}$	-----
30	$.14 \times 10^{-3}$	$.26 \times 10^{-9}$	-----
40	$.18 \times 10^{-5}$	$.58 \times 10^{-13}$	-----
50	$.24 \times 10^{-7}$	-----	-----
60	$.30 \times 10^{-9}$	-----	-----
70	$.39 \times 10^{-11}$	-----	-----

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4. The Shape-Preserving Spline Interpolant

We addressed in chapter 3 the problem of finding, for convex data, the smoothest convex interpolant. We begin this chapter by considering the problem of finding, for concave data, the smoothest concave interpolant. Then we continue the chapter by examining the problem of finding, for general data, the smoothest interpolant which is locally convex where the data are locally convex and is locally concave where the data are locally concave.

Let $\{(t_i,y_i)\}_{i=1}^n$ denote concave data and let A denote the set of all concave interpolants in $L_2^{(2)}[a,b]$. Assume A is nonempty. Using the Peano kernel theorem as we did in chapter 1, we see that, if $f \in A$, then

$$\int_a^b f^{(2)}(t)N_i(t)dt = d_i \qquad i=1,2,\ldots,m(=n-2)$$

Equivalently, we have $T(f^{(2)}) = \underline{d}$.

Defining

$$B: = \{g \in L_2[a,b] : g \leq 0 \text{ and } Tg = \underline{d}\} ,$$

we conclude that the problems

$$\text{Find } f_* \in A \text{ such that } \|f_*^{(2)}\|_2 \leq \|f^{(2)}\|_2 \text{ for all } f \in A \quad (4.1)$$

(the problem of finding the smoothest concave interpolant) and

$$\text{Find } g_* \in B \text{ such that } \|g_*\|_2 \leq \|g\|_2 \text{ for all } g \in B$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

are equivalent and the solutions are related via $g_* = f_*^{(2)}$.

Of course, the smoothest concave interpolant to the concave data $\{(t_i,y_i)\}_{i=1}^n$ is the negative of the smoothest convex interpolant to the convex data $\{(t_i,-y_i)\}_{i=1}^n$. We highlight this with the following proposition.

Proposition [MSSW]: If there exist coefficients $\alpha_1,\alpha_2,\alpha_3,\ldots\alpha_m$ satisfying

$$\int_a^b -\Big(\sum_{j=1}^m \alpha_j N_j\Big)_- N_i(t)dt = d_i \quad i=1,2,\ldots,m \tag{4.3}$$

then $g_* = -\Big(\sum_{j=1}^m \alpha_j N_j\Big)_-$. Furthermore, such coefficients exist if there exists $\hat{g} \in B$ such that $\{N_i\}_{i=1}^m$ are linearly independent over the support of $\hat{g}$.

We note that the existence of $\hat{g} \in B$, such that $\{N_i\}_{i=1}^m$ are linearly independent over the support of $\hat{g}$, in the previous proposition is guaranteed if $d_i < 0$ for each i. Then each $g \in B$ is negative on some subinterval of $[t_i,t_{i+2}]$, the support for N_i, for each i.

Now we consider the problem of finding, for general data, a smooth shape-preserving interpolant - a smooth interpolant which is locally convex where the data are locally convex and is locally concave where the data are locally concave. Assuming for the moment that d_i is nonzero for each i, we define the sets

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$T_1 := \{[t_i, t_{i+2}] : d_i > 0\} ,$$

$$T_2 := \{[t_i, t_{i+2}] : d_i > 0\} ,$$

$$\Omega_1 := T_1 / T_2 ,$$

$$\Omega_2 := T_2 / T_1 ,$$

and

$$\Omega_3 := [a,b] / (\Omega_1 \cup \Omega_2).$$

Now we define the sets

$$A := \{f \in L_2^{(2)}[a,b] : f^{(2)} \chi_{\Omega_1} \geq 0 , \ f^{(2)} \chi_{\Omega_2} \leq 0,$$

$$\text{and} \quad f(t_i) = y_i \quad i=1,2,\ldots,n\}$$

(which we assume is nonempty) and

$$B := \{g \in L_2[a,b] : g\chi_{\Omega_1} \geq 0 , \ g\chi_{\Omega_2} \leq 0, \quad \text{and} \quad Tg = \underline{d}\} .$$

We conclude that the problems

$$\text{Find } f_* \in A \text{ such that } \|f_*^{(2)}\|_2 \leq \|f^{(2)}\|_2 \text{ for all } f \in A \qquad (4.4)$$

and

$$\text{Find } g_* \in B \text{ such that } \|g_*\|_2 \leq \|g\|_2 \text{ for all } g \in B \qquad (4.5)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

are equivalent and $g_* = f_*^{(2)}$.

The following proposition gives the solution to (4.5). We see that $f_*\chi_{\Omega_1}$ has the character of the convex spline interpolant, $f_*\chi_{\Omega_2}$ has the character of the concave spline interpolant, and $f_*\chi_{\Omega_3}$ has the character of the natural spline interpolant.

Proposition [MSSW]: If there exists coefficents $\alpha_1, \alpha_2, \ldots, \alpha_m$ satisfying

$$\int_a^b \{(\sum_{j=1}^m \alpha_j N_j)_+ \chi_{\Omega_1} - (\sum_{j=1}^m \alpha_j N_j)_- \chi_{\Omega_2} + (\sum_{j=1}^m \alpha_j N_j)\chi_{\Omega_3}\} N_i(t)dt = d_i \quad i=1,2,\ldots,m \tag{4.6}$$

then

$$g_* = (\sum_{j=1}^m \alpha_j N_j)_+ \chi_{\Omega_1} - (\sum_{j=1}^m \alpha_j N_j)_- \chi_{\Omega_2} + (\sum_{j=1}^m \alpha_j N_j)\chi_{\Omega_3}.$$

Furthermore, such coefficients exist if there exists $\hat{g} \in B$ such that $\{N_i\}_{i=1}^m$ are linearly independent over the support of $\hat{g}$.

We note that the existence of $\hat{g} \in B$, such that $\{N\}_{i=1}^m$ are linearly independent over the support of $\hat{g}$, in the previous proposition is guaranteed if d_i is nonzero for each i. Then each $g \in B$ is nonzero on

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

some subinterval of $[t_i,t_{i+2}]$, the support of N_i, for each i.

We now solve (4.6). Define $F : R^m \to R^m$ where $F = (F_1,F_2,\ldots,F_m)^T$ is given

$$F_i(\underline{x}) = \int_{\Omega_1} (\sum_{j=1}^{m} x_j N_j) + N_i(t)dt$$

$$= \int_{\Omega_2} (\sum_{j=1}^{m} x_j N_j) - N_i(t)dt$$

$$+ \int_{\Omega_3} (\sum_{j=1}^{m} x_j N_j) N_i(t)dt \qquad i=1,2,\ldots,m \tag{4.7}$$

We use Newton's method to solve $F(\underline{\alpha}) = \underline{d}$. Picking a suitable initial guess $\underline{x}^{(0)}$ we produce a sequence $\{\underline{x}^{(0)}, \underline{x}^{(1)},\ldots,\}$ by solving

$$(\nabla F)(\underline{x}^{(k)})(\underline{x}^{(k+1)} - \underline{x}^{(k)}) = \underline{d} - F(\underline{x}^{(k)}) \tag{4.8}$$

for $\underline{x}^{(k+1)}$ once $\underline{x}^{(k)}$ is known. The Jacobian matrix has entries given by

$$(\nabla F)_{ij}(\underline{\alpha}) = \int_a^b P(\underline{\alpha})N_j(t)N_i(t)dt \tag{4.9}$$

where

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$P(\underline{\alpha}) = \Big(\sum_{j=1}^{m} \alpha_j N_j\Big)^{\circ}_{+}\chi_{\Omega_1} + \Big(\sum_{j=1}^{m} \alpha_j N_j\Big)^{\circ}_{-}\chi_{\Omega_2} + \chi_{\Omega_3}. \tag{4.10}$$

From (4.9) we see that ∇F is symmetric and tridiagonal at each $\underline{\alpha}$. We also note that

$$P(\underline{x})\Big(\sum_{j=1}^{m} x_j N_j\Big) = \Big(\sum_{j=1}^{m} x_j N_j\Big)_{+}\chi_{\Omega_1}$$

$$- \Big(\sum_{j=1}^{m} x_j N_j\Big)_{-}\chi_{\Omega_2}$$

$$+ \Big(\sum_{j=1}^{m} x_j N_j\Big)\chi_{\Omega_3}$$

so that $F(\underline{x}) = (\nabla F)(\underline{x})\underline{x}$ and, hence, (4.8) reduces to

$$(\nabla F)(\underline{x}^{(k)})\underline{x}^{(k+1)} = \underline{d}. \tag{4.11}$$

The following lemma (with proof similar to its counterpart in chapter 3) characterizes those $\underline{\alpha}$ for which $(\nabla F)(\underline{\alpha})$ is positive definite.

Lemma(4.1): The Jacobian $(\nabla F)(\underline{\alpha})$ is positive definite if and only if $P(\underline{\alpha})$ does not vanish identically on any of the subintervals $[t_i, t_{i+2}]$ for $i=1,2,\ldots,m$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The following theorem is modeled after theorem (3.2).

Theorem (4.2): If $(\nabla F)(\underline{x}^{(0)})$ is positive definite, then Newton's method - equation (4.10) - is always well-defined.

Note that if $\underline{x}^{(0)}$ is given by $x_i^{(0)} = \text{signum}\,(d_i)$ for each i, then $P(\underline{x}^{(0)})$ is the characteristic function for the interval [a,b] and $\sum_{j=1}^{m} x_j^{(1)} N_j$ is the second derivative of the natural cubic spline interpolant.

If $d_k = 0$ for some k, then we already know that any shape-preserving interpolant must be linear on $[t_k, t_{k+2}]$. In fact any $g \in B$ must satisfy

$$g = g\{\chi_{[a,t_k]} + \chi_{[t_{k+2},b]}\}.$$

The solution in this case is of the form

$$g_* = h\{\chi_{[a,t_k]} + \chi_{[t_{k+2},b]}\}$$

where

$$h = (\sum_{j=1}^{m} \alpha_j N_j)_+ \chi_{\Omega_1} - (\sum_{j=1}^{m} \alpha_j N_j)_- \chi_{\Omega_2} + (\sum_{j=1}^{m} \alpha_j N_j)\chi_{\Omega_3}.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Since the value of α_k is immaterial - the k-th equation $F_k(\underline{\alpha}) = d_k$ of (4.11) being automatically satisfied - the number of equations and unknowns reduce by one each. For computational convenience we can still use (4.10) by setting $(\nabla F)_{kk} = 1$, $(\nabla F)_{k,k+1} = 0$, and $(\nabla F)_{k,k-1}=0$.

Once we solve $F(\underline{\alpha}) = \underline{d}$ we proceed to integrate g_* which is piecewise linear (but not necessarily continuous, even if d_k is non-zero for each k) to obtain f_* which is piecewise cubic. On the interval $[t_i,t_{i+1}]$ f_* is given by the solution to the differential equation

$$p_i^{(2)}(t) = \beta_i + (\Delta\beta_i/\Delta t_i)(t-t_i) \tag{4.12}$$

for $t_i \leqq t \leqq t_{i+1}$ if $[t_i,t_{i+1}] \subset \Omega_3$,

$$p_i^{(2)}(t) = (\beta_i + (\Delta\beta_i/\Delta t_i)(t-t_i))_+ \tag{4.13}$$

for $t_i \leqq t \leqq t_{i+1}$ if $[t_i,t_{i+1}] \subset \Omega_1$, or

$$p_i^{(2)}(t) = -(\beta_i + (\Delta\beta_i/\Delta t_i)(t-t_i))_- \tag{4.14}$$

for $t_i \leqq t \leqq t_{i+1}$ if $[t_i,t_{i+1}] \subset \Omega_2$ with boundary conditions

$$p_i(t_i) = y_i \quad \text{and} \quad p_i(t_{i+1}) = y_{i+1}.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The function p_i is either a cubic polynomial or piecewise cubic given by two polynomials q_{i1} and q_{i2} defined on separate subintervals of $[t_i,t_{i+1}]$. The solution p_i to (4.11) is given by (1.18). The solution to (4.12) is, depending on signum (β_i) and signum (β_{i+1}), given by (1.18), (3.16), (3.17), and (3.18). The solution to (4.13) is determined by (1.18) if $\beta_i \leq 0$ and $\beta_{i+1} \leq 0$, by (3.16) if $\beta_i > 0$ and $\beta_{i+1} < 0$, by (3.17) if $\beta_i < 0$ and $\beta_{i+1} > 0$, and by (3.18) if $\beta_i \geq 0$ and $\beta_{i+1} \geq 0$.

Figures (4.1), (4.3), (4.5) and (4.7) display the natural cubic spline interpolants to the given data. Figures (4.2), (4.4), (4.6), and (4.8) display the corresponding shape-preserving interpolants. Tables (4.1), (4.2), (4.3), and (4.4) give convergence results for Newton's method. Note the quadratic convergence characteristic of Newton's method.

Appendix B lists a FORTRAN program for computing the shape-preserving cubic spline interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

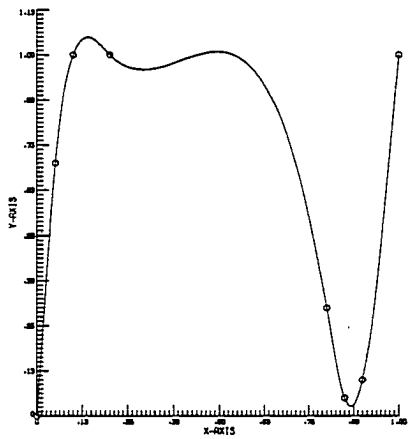

Figure (4.1): The Natural Cubic Spline Interpolant.

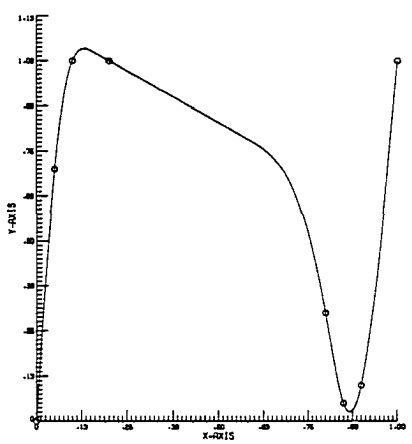

Figure (4.2): The Shape-Preserving Cubic Spline Interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

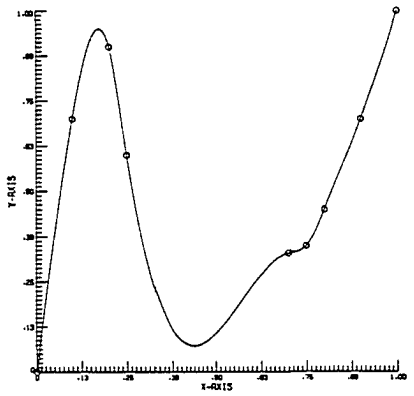

Figure:(4.3): The Natural Cubic Spline Interpolant.

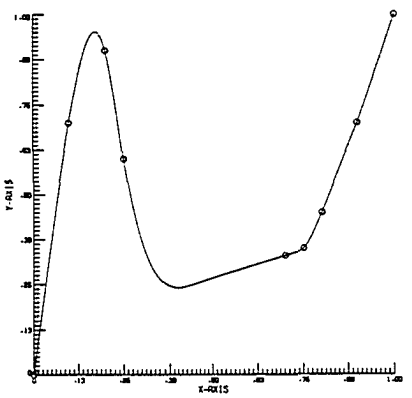

Figure (4.4): The Shape-Preserving Cubic Spline Interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

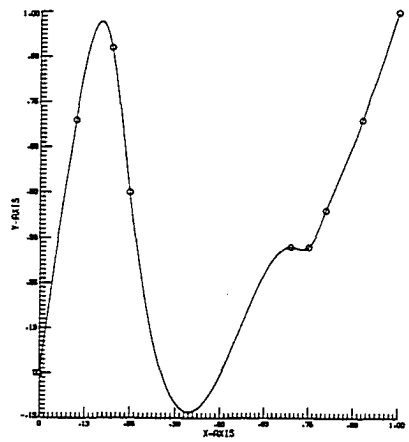

Figure (4.5): The Natural Cubic Spline Interpolant.

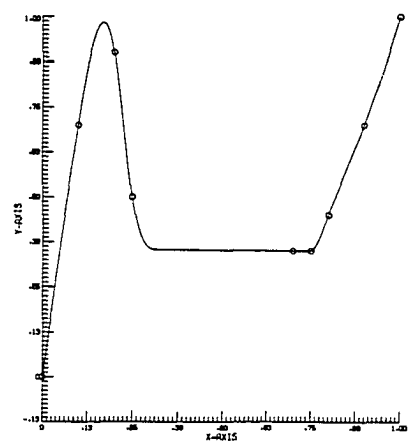

Figure (4.6): The Shape-Preserving Cubic Spline Interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

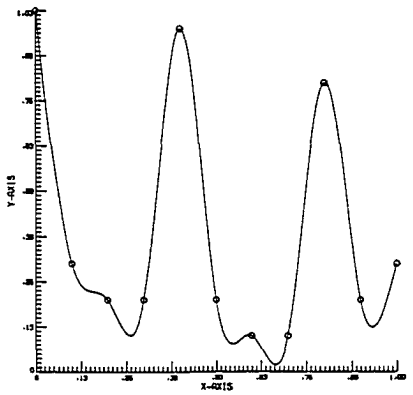

Figure (4.7): The Natural Cubic Spline Interpolant.

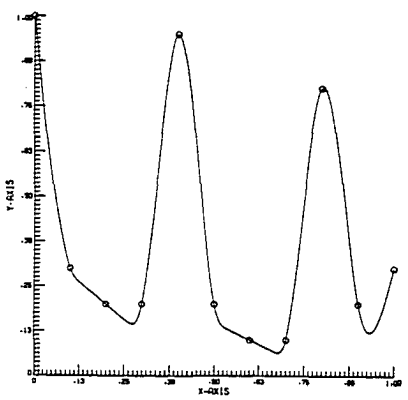

Figure (4.8): The Shape-Preserving Cubic Spline Interpolant.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.1

Iteration Number	$\|F(\underline{x}^{(n)}) - \underline{d}\|_2$
1	0.13×10^{1}
2	0.67×10^{0}
3	0.25×10^{0}
4	0.42×10^{-1}
5	0.12×10^{-2}
6	0.88×10^{-6}
7	0.58×10^{-12}
8	0.64×10^{-13}

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.2

Iteration Number	$\|F(\underline{x}^{(n)}) - \underline{d}\|_2$
1	0.12×10^{1}
2	0.56×10^{0}
3	0121×10^{0}
4	0.36×10^{-1}
5	0.11×10^{-2}
6	0.85×10^{-6}
7	0.54×10^{-12}
8	0.70×10^{-13}

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.3

Iteration Number	$\|F(\underline{x}^{(n)}) - \underline{d}\|_2$
1	0.24×10^{1}
2	0.16×10^{1}
3	0.12×10^{1}
4	0.90×10^{0}
5	0.53×10^{0}
6	0.20×10^{0}
7	0.26×10^{-1}
8	0.42×10^{-3}
9	0.97×10^{-7}
10	0.37×10^{-12}
11	0.21×10^{-12}

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.4

Iteration Number	$\|F(\underline{x}^{(n)} - \underline{d}\|_2$
1	0.29×10^{1}
2	0.13×10^{1}
3	0.50×10^{0}
4	0.11×10^{0}
5	0.56×10^{-2}
6	0.16×10^{-4}
7	0.13×10^{-9}
8	0.26×10^{-12}

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5. Constrained Minimization in a Dual Space

Let C be a convex cone in a normed dual space X with predual Y. Assume $y_1, y_2, \ldots, y_n$ are elements of Y and define $T: X \to R^n$ by

$$Tx = (x(y_1), x(y_2), \ldots, x(y_n))^T$$

Let $B := \{x \in C : Tx = \underline{d}\}$ for a given vector $\underline{d}$. Consider the problem

$$\text{Find } x_* \in B \text{ such that } \|x_*\| \leq \|x\| \text{ for all } x \in B \qquad (5.1)$$

of which (1.10), (3.2), and (4.5) are special cases. In this chapter we study existence and characterization of solutions to (5.1). The following lemma gives sufficient conditions for existence of a solution.

Lemma(5.1): If B is nonempty, if C is weak* closed, and if Y is separable, then there exists a solution to problem (5.1).

Proof: Let $\gamma := \inf \{\|x\| : x \in C \text{ and } Tx = \underline{d}\}$. Let $\{x_n\}$ be a sequence in C such that

$$Tx_n = \underline{d} \qquad (5.2)$$

and

$$\|x_n\| \leq \gamma + 1/n \qquad (5.3)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

for each n. Since Y is separable, by Alaoglu's theorem there exists a weak* convergent subsequence of $\{x_n\}$ with weak* limit x. Since C is weak* closed we have $x \in C$, from (5.2) we have $Tx = \underline{d}$, and from (5.3) we have $\|x\| \leq \gamma$ (and hence $\|x\| = \gamma$). This completes the proof of the lemma.

Throughout this chapter we assume that B is nonempty, C is weak* closed, and Y is separable. Since $x_* = \theta$ if $\underline{d} = \theta$, we assume also that $\underline{d} \neq \theta$. The following proposition gives us sufficient conditions for C being weak* closed.

Proposition (5.2): If C is normed closed and if Y is a reflexive space, then C is weak* closed.

Proof: Assume $\{x_n\}$ is a sequence in C with weak* limit x. We want to show that x is in C. We do this by contradiction. If x is not an element of C, then there exists an element y (an element of both the dual and predual of X) which serves to separate x from C in the sense that

$$x_n(y) > K$$

for each n and

$$x(y) < K$$

for some constant K. This implies that

$$\lim_{n \to \infty} x_n(y) \neq x(y)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

which is a contradiction. Therefore $x \in C$ and C is weak* closed. This completes the proof of the proposition.

For $\gamma > 0$ we define the convex set $G(\gamma) \subset R^n$ by

$$G(\gamma) := \{Tx : x \in C \text{ and } \|x\| \leq \gamma\}.$$

We now show that $G(\gamma) = \gamma G(1)$ and $G(\gamma)$ is closed.

Proposition (5.3): For each $\gamma > 0$ we have $G(\gamma) = \gamma G(1)$.

Proof: By definition

$$\begin{aligned} G(\gamma) &= \{Tx : x \in C \text{ and } \|x\| \leq \gamma\} \\ &= \{Tx : \tfrac{x}{\gamma} \in C \text{ and } \|x/\gamma\| \leq 1\} \\ &= \{T(x/\gamma) : \tfrac{x}{\gamma} \in C \text{ and } \|x/\gamma\| \leq 1\} \\ &= \gamma\{Tw : w \in C \text{ and } \|w\| \leq 1\} \\ &= \gamma G(1). \end{aligned}$$

Proposition (5.4): The set G(1) is closed.

Proof: Assume $\{z_n\}$ is a sequence in G(1) which converges to z. We want to show that z is an element of G(1). Equivalently, we want to show that $x \in C$ exists such that $\|x\| \leq 1$ and $Tx = z$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

For each n there exists $x_n \in C$ such that $\|x_n\| \leq 1$ and $Tx_n = \underline{z}_n$. By Alaoglu's theorem there exists a subsequence of $\|x_n\|$ which converges weak* to some $x \in C$. Hence $\|x\| \leq 1$ and $Tx = \underline{z}$. This completes the proof of the proposition.

We define

$$\gamma^* := \inf\{\gamma : \underline{d} \in G(\gamma)\} . \tag{5.4}$$

Equivalently,

$$\gamma^* = \inf\{\gamma : \text{There exists } x \in C \text{ such that } Tx = \underline{d} \text{ and } \|x\| \leq \gamma\}$$

$$= \inf\{\|x\| : x \in C \text{ and } Tx = \underline{d}\} . \tag{5.5}$$

By lemma (5.1) we know that there exists $x_* \in C$ such that $\|x_*\| = \gamma^*$ and $Tx_* = \underline{d}$. We call x_* an interpolant of minimal norm. We now attempt to characterize x_* via the Hahn-Banach theorem.

We begin by defining a functional $\rho : Y \to R$ by

$$\rho(y) = \sup\{x(y) : x \in C \text{ and } \|x\| \leq 1\} .$$

Notice that if $C = X$ (the unconstrained problem), then ρ is the norm on Y. In general, since we are taking the supremum over a subset of the closed unit ball U in X, we have $\rho(y) \leq \|y\|$ for all $y \in Y$. Since θ is an element of C, we have $\rho \geq 0$. In convex analysis ρ is called the support functional of the convex set $\{x \in C : \|x\| \leq 1\}$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Since C is weak* closed, the supremum is attained at some element of $\{x \in C : \|x\| \leq 1\}$; that is, for any $y \in Y$ there exists an x (a function of y) such that $x \in C$, $\|x\| \leq 1$, and $\rho(y) = x(y)$. In fact we have $\|x\| = 1$ unless $x = \theta$. The following two propositions reveal that ρ is continuous, subadditive, and positive homogeneous.

Lemma(5.5): The functional ρ is continuous.

Proof: Assume y_1 and y_2 are elements of Y and define $y = y_1 - y_2$. Let x be the element in $\{x \in C : \|x\| \leq 1\}$ such that $\rho(y_2) = x(y_2)$. Since $|x(y)| \leq \|y\|$, we have

$$x(y_2) - \|y\| \leq x(y_2) + x(y)$$

or

$$x(y_2) - \|y\| \leq x(y_1).$$

Therefore,

$$\rho(y_2) - \|y\| \leq \rho(y_1).$$

The elements y_1 and y_2 can be interchanged to obtain

$$\rho(y_1) - \|y\| \leq \rho(y_2)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and hence

$$|\rho(y_1) - \rho(y_2)| \leq \|y_1 - y_2\| .$$

Lemma (5.6): The functional ρ is subadditive and positive homogeneous (hence convex).

Proof: Assume y_1 and y_2 are in Y. To show that ρ is subadditive we must show that

$$\rho(y_1 + y_2) \leq \rho(y_1) + \rho(y_2).$$

By definition

$$\rho(y_1+y_2) = \sup \{x(y_1+y_2) : x \in C \text{ and } \|x\| \leq 1\}$$

$$\leq \sup \{x(y_1) : x \in C \text{ and } \|x\| \leq 1\}$$

$$+ \sup \{x(y_2) : x \in c \text{ and } \|x\| \leq 1\}$$

$$= \rho(y_1) + \rho(y_2).$$

Now assume $\alpha > 0$ and $y \in Y$. To show that ρ is positive homogeneous we must show that

$$\rho(\alpha y) = \alpha\rho(y).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

By definition

$$\rho(\alpha y) = \sup \{x(\alpha y) : x \in C \text{ and } \|x\| \leq 1\}$$

$$= \alpha \cdot \sup \{x(y) : x \in C \text{ and } \|x\| \leq 1\}$$

$$= \alpha\rho(y).$$

This completes the proof of the lemma.

As an example we compute ρ for the case $C = \{x \in L_p[a,b]: x \geq 0\}$ where $1 < p < \infty$. For an arbitrary element g in $L_q[a,b]$, the predual of $L_p[a,b]$ where $p + q = pq$, we have for any $f \in C$ with $\|f\|_p \leq 1$ by the Minkowski inequality

$$\int_a^b f(t)g(t)dt \leq \int_a^b f(t)g_+(t)dt$$

$$\leq \|f\|_p \cdot \|g_+\|_q$$

$$\leq \|g_+\|_q .$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Assuming $g_+ \neq 0$, let

$$f = (g)_+^{q-1} / \left\| (g)_+^{q-1} \right\|_p .$$

Then we have $f \in C$, $\| f \|_p = 1$, and

$$\int_a^b f(t)g(t)dt = \| g_+ \| .$$

Hence

$$\rho(g) = \sup \{ \int_a^b f(t)g(t)dt : f \in C \text{ and } \| f \|_p \leq 1 \}$$

$$= \| g_+ \|_q .$$

If $g_+ = 0$, then $\rho(g) = 0$.

Lemma(5.7): For all $\underline{\alpha} \in R^n$ we have

$$\sum_{i=1}^n \alpha_i d_i \leq \gamma^* \rho(\sum_{i=1}^n \alpha_i y_i). \tag{5.6}$$

Proof: Since $\gamma^* = \inf\{ \gamma : \underline{d} \in G(\gamma) \}$, we have $\underline{d} \in G(\gamma^* + \varepsilon)$ for any $\varepsilon > 0$. Hence for every positive integer n there exists $x_m \in C$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

such that $Tx_m = \underline{d}$ and $\|x_m\| \leq \gamma^* + 1/m$. Therefore, for any $\underline{\alpha} \in R^n$

$$\sum_{i=1}^{n} \alpha_i d_i = \sum_{i=1}^{n} \alpha_i x_m(y_i)$$

$$= x_m\left(\sum_{i=1}^{n} \alpha_i y_i\right)$$

$$\leq \|x_m\| \rho\left(\sum_{i=1}^{n} \alpha_i y_i\right)$$

$$\leq (\gamma^* + 1/m)\rho\left(\sum_{i=1}^{n} \alpha_i y_i\right).$$

Now let $m \to \infty$ to obtain (5.6). This completes the proof of the lemma.

Since we know that $G(\gamma^*)$ is closed from proposition (5.4), we could have used x_* in place of x_m in the proof of lemma (5.7). The next lemma states that there exists a nonzero vector $\underline{\beta} \in R^n$ such that equality holds in (5.6).

Proposition (5.8): There exists a vector $\underline{\beta} \in R^n$ such that $\|\underline{\beta}\| = 1$ and

$$\underline{\beta} \cdot \underline{d} = \gamma^* \left(\sum_{i=1}^{n} \beta_i y_i\right). \tag{5.7}$$

Proof: The vector $\underline{d}$ is an element of $G(\gamma^*)$, but not an element of $G(\gamma^* - \varepsilon)$ for any $\varepsilon > 0$. Hence the closed convex set $G(\gamma^* - \varepsilon)$ and the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

vector $\underline{d}$ can be strictly separated by a hyperplane. This implies the existence of a nonzero vector $\underline{\beta}(\varepsilon)$ such that

$$\underline{\beta}(\varepsilon) \cdot \underline{y} < \beta(\varepsilon) \cdot \underline{d}$$

for all $\underline{y} \in G(\gamma^* - \varepsilon)$ and without loss of generality we may assume that $\|\underline{\beta}(\varepsilon)\| = 1$. Equivalently, we have

$$\underline{\beta}(\varepsilon) \cdot Tx < \underline{\beta}(\varepsilon) \cdot \underline{d}$$

and by the linearity of T

$$x\left(\sum_{i=1}^{n} \beta_i(\varepsilon) y_i\right) < \underline{\beta}(\varepsilon) \cdot \underline{d}$$

for all $x \in C$ such that $\|x\| \leq \gamma^* - \varepsilon$. Hence we obtain

$$(\gamma^* - \varepsilon)\rho\left(\sum_{i=1}^{n} \beta_i(\varepsilon) y_i\right) < \underline{\beta}(\varepsilon) \cdot \underline{d}$$

We can take the limit as $\varepsilon \to 0$ to obtain a vector $\underline{\beta}$ such that $\|\underline{\beta}\| = 1$ and

$$\gamma^* \rho\left(\sum_{i=1}^{n} \beta_i y_i\right) \leq \underline{\beta} \cdot \underline{d}.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We have the reverse inequality from lemma (5.7) and therefore

$$\underline{\beta} \cdot \underline{d} = \gamma^{*}\rho(\sum_{i=1}^{n} \beta_i y_i).$$

This completes the proof of the lemma.

Let λ be a linear functional defined on the subspace

$$S := \text{span}(y_1, y_2, \ldots, y_n)$$

by

$$\lambda(\sum_{i=1}^{n} \alpha_i y_i) = \sum_{i=1}^{n} \alpha_i d_i$$

so that (5.6) can now be written

$$\lambda(y) \leq \gamma^{*}\rho(y) \qquad \text{for all } y \in S.$$

The Hahn-Banach theorem states that there exists an element w in X such that

$$w(y) = \lambda(y) \qquad \text{for all } y \in S \tag{5.8}$$

and

$$w(y) \leq \gamma^{*}\rho(y) \qquad \text{for all } y \in Y. \tag{5.9}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem (5.9): The Hahn-Banach extension w is an interpolant of minimal norm.

Proof: From (5.8) we see that $Tw = \underline{d}$ so that w interpolates the data. To complete the proof we show that $w \in C$ and $||w|| = \gamma^*$.

We show that w is in C by contradiction. Assume w is not an element of C. Since C is weak* closed, there exists an element y_o in Y which strictly separates w from C in the sense that

$$w(y_o) > x(y_o) \qquad \text{for all } x \in C. \tag{5.10}$$

Since C is a cone we have $\lambda x \in C$ whenever $\lambda > 0$ and $x \in C$. Hence (5.10) implies

$$0 \geq x(y_o) \qquad \text{for all } x \in C \tag{5.11}$$

(or $\rho(y_o) = 0$) and

$$w(y_o) > 0. \tag{5.12}$$

However, from (5.9) and (5.12) we have

$$0 < w(y_o) \leq \gamma^* \rho(y_o) = 0$$

which is a contradiction. Hence w must be an element of C.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Lastly, we show that $\|w\| = \gamma^*$. We already know that

$$\gamma^* \leq \|w\| \tag{5.13}$$

since $w \in B$ ($w \in C$ and $Tw = \underline{d}$). Because ρ is bounded above by the norm on Y, (5.9) yields

$$w(y) \leq \gamma^* \|y\| \qquad \text{for all } y \in Y$$

and hence

$$\|w\| \leq \gamma^*. \tag{5.14}$$

Taken together, (5.13) and (5.14) imply that $\|w\| = \gamma^*$. This completes the proof of the theorem.

Recall that for a given element y_o in Y there exists an element x_o (a function of y_o) in C such that $\rho(y_o) = x(y_o)$. Furthermore, either $\|x_o\| = 1$ or x_o is the zero element. The following lemma will lead us to the conclusion that, if ρ is differentiable at y_o, then $\rho'(y_o) = x_o$.

Lemma (5.10): Let f be a functional defined on a normed linear space Z. If f is differentiable at $x_o \in Z$ and if there exists a linear functional λ such that

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$f(z_o) + \lambda(z-z_o) \leqq f(z) \tag{5.15}$$

for all z in some neighborhood of z_o, then $\lambda = (\nabla f)(z_o)$.

Proof: Let $z = z_o + tu$ where $t > 0$ and $u \in Z$. Inequality (5.15) yields

$$\lambda(u) \leqq \frac{f(z_o + tu) - f(z_o)}{t} . \tag{5.16}$$

Since (5.16) holds for all $t > 0$ (and sufficiently small) and for all $u \in Z$, we have $\lambda \leqq (\nabla f)(z_o)$. Substituting $-u$ for u in (5.16) yields

$$\lambda(u) \geqq \frac{f(z_o - tu) - f(z_o)}{t} \tag{5.17}$$

for all $t > 0$ (and sufficiently small) and for all $u \in Z$. Taken together, (5.16) and (5.17) imply $\lambda = (\nabla f)(z_o)$.

Corollary (5.11): If ρ is differentiable at $y_o \in Y$, then $\rho'(y_o) = x_o$.

Proof: Since $\rho(y_o) = x_o(y_o)$ and $x_o(y) \leqq \rho(y)$ for all $y \in Y$, we have

$$\rho(y_o) + x_o(y - y_o) \leqq \rho(y)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

for all $y \in Y$. By the previous lemma we have $\rho'(y_o) = x_o$. This completes the proof of the corollary.

Inequality (5.6) motivates the problem

$$\inf_{\underline{\alpha}} \{ \rho(\sum_{i=1}^{n} \alpha_i y_i) : \underline{\alpha} \cdot \underline{d} = 1 \} . \tag{5.18}$$

Notice that if $\underline{\alpha}$ is any vector satisfying $\underline{\alpha} \cdot \underline{d} = 1$ and if x is any element of B, then

$$1 = \sum_{i=1}^{n} \alpha_i d_i = x(\sum_{i=1}^{n} \alpha_i y_i)$$

$$\leq \|x\| \rho(\sum_{i=1}^{n} \alpha_i y_i)$$

and hence

$$\rho(\sum_{i=1}^{n} \alpha_i y_i) \geq \frac{1}{\|x\|}$$

This implies that the infimum is positive (and, in fact, is bounded below by $(\gamma^*)^{-1}$. If the infimum is attained at some $\underline{\alpha}^* \in R^n$ and if ρ is differentiable at $\sum_{i=1}^{n} \alpha_i^* y_i$, then we are led to a solution to (5.1) as

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the next theorem reveals.

Theorem (5.12): If there exists $\underline{\alpha}^* \in R^n$ such that $\underline{\alpha}^* \cdot \underline{d} = 1$ and

$$\rho\left(\sum_{i=1}^{n} \alpha_i^* y_i\right) = \inf_{\underline{\alpha}} \{ \rho\left(\sum_{i=1}^{n} \alpha_i y_i\right) : \underline{\alpha} \cdot \underline{d} = 1\}$$

and if ρ is differentiable at $\sum_{i=1}^{n} \alpha_i^* y_i$, then

$$\gamma^* \rho'\left(\sum_{i=1}^{n} \alpha_i y_i\right)$$

is an interpolant of minimal norm.

Proof: Problem (5.18) has Lagrangian

$$L(\underline{\alpha},\lambda) = \rho\left(\sum_{i=1}^{n} \alpha_i y_i\right) - \lambda\left(\sum_{i=1}^{n} \alpha_i d_i - 1\right). \qquad (5.19)$$

If there exists a solution $\underline{\alpha}^*$ to (5.18), then there exists λ^* so that $(\underline{\alpha}^*, \lambda^*)$ is a stationary point of (5.19). Hence

$$x(y_i) - \lambda^* d_i = 0 \qquad i=1,2,\ldots,n \qquad (5.20)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where $x = \rho'(\sum_{i=1}^{n} \alpha_i^* y_i)$, $x \in C$, $\|x\| = 1$, and $\alpha^* \cdot \underline{d} = 1$.

We first show that $\lambda^* > 0$. Multiply (5.20) by α_i^* and sum over i to obtain

$$x(\sum_{i=1}^{n} \alpha_i^* y_i) = \lambda^* \sum_{i-1}^{n} \alpha_i^* d_i = \lambda^*.$$

Since $x = \rho'(\sum_{i=1}^{n} \alpha_i^* y_i)$, we have

$$x(\sum_{i=1}^{n} \alpha_i^* y_i) = \rho(\sum_{i=1}^{n} \alpha_i^* y_i)$$

so that

$$\lambda^* = \rho(\sum_{i=1}^{n} \alpha_i^* y_i) \geq 0$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Actually, we know that since the infimum is positive, we have $\lambda^* > 0$. We can also show this by contradiction. If $\lambda^* = 0$, then

$$x(\sum_{i=1}^{n} \alpha_i^* y_i) \leq 0 \quad \text{for all } x \in C. \tag{5.21}$$

Let s be any interpolant in C. (We know that there exists an interpolant in C since B is nonempty.) Then

$$s(\sum_{i=1}^{n} a_i^* y_i) = \sum_{i=1}^{n} \alpha_i^* d_i = 1$$

which contradicts (5.21). Therefore, $\lambda^* > 0$.

Now we show that $\lambda^* \gamma^* = 1$. From (5.20) we see that x/λ^* is an interpolant in C. Hence

$$\gamma^* \leq \|x\| / \lambda^* = 1/\lambda^*$$

or

$$\gamma^* \lambda^* \leq 1 \tag{5.22}$$

Let w be an interpolant of minimal norm satisfying (5.9). Then

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$w\left(\sum_{i=1}^{n} \alpha_i^* y_i\right) \leq \gamma^* \rho\left(\sum_{i=1}^{n} \alpha_i^* y_i\right).$$

Equivalently, we have

$$w\left(\sum_{i=1}^{n} \alpha_i^* y_i\right) \leq \gamma^* x\left(\sum_{i=1}^{n} \alpha_i^* y_i\right)$$

which leads to

$$1 \leq \gamma^* \lambda^*. \tag{5.23}$$

Taken together, (5.22) and (5.23) imply

$$1 = \gamma^* \lambda^*.$$

This concludes the proof of the theorem.

We consider now the problem of determining when the infimum is attained in (5.18). From proposition (5.8) we know that there exist a nonzero vector $\underline{\beta}$ such that

$$0 \leq \underline{\beta} \cdot \underline{d} = \gamma^* \rho\left(\sum_{i=1}^{n} \beta_i y_i\right).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

If $\underline{\beta} \cdot \underline{d} > 0$, then the infimum is attained in (5.18) at $\underline{\alpha}^* = \underline{\beta}/(\underline{\beta} \cdot \underline{d})$.

Proposition (5.13): If $\underline{d}$ is in the relative interior of

$$S: = \{\underline{r} : \underline{r} \in G(\gamma) \text{ for some } \gamma \},$$

then there exists a vector $\underline{\beta}$ such that

$$1 = \underline{\beta} \cdot \underline{d} = \gamma^* \rho(\sum_{i=1}^{n} \beta_i y_i).$$

Proof: We prove by contradiction. Assume that every vector $\underline{\beta}$ which satisfies

$$\underline{\beta} \cdot \underline{d} = \gamma^* \rho(\sum_{i=1}^{n} \beta_i y_i)$$

also satisfies $\underline{\beta} \cdot \underline{d} = 0$. Without loss of generality it can be assumed that there exists a nonzero vector $\underline{\beta}$ such that

$$0 = \underline{\beta} \cdot \underline{d} = \gamma^* \rho(\sum_{i=1}^{n} \beta_i y_i)$$

and

$$\underline{\beta} \cdot \underline{y} \geq 0 \quad \text{for all } \underline{y} \in G(\gamma^*).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In any relative neighborhood of $\underline{d}$ there is a vector $\underline{z}$ such that $\underline{\beta} \cdot \underline{z} < 0$. If $\underline{z}$ were an element of S, then there would be an element $\underline{r}$ in $G(\gamma^*)$ such that $\underline{z} = \alpha\underline{r}$ for some $\alpha > 0$. However, we would then have

$$\underline{\beta} \cdot \underline{z} = \alpha\, \underline{\beta} \cdot \underline{r} \geq 0$$

which is a contradiction. Therefore $\underline{z}$ is not an element of S and $\underline{d}$ is not in the relative interior of S. This completes the proof of the proposition.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

References

[de B(1)] C. de Boor (1978): "A Practical Guide to Splines," Springer-Verlag, New York.

[de B(2)] C. de Boor (1976): On "best" interpolation, J. Approx. Theory, 16:28-42.

[IMS] L.D. Irvine, P.W. Smith, S.P. Marin (1985): Constrained interpolation and smoothing, General Motors Research Report No. MA-312, Warren, Michigan.

[L] D.G. Luenberger (1973): "Optimization by Vector Space Methods," John Wiley and Sons, Inc., New York.

[MSSW] C.A. Micchelli, P.W. Smith, J. Swetits, J.D. Ward (1985): Constrained L_p approximation, J. Constructive Approximation, 1:93-102.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix A

A Program for Constructing the Natural Cubic Spline Interpolant to Given Data.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00001       PROGRAM UNCON(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
00002C
00003C        WE FORM THE NATURAL CUBIC SPLINE INTERPOLANT.
00004C
00005       INTEGER N,M,I
00006       REAL T(50),F(50),D(50),X(50),A(50),PP(4,50)
00007       REAL AA(50),BB(50),CC(50)
00008C
00009C        THE ARRAYS (T) AND (F) - EACH OF SIZE M, THE NUMBER
00010C        OF DATA. POINTS - CONTAIN THE COMPONENTS OF THE DATA.
00011C        THE DATA FILE IS OF THE FOLLOWING FORM
00012C
00013C          M
00014C          T(1),F(I)
00015C          T(2),F(2)
00016C              .
00017C              .
00018C              .
00019C          T(M),F(M)
00020C
00021C        WHERE WE ASSUME (T) HAS STRICTLY INCREASING COMPONENTS.
00022C
00023       READ(3,*) M
00024       READ(3,*) (T(I),F(I), I=1,M)
00025       N= M-2
00026C
00027C        THE ARRAY (D) CONSISTS OF THE SCALED
00028C        SECOND DIVIDED DIFFERENCES.
00029C
00030       DO 100 I=1,N
00031       D(I)= ( F(I+2)-F(I+1) )/( T(I+2)-T(I+1) )
00032      C        - ( F(I+1)-F(I) )/( T(I+1)-T(I) )
00033 100   CONTINUE
00034C
00035C
00036C        THE SECOND DERIVATIVE OF THE NATURAL CUBIC SPLINE
00037C        INTERPOLANT IS A LINEAR COMBINATION OF LINEAR B-SPLINES.
00038C        WE CALCULATE THE COEFFICIENTS.
00039C
00040C
00041       AA(1)= 0.0
00042       BB(1)= (T(3)-T(1))/3.0
00043       CC(1)= (T(3)-T(2))/6.0
00044       DO 200 I=2,N-1
00045       AA(I)= (T(I+1)-T(I))/6.0
00046       BB(I)= (T(I+2)-T(I))/3.0
00047       CC(I)= (T(I+2)-T(I+1))/6.0
00048 200   CONTINUE
00049       AA(N)= (T(N+1)-T(N))/6.0
00050       BB(N)= (T(N+2)-T(N))/3.0
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00051       CC(N)= 0.0
00052       CALL TRID(AA,BB,CC,D,N)
00053C
00054C
00055C
00056       A(1)= 0.0
00057       A(M)= 0.0
00058       DO 300 I=2,N+1
00059       A(I)= D(I-1)
00060 300   CONTINUE
00061C
00062C
00063C        NOW WE COMPUTE THE NUMBERS PP(J,I) - THE VALUE
00064C        OF THE (J-1)ST DERIVATIVE OF THE NATURAL CUBIC
00065C        SPLINE INTERPOLANT EVALUATED AT T(I).
00066C
00067C
00068       DO 400 K=1,N+1
00069       DF= F(K+1)-F(K)
00070       DT= T(K+1)-T(K)
00071       DA= A(K+1)-A(K)
00072       PP(4,K)= DA/DT
00073       PP(3,K)= A(K)
00074       PP(2,K)= DF/DT - (A(K)/2. + DA/6.)*DT
00075       PP(1,K)= F(K)
00076 400   CONTINUE
00077       PP(4,M)= 0.0
00078       PP(3,M)= 0.0
00079       PP(2,M)= 0.0
00080       PP(1,M)= F(M)
00081C
00082C
00083C
00084       DO 500 K=1,M
00085       WRITE(6,450) K,T(K),(PP(I,K), I=1,4)
00086 450   FORMAT(5X,I5,5F14.6)
00087 500   CONTINUE
00088C
00089C        WE CREATE A DATA FILE FOR PLOTTING THE (JDER)-TH
00090C        DERIVATIVE OF THE NATURAL CUBIC SPLINE INTERPOLANT
00091C        BY EVALUATING IT AT (MM) EQUALLY SPACED POINTS,
00092C        INCLUDING THE ENDPOINTS.  WE ASSUME THAT (JDER)
00093C        HAS VALUE 0, 1, 2, OR 3.
00094C
00095       JDER= 0
00096       MM= 201
00097       CALL DATAFL(T,PP,M,MM,JDER)
00098C
00099       STOP
00100       END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00001       SUBROUTINE DATAFL(TX,PP,LI,MM,JDER)
00002C
00003C        WE CREATE A DATA FILE FOR PLOTTING THE (JDER)-TH
00004C        DERIVATIVE OF THE PIECEWISE CUBIC POLYNOMIAL.  WE
00005C        ASSUME (JDER) HAS VALUE 0, 1, 2, OR 3.
00006C
00007       INTEGER LI,MM,JDER
00008       REAL TX(100),PP(4,100)
00009       LEFT= 1
00010       MMONE= MM - 1
00011       WRITE(4,*) MM
00012       XE= ( TX(LI)-TX(1) )/FLOAT(MMONE)
00013       DO 500 IP=1,MM
00014       XT= TX(1) + XE*FLOAT(IP-1)
00015C
00016C        WE FIND THE INTERVAL IN WHICH THE POINT (XT) LIES.
00017C
00018           IF ( LEFT .NE. LI ) THEN
00019             DO 200 IS=LEFT,LI-1
00020             IF ( XT .LT. TX(IS+1) ) GO TO 300
00021 200         CONTINUE
00022 300         CONTINUE
00023           END IF
00024           LEFT= IS
00025C
00026C        WE NOW COMPUTE THE VALUE OF THE POLYNOMIAL AT
00027C        THE POINT (XT) BY USING NESTED MULTIPLICATION.
00028C
00029       H= XT - TX(LEFT)
00030       FAC= 4.0 - FLOAT(JDER)
00031       YT= 0.0
00032       DO 400 M=4,JDER+1,-1
00033           YT= (YT/FAC)*H + PP(M,LEFT)
00034           FAC= FAC - 1.0
00035 400   CONTINUE
00036       WRITE(4,450) XT,YT
00037 450   FORMAT(F8.4,E18.9)
00038 500   CONTINUE
00039       RETURN
00040       END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00001       SUBROUTINE TRID(SUB,DIAG,SUP,B,N)
00002       INTEGER N,I
00003       REAL B(N),DIAG(N),SUB(N),SUP(N)
00004            IF (N.LE.1) THEN
00005            B(1)= B(1)/DIAG(1)
00006            RETURN
00007            END IF
00008       DO 111 I=2,N
00009            SUB(I)= SUB(I)/DIAG(I-1)
00010            DIAG(I)= DIAG(I) - SUB(I)*SUP(I-1)
00011            B(I)= B(I) - SUB(I)*B(I-1)
00012 111   CONTINUE
00013            B(N)= B(N)/DIAG(N)
00014       DO 222 I=N-1,1,-1
00015            B(I)= (B(I)-SUP(I)*B(I+1))/DIAG(I)
00016 222   CONTINUE
00017       RETURN
00018       END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix B

A Program for Constructing the Shape-Preserving Cubic Spline Interpolant to Given Data

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00001       PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
00002C
00003C        WE COMPUTE A SHAPE-PRESERVING INTERPOLANT
00004C        TO GIVEN DATA.
00005C
00006C
00007C        NOTE ON THE SIZE OF THE ARRAYS:
00008C        THE ARRAYS (T), (F), AND (A) MUST BE OF LENGTH
00009C        AT LEAST M, THE NUMBER OF DATA POINTS.  THE
00010C        ARRAY (TX) AND THE SECOND COMPONENT OF THE
00011C        ARRAY (PP) SHOULD BE OF LENGTH 2M.  THE ARRAYS
00012C        (X), (Y), AND (D) MUST BE OF LENGTH AT LEAST M-2.
00013C        THE ARRAY (ID) MUST BE OF LENGTH AT LEAST M-1.
00014C
00015C
00016       REAL T(50),F(50),X(50),Y(50),A(50)
00017       REAL TX(100),PP(4,100),TL,TR,EPS
00018       INTEGER M,N,ITMAX,I,J,IFLAG,MM
00019       COMMON D(50),ID(50)
00020C
00021C
00022C
00023C        THE ARRAYS (T) AND (F) - EACH OF SIZE M, THE NUMBER
00024C        OF DATA POINTS - CONTAIN THE COMPONENTS OF THE DATA.
00025C        THE DATA FILE IS OF THE FOLLOWING FORM
00026C
00027C            M
00028C            T(1),F(1)
00029C            T(2),F(2)
00030C                .
00031C                .
00032C                .
00033C            T(M),F(M)
00034C
00035C        WHERE WE ASSUME (T) HAS STRICTLY INCREASING COMPONENTS.
00036C
00037       READ(3,*) M
00038       READ(3,*) (T(I),F(I), I=1,M)
00039       N= M-2
00040C
00041C        (EPS) IS A SMALL POSITIVE NUMBER USED TO TEST FOR
00042C        CONVERGENCE IN NEWTON'S METHOD - SUBROUTINE (ZERO).
00043C        (ITMAX) IS THE MAXIMUM NUMBER OF ITERATIONS
00044C        WHICH WE PERMIT FOR NEWTONS METHOD TO CONVERGE.
00045C
00046       EPS= 1.0E-8
00047       ITMAX= 25
00048C
00049C        THE ARRAY (X) IS THE KNOT SEQUENCE (T) WITH THE
00050C        ENDPOINTS TL AND TR DELETED.
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00051C
00052       TL= T(1)
00053       TR= T(M)
00054       DO 120 I=1,N
00055       X(I)= T(I+1)
00056 120   CONTINUE
00057C
00058C        THE ARRAY (D) CONSISTS OF THE SCALED
00059C        SECOND DIVIDED DIFFERENCES.
00060C
00061C        IT IS IMPORTANT THAT WE IDENTIFY DIVIDED DIFFERENCES
00062C        WHICH ARE ZERO.  THIS MEANS THAT WE MUST COMPARE TWO
00063C        FLOATING-POINT NUMBERS.  TO DO THIS WE ASSUME D(K) IS
00064C        ZERO IF D(K) IS SMALL.
00065C
00066            XEPS= 1.0
00067            DO 130 J=1,20
00068            XEPS= XEPS/10.
00069            Z= 1.0 + XEPS
00070            IF ( Z .EQ. 1.0 ) GO TO 135
00071            YEPS= XEPS
00072 130        CONTINUE
00073 135        CONTINUE
00074            YEPS= YEPS*1000.
00075C
00076       DO 140 K=1,N
00077       D(K)= ( F(K+2)-F(K+1) )/( T(K+2)-T(K+1) )
00078     C         - ( F(K+1)-F(K) )/( T(K+1)-T(K) )
00079       IF ( ABS(D(K)) .LE. YEPS ) D(K)= 0.0
00080 140   CONTINUE
00081C
00082C
00083C        THE INITIAL GUESS (Y) FOR NEWTON'S METHOD
00084C        WILL YIELD THE SECOND DERIVATIVE OF THE
00085C        NATURAL SPLINE SOLUTION, EXCEPT POSSIBLY
00086C        WHEN D(K)= 0.0 FOR SOME K.
00087C
00088       DO 145 K=1,N
00089C
00090C
00091       IF ( D(K) .GT. 0.0 ) THEN
00092            Y(K)=  1.0
00093       ELSE
00094            Y(K)= -1.0
00095       END IF
00096C
00097C
00098 145   CONTINUE
00099C
00100C
00101       WRITE(6,150)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00102 150   FORMAT(/,' DATA VALUES ',/)
00103       WRITE(6,160) (D(I), I=1,N)
00104 160   FORMAT(5X,4E12.6)
00105       WRITE(6,170)
00106 170   FORMAT(//)
00107C
00108C
00109C        ID(K)= 1 INDICATES THAT THE INTERPOLATING FUNCTION
00110C        IS CONSTRAINED TO BE CONVEX ON [T(K),T(K+1)]
00111C        AND, HENCE, ITS SECOND DERIVATIVE IS CONSTRAINED
00112C        TO BE NONNEGATIVE ON THIS INTERVAL.
00113C
00114C        ID(K)= -1 INDICATES THAT THE INTERPOLATING FUNCTION
00115C        IS CONSTRAINED TO BE CONCAVE ON [T(K),T(K+1)]
00116C        AND, HENCE, ITS SECOND DERIVATIVE IS CONSTRAINED
00117C        TO BE NONPOSITIVE ON THIS INTERVAL.
00118C
00119C        ID(K)= 0 INDICATES THAT THE INTERPOLATING FUNCTION
00120C        IS UNCONSTRAINED ON [T(K),T(K+1)].
00121C
00122C
00123       DO 180 I=1,N-1
00124       ID(I+1)= 0
00125       IF (D(I).GE.0.0 .AND. D(I+1).GE.0.0) ID(I+1)= 1
00126       IF (D(I).LE.0.0 .AND. D(I+1).LE.0.0) ID(I+1)= -1
00127 180   CONTINUE
00128       IF ( D(1) .GE. 0.0 ) THEN
00129            ID(1)=  1
00130       ELSE
00131            ID(1)= -1
00132       END IF
00133C
00134       IF ( D(N) .GE. 0.0 ) THEN
00135            ID(N+1)=  1
00136       ELSE
00137            ID(N+1)= -1
00138       END IF
00139C
00140C        IF A NONZERO DATA VALUE D(I) LIES BETWEEN TWO
00141C        ZERO DATA VALUES D(I-1) AND D(I+1), THEN D(I)
00142C        IS TAKEN TO BE ZERO FOR COMPUTATIONAL PURPOSES.
00143C
00144       DO 185 I=2,N-1
00145       IF ( D(I-1).EQ.0.0 .AND. D(I+1).EQ.0.0 ) D(I)= 0.0
00146 185   CONTINUE
00147C
00148C        SUBROUTINE (ZERO) CALCULATES THE PIECEWISE
00149C        LINEAR SECOND DERIVATIVE OF THE SHAPE-
00150C        PRESERVING INTERPOLANT.
00151C
00152       CALL ZERO(Y,X,N,ITMAX,EPS,IFLAG,TL,TR)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00153C
00154       A(1)=  0.0
00155       A(M)= 0.0
00156       DO 190 I=2,N+1
00157       A(I)= Y(I-1)
00158 190   CONTINUE
00159C
00160       WRITE(6,200)
00161 200   FORMAT(/,' PIECEWISE LINEAR 2ND DERIVATIVE ',/)
00162       WRITE(6,210) (I,T(I),A(I), I=1,M)
00163 210   FORMAT(5X,I5,' ( ',F14.6,' , ',F14.6,' )')
00164       WRITE(6,220)
00165 220   FORMAT(//)
00166C
00167C        SUBROUTINE (POLY) INTEGRATES THE RESULT
00168C        FROM SUBROUTINE (ZERO).
00169C
00170       CALL POLY(A,T,PP,M,F,LI,TX)
00171C
00172       WRITE(6,230)
00173 230   FORMAT(/,' KNOTS AND COEFFICIENTS OF PIECEWISE CUBIC',/)
00174       DO 250 I=1,LI
00175       WRITE(6,240) I,TX(I),(PP(J,I), J=1,4)
00176 240   FORMAT(5X,I5,5F14.6)
00177 250   CONTINUE
00178       WRITE(6,260) IFLAG
00179 260   FORMAT(/,' ERROR CODE = ',I5,/)
00180       WRITE(6,270) ITMAX
00181 270   FORMAT(/,' NUMBER OF ITERATIONS =',I5,)
00182C
00183C        SUBROUTINE (DATAFL) IS USED TO CREATE A
00184C        DATA FILE FOR PLOTTING.  WE EVALUATE THE
00185C        (JDER)-TH DERIVATIVE OF THE PIECEWISE CUBIC
00186C        POLYNOMIAL AT MM EQUALLY SPACED POINTS,
00187C        INCLUDING THE ENDPOINTS TL AND TR. WE
00188C        ASSUME (JDER) HAS VALUE 0, 1, 2, OR 3.
00189C
00190       MM= 201
00191       JDER= 0
00192       CALL DATAFL(TX,PP,LI,MM,JDER)
00193C
00194       STOP
00195       END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      SUBROUTINE ZERO(A,X,N,ITMAX,EPS,IFLAG,TL,TR)
C
C
      INTEGER N,ITMAX,K,J,LJ,L,IFLAG
      REAL A(N),X(N),FX(50),AL,XL,AR,XR,DT,DA,T,W
      REAL SUB(50),DIAG(50),SUP(50),H(50),SUM1,SUM2
      REAL RATIO,GLEFT,GRIGH,EPS,FNORM1,TL,TR
      COMMON D(50),ID(50)
C
C           INPUT PARAMETERS:
C
C  A...INITIAL ESTIMATE FOR NEWTON'S METHOD.
C
C  X...KNOT SEQUENCE WITH THE ENDPOINTS DELETED.
C
C  N...THE SIZE OF THE ARRAY (A): THE NUMBER OF UNKNOWNS.
C
C  ITMAX...MAXIMUM NUMBER OF ITERATIONS FOR NEWTON'S METHOD.
C
C  EPS...PARAMETER USED TO TEST FOR CONVERGENCE.
C
C  TL,TR...LEFT- AND RIGHT-ENDPOINTS OF THE
C          INTERVAL RESPECTIVELY.
C
C           OUTPUT PARAMETERS:
C
C  A...THE CALCULATED ZERO IF CONVERGENCE OCCURRED.
C
C  ITMAX...NUMBER OF ITERATIONS REQUIRED FOR NEWTON'S
C          METHOD TO CONVERGE.
C
C  IFLAG...IFLAG= 1: CONVERGENCE INDICATED BY COMPARING
C          THE L1 NORMS OF THE ITERATES
C          IFLAG= 2: NUMBER OF ITERATIONS EXCEEDED ITMAX.
C
      PRINT 100
  100 FORMAT(' ITERATION NUMBER AND RESIDUAL:',/
     C          ' QUADRATIC CONVERGENCE IS EXPECTED.',/)
      DO 350 LJ=1,ITMAX
C
C           THE ARRAYS (SUB), (DIAG), AND (SUP) CONTAIN
C  THE ELEMENTS OF THE TRIDIAGONAL POSITIVE-DEFINITE
C  JACOBIAN MATRIX (J), EVALUATED AT THE VECTOR (A).
C  IT SHOULD BE NOTED THAT THE MATRIX EQUATION SOLVER,
C  THE SUBROUTINE (TRID), DOES NOT TAKE ADVANTAGE OF
C  THE SYMMETRY OF (J).  HENCE (SUB) AND (SUP) ARE
C  BOTH NECESSARY. ALTHOUGH SUB(K)=SUP(K-1), EQUATIONS
C  FOR BOTH ARRAYS ARE WRITTEN OUT IN FULL.
C
C           IF D(K)=0.0 FOR SOME K, THEN THE NUMBER
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00051C        OF UNKNOWNS (AND EQUATIONS) REDUCE. IN ORDER
00052C        TO PERMIT THE COMPUTATION OF ONE JACOBIAN
00053C        MATRIX THE PROGRAM SETS SUB(K)=SUP(K-1)=0.0
00054C        AND DIAG(K)=1.0.
00055C
00056       DO 125 K=1,N
00057C
00058       IF (K.EQ.1) THEN
00059                   AL= 0.0
00060                   XL= TL
00061       ELSE
00062            AL= A(K-1)
00063            XL= X(K-1)
00064       END IF
00065C
00066C
00067       IF (K.EQ.N) THEN
00068                   AR= 0.0
00069                   XR= TR
00070       ELSE
00071            AR= A(K+1)
00072            XR= X(K+1)
00073       END IF
00074C
00075C
00076       IF ( AL.GE.0.0 .AND. A(K).GE.0.0 ) J1= 1
00077       IF ( AL.LT.0.0 .AND. A(K).GE.0.0 ) J1= 2
00078       IF ( AL.GE.0.0 .AND. A(K).LT.0.0 ) J1= 3
00079       IF ( AL.LE.0.0 .AND. A(K).LE.0.0 ) J1= 4
00080C
00081       IF ( A(K).GE.0.0 .AND. AR.GE.0.0 ) J2= 1
00082       IF ( A(K).LT.0.0 .AND. AR.GE.0.0 ) J2= 2
00083       IF ( A(K).GE.0.0 .AND. AR.LT.0.0 ) J2= 3
00084       IF ( A(K).LE.0.0 .AND. AR.LE.0.0 ) J2= 4
00085C
00086       DT= X(K)-XL
00087       DA= A(K)-AL
00088C
00089       IF ( ID(K) .EQ. 1) THEN
00090C
00091           IF (K.NE.1) THEN
00092C
00093               IF (J1.EQ.1) THEN
00094C
00095       SUB(K)= DT/6.0
00096       GLEFT=  DT/3.0
00097C
00098               ELSE IF (J1.EQ.2) THEN
00099C
00100       T= XL-(DT/DA)*AL
00101       W= 0.5*( X(K)+T )
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00102       SUB(K)= (X(K)-T)/6.0 * ( ((T-XL)/DT)*((X(K)-T)/DT)
00103      C              + 4.0*((W-XL)/DT)*((X(K)-W)/DT) )
00104       GLEFT=  (X(K)-T)/6.0 * ( ((T-XL)/DT)**2
00105      C              + 4.0*(((W-XL)/DT)**2) + 1.0 )
00106C
00107              ELSE IF (J1.EQ.3) THEN
00108C
00109       T= XL-(DT/DA)*AL
00110       W= 0.5*( T+XL )
00111       SUB(K)= (T-XL)/6.0 * ( 4.0*((W-XL)/DT)*((X(K)-W)/DT)
00112      C              + ((T-XL)/DT)*((X(K)-T)/DT) )
00113       GLEFT=  (T-XL)/6.0 * ( 4.0*(((W-XL)/DT)**2)
00114      C              + ((T-XL)/DT)**2 )
00115C
00116              ELSE IF (J1.EQ.4) THEN
00117C
00118       SUB(K)= 0.0
00119       GLEFT=  0.0
00120C
00121              END IF
00122C
00123          ELSE IF (K.EQ.1) THEN
00124C
00125       SUB(1)= 0.0
00126       GLEFT=  0.0
00127       IF (J1.EQ.1) GLEFT= DT/3.0
00128C
00129          END IF
00130C
00131       ELSE IF ( ID(K) .EQ. 0 ) THEN
00132C
00133       SUB(K)= DT/6.0
00134       GLEFT= DT/3.0
00135C
00136       ELSE IF ( ID(K) .EQ. -1 ) THEN
00137C
00138          IF (K.NE.1) THEN
00139C
00140              IF (J1.EQ.4) THEN
00141C
00142       SUB(K)= DT/6.0
00143       GLEFT=  DT/3.0
00144C
00145              ELSE IF (J1.EQ.3) THEN
00146C
00147       T= XL-(DT/DA)*AL
00148       W= 0.5*( X(K)+T )
00149       SUB(K)= (X(K)-T)/6.0 * ( ((T-XL)/DT)*((X(K)-T)/DT)
00150      C              + 4.0*((W-XL)/DT)*((X(K)-W)/DT) )
00151       GLEFT=  (X(K)-T)/6.0 * ( ((T-XL)/DT)**2
00152   -  C              + 4.0*(((W-XL)/DT)**2) + 1.0 )
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00153C
00154                ELSE IF (J1.EQ.2) THEN
00155C
00156          T= XL-(DT/DA)*AL
00157          W= 0.5*( T+XL )
00158          SUB(K)= (T-XL)/6.0 * ( 4.0*((W-XL)/DT)*((X(K)-W)/DT)
00159       C               + ((T-XL)/DT)*((X(K)-T)/DT) )
00160          GLEFT=  (T-XL)/6.0 * ( 4.0*(((W-XL)/DT)**2)
00161       C               + ((T-XL)/DT)**2 )
00162C
00163                ELSE IF (J1.EQ.1) THEN
00164C
00165          SUB(K)= 0.0
00166          GLEFT=  0.0
00167C
00168                END IF
00169C
00170            ELSE IF (K.EQ.1) THEN
00171C
00172          SUB(1)= 0.0
00173          GLEFT=  0.0
00174          IF (J1.EQ.4) GLEFT= DT/3.0
00175C
00176            END IF
00177C
00178          END IF
00179C
00180          IF (K.NE.1) THEN
00181          IF ( D(K-1) .EQ. 0.0 ) THEN
00182               SUB(K) = 0.0
00183               GLEFT  = 0.0
00184          END IF
00185          END IF
00186C
00187          DT= XR-X(K)
00188          DA= AR-A(K)
00189C
00190          IF ( ID(K+1) .EQ. 1 ) THEN
00191C
00192            IF (K.NE.N) THEN
00193C
00194                IF (J2.EQ.1) THEN
00195C
00196          SUP(K)= DT/6.0
00197          GRIGH=  DT/3.0
00198C
00199                ELSE IF (J2.EQ.2) THEN
00200C
00201          T= X(K)-(DT/DA)*A(K)
00202          W= 0.5*( XR+T )
00203          SUP(K)= (XR-T)/6.0 * ( ((T-X(K))/DT)*((XR-T)/DT)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00204     C              + 4.0*((W-X(K))/DT)*((XR-W)/DT) )
00205       GRIGH=  (XR-T)/6.0 * ( ((XR-T)/DT)**2
00206     C              + 4.0*(((XR-W)/DT)**2) )
00207C
00208               ELSE IF (J2.EQ.3) THEN
00209C
00210       T= X(K)-(DT/DA)*A(K)
00211       W= 0.5*( T+X(K) )
00212       SUP(K)= (T-X(K))/6.0 * ( 4.0*((W-X(K))/DT)*((XR-W)/DT)
00213     C              + ((T-X(K))/DT)*((XR-T)/DT) )
00214       GRIGH=  (T-X(K))/6.0 * ( 1.0 + 4.0*(((XR-W)/DT)**2)
00215     C              + ((XR-T)/DT)**2 )
00216C
00217               ELSE IF (J2.EQ.4) THEN
00218C
00219       SUP(K)= 0.0
00220       GRIGH=  0.0
00221C
00222               END IF
00223C
00224           ELSE IF (K.EQ.N) THEN
00225C
00226       SUP(N)= 0.0
00227       GRIGH=  0.0
00228       IF (J2.EQ.1) GRIGH= DT/3.0
00229C
00230           END IF
00231C
00232       ELSE IF ( ID(K+1) .EQ. 0 ) THEN
00233C
00234       SUP(K)= DT/6.0
00235       GRIGH=  DT/3.0
00236C
00237       ELSE IF ( ID(K+1) .EQ. -1 ) THEN
00238C
00239           IF (K.NE.N) THEN
00240C
00241               IF (J2.EQ.4) THEN
00242C
00243       SUP(K)= DT/6.0
00244       GRIGH=  DT/3.0
00245C
00246               ELSE IF (J2.EQ.3) THEN
00247C
00248       T= X(K)-(DT/DA)*A(K)
00249       W= 0.5*( XR+T )
00250       SUP(K)= (XR-T)/6.0 * ( ((T-X(K))/DT)*((XR-T)/DT)
00251     C              + 4.0*((W-X(K))/DT)*((XR-W)/DT) )
00252       GRIGH=  (XR-T)/6.0 * ( ((XR-T)/DT)**2
00253     C              + 4.0*(((XR-W)/DT)**2) )
00254C
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00255               ELSE IF (J2.EQ.2) THEN
00256C
00257         T= X(K)-(DT/DA)*A(K)
00258         W= 0.5*( T+X(K) )
00259         SUP(K)= (T-X(K))/6.0 * ( 4.0*((W-X(K))/DT)*((XR-W)/DT)
00260      C                + ((T-X(K))/DT)*((XR-T)/DT) )
00261         GRIGH=  (T-X(K))/6.0 * ( 1.0 + 4.0*(((XR-W)/DT)**2)
00262      C                + ((XR-T)/DT)**2 )
00263C
00264               ELSE IF (J2.EQ.1) THEN
00265C
00266         SUP(K)= 0.0
00267         GRIGH=  0.0
00268C
00269               END IF
00270C
00271           ELSE IF (K.EQ.N) THEN
00272C
00273         SUP(N)= 0.0
00274         GRIGH= 0.0
00275         IF (J2.EQ.4) GRIGH= DT/3.0
00276C
00277           END IF
00278C
00279         END IF
00280C
00281         IF (K.NE.N) THEN
00282         IF ( D(K+1) .EQ. 0.0 ) THEN
00283              SUP(K)= 0.0
00284              GRIGH = 0.0
00285         END IF
00286         END IF
00287C
00288         DIAG(K)= GLEFT+GRIGH
00289C
00290C
00291         IF ( D(K) .EQ. 0.0 ) THEN
00292               DIAG(K)= 1.0
00293               SUB(K)=  0.0
00294               SUP(K)=  0.0
00295         END IF
00296C
00297 125     CONTINUE
00298C
00299         DO 150 L=1,N
00300         H(L)= D(L)
00301 150     CONTINUE
00302C
00303C
00304C          WE SOLVE THE MATRIX EQUATION JX=H, THE ARRAY (H)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00305C        BEING IDENTICAL TO THE ARRAY (D).  THE SOLUTION
00306C        IS RETURNED IN THE ARRAY (H).
00307C
00308C
00309         CALL TRID(SUB,DIAG,SUP,H,N)
00310C
00311         SUM1= 0.0
00312         DO 200 L=1,N
00313         A(L)= H(L)
00314         SUM1= SUM1 + ABS(A(L))
00315 200     CONTINUE
00316C
00317C        THE FUNCTION EVALUATION SUBROUTINE COMPUT MAY
00318C        BE DELETED. IN THIS CASE THE FOLLOWING EIGHT
00319C        LINES ARE TO BE DELETED AND THE ARRAY (FX)
00320C        CAN BE TAKEN FROM THE REAL STATEMENT AT THE
00321C        BEGINNING OF THIS SUBROUTINE.
00322C
00323         CALL COMPUT(A,FX,N,X,TL,TR)
00324         FNORM1= 0.0
00325         DO 250 L=1,N
00326         FNORM1= FNORM1 + FX(L)*FX(L)
00327 250     CONTINUE
00328         FNORM1= SQRT(FNORM1)
00329         WRITE(6,300) LJ,FNORM1
00330 300     FORMAT(I5,E15.6)
00331C
00332C
00333         IF (LJ.NE.1) THEN
00334         RATIO= ABS(SUM1-SUM2)
00335         AB= EPS*SUM2
00336         IFLAG= 1
00337         IF (RATIO .LE. AB) GO TO 400
00338         END IF
00339         SUM2= SUM1
00340 350     CONTINUE
00341         IFLAG= 2
00342 400     CONTINUE
00343         ITMAX= LJ
00344         RETURN
00345         END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00001       SUBROUTINE COMPUT(A,FX,N,X,TL,TR)
00002C
00003C        SUBROUTINE (COMPUT), THE FUNCTION EVAULATING
00004C        SUBROUTINE, IS OPTIONAL.
00005C
00006C
00007       REAL A(N),FX(N),F1,ALO,AHI,TLO,THI
00008       REAL GLEF,GRIG,TS,X(N)
00009       INTEGER N,K,J1,J2
00010       COMMON D(50),ID(50)
00011       DO 100 K=1,N
00012C
00013       IF ( D(K) .NE. 0.0 ) THEN
00014C
00015       IF (K.EQ.1) THEN
00016                   ALO= 0.0
00017                   TLO= TL
00018       ELSE
00019                   ALO= A(K-1)
00020                   TLO= X(K-1)
00021       END IF
00022C
00023       IF (K.EQ.N) THEN
00024                   AHI= 0.0
00025                   THI= TR
00026       ELSE
00027                   AHI= A(K+1)
00028                   THI= X(K+1)
00029       END IF
00030C
00031C
00032C
00033       IF (ALO.GE.0.0 .AND. A(K).GE.0.0) J1= 1
00034       IF (ALO.LT.0.0 .AND. A(K).GE.0.0) J1= 2
00035       IF (ALO.GE.0.0 .AND. A(K).LT.0.0) J1= 3
00036       IF (ALO.LT.0.0 .AND. A(K).LT.0.0) J1= 4
00037C
00038       IF (A(K).GE.0.0 .AND. AHI.GE.0.0) J2= 1
00039       IF (A(K).LT.0.0 .AND. AHI.GE.0.0) J2= 2
00040       IF (A(K).GE.0.0 .AND. AHI.LT.0.0) J2= 3
00041       IF (A(K).LT.0.0 .AND. AHI.LT.0.0) J2= 4
00042C
00043       DT= X(K)-TLO
00044C
00045           IF ( ID(K) .EQ. 1 ) THEN
00046C
00047               IF (J1.EQ.1) THEN
00048C
00049       GLEF= DT*( 2.0*A(K) + ALO )/6.0
00050C
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00051                ELSE IF (J1.EQ.2) THEN
00052C
00053          TS= TLO - ALO*DT/( A(K)-ALO )
00054          F1= ( (TS+X(K))*0.5 - TLO )/DT
00055          GLEF= (X(K)-TS)*A(K)*( 2.0*F1 + 1.0 )/6.0
00056C
00057                ELSE IF (J1.EQ.3) THEN
00058C
00059          TS= TLO - ALO*DT/( A(K)-ALO )
00060          GLEF= ( (TS-TLO)**2 )*ALO/(6.0*DT)
00061C
00062                ELSE IF (J1.EQ.4) THEN
00063C
00064          GLEF= 0.0
00065C
00066                END IF
00067C
00068             ELSE IF ( ID(K) .EQ. 0 ) THEN
00069C
00070          GLEF= DT*( 2.0*A(K) + ALO )/6.0
00071C
00072             ELSE IF ( ID(K) .EQ. -1 ) THEN
00073C
00074                IF (J1.EQ.4) THEN
00075C
00076          GLEF= DT*( 2.0*A(K) + ALO )/6.0
00077C
00078                ELSE IF (J1.EQ.3) THEN
00079C
00080          TS= TLO - ALO*DT/( A(K)-ALO )
00081          F1= ( (TS+X(K))*0.5 - TLO )/DT
00082          GLEF= (X(K)-TS)*A(K)*( 2.0*F1 + 1.0 )/6.0
00083C
00084                ELSE IF (J1.EQ.2) THEN
00085C
00086          TS= TLO - ALO*DT/( A(K)-ALO )
00087          GLEF= ( (TS-TLO)**2 )*ALO/(6.0*DT)
00088C
00089                ELSE IF (J1.EQ.1) THEN
00090C
00091          GLEF= 0.0
00092C
00093                END IF
00094C
00095             END IF
00096C
00097          DT= THI-X(K)
00098C
00099             IF ( ID(K+1) .EQ. 1 ) THEN
00100C
00101                IF (J2.EQ.1) THEN
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00102C
00103       GRIG= DT*( 2.0*A(K) + AHI )/6.0
00104C
00105               ELSE IF (J2.EQ.2) THEN
00106C
00107       TS= X(K) - A(K)*DT/( AHI-A(K) )
00108       GRIG= ( (THI-TS)**2 )*AHI/(6.0*DT)
00109C
00110               ELSE IF (J2.EQ.3) THEN
00111C
00112       TS= X(K) - A(K)*DT/( AHI-A(K) )
00113       F1= ( THI-0.5*( TS+X(K) ) )/DT
00114       GRIG= ( TS-X(K) )*A(K)*( 1.0 + 2.0*F1 )/6.0
00115C
00116               ELSE IF (J2.EQ.4) THEN
00117C
00118       GRIG= 0.0
00119C
00120               END IF
00121C
00122           ELSE IF ( ID(K+1) .EQ. 0 ) THEN
00123C
00124       GRIG= DT*( 2.0*A(K) + AHI )/6.0
00125C
00126           ELSE IF ( ID(K+1) .EQ. -1 ) THEN
00127C
00128               IF (J2.EQ.4) THEN
00129C
00130       GRIG= DT*( 2.0*A(K) + AHI )/6.0
00131C
00132               ELSE IF (J2.EQ.3) THEN
00133C
00134       TS= X(K) - A(K)*DT/( AHI-A(K) )
00135       GRIG= ( (THI-TS)**2 )*AHI/(6.0*DT)
00136C
00137               ELSE IF (J2.EQ.2) THEN
00138C
00139       TS= X(K) - A(K)*DT/( AHI-A(K) )
00140       F1= ( THI-0.5*( TS+X(K) ) )/DT
00141       GRIG= ( TS-X(K) )*A(K)*( 1.0 + 2.0*F1 )/6.0
00142C
00143               ELSE IF (J2.EQ.1) THEN
00144C
00145       GRIG= 0.0
00146C
00147               END IF
00148C
00149           END IF
00150C
00151C
00152           IF (K.NE.1) THEN
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00153                IF ( D(K-1) .EQ. 0.0 ) GLEF= 0.0
00154              END IF
00155C
00156              IF (K.NE.N) THEN
00157                IF ( D(K+1) .EQ. 0.0 ) GRIG= 0.0
00158              END IF
00159C
00160         FX(K)= GLEF + GRIG - D(K)
00161C
00162         ELSE IF ( D(K) .EQ. 0.0 ) THEN
00163                                    FX(K)= 0.0
00164         END IF
00165C
00166 100     CONTINUE
00167         RETURN
00168         END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00001       SUBROUTINE POLY(A,T,PP,M,F,LI,TX)
00002C
00003C            SUBROUTINE POLY INTEGRATES BACK TWICE THE
00004C          POSITIVE PART OF THE PIECEWISE LINEAR SECOND
00005C          DERIVATIVE WHERE THE DATA SUGGESTS THAT THE
00006C          INTERPOLATING CURVE SHOULD BE CONVEX, THE
00007C          NEGATIVE PART OF THE PIECEWISE LINEAR SECOND
00008C          DERIVATIVE WHERE THE DATA SUGGESTS THAT THE
00009C          INTERPOLATING CURVE SHOULD BE CONCAVE, AND
00010C          THE REMAINING PORTION OF THE PIECEWISE LINEAR
00011C          SECOND DERIVATIVE ON THE TRANSITION INTERVALS.
00012C
00013C            THE INTEGRATION YIELDS A PIECEWISE CUBIC
00014C          POLYNOMIAL WITH KNOTS GIVEN BY THE SEQUENCE
00015C          (TX).  THIS CUBIC POLYNOMIAL INTERPOLATES THE
00016C          DATA AND ITS COEFFICIENTS ARE DENOTED BY THE
00017C          NUMBERS PP(J,I) - THE VALUE OF THE (J-1)ST
00018C          DERIVATIVE OF THE FUNCTION EVALUATED AT TX(I).
00019C          FOR X SUCH THAT TX(I).GE.X.LT.TX(I+1) THE VALUE
00020C          OF THE CUBIC POLYNOMIAL IS
00021C
00022C                  PP(1,I)
00023C           +      PP(2,I) * ( X-TX(I) )
00024C           + (1/2)PP(3,I) * ( X-TX(I) )**2
00025C           + (1/6)PP(4,I) * ( X-TX(I) )**3
00026C
00027C
00028       INTEGER M,J,L,LI
00029       REAL A(50),T(50),PP(4,100),F(50),TX(100),TAU
00030       REAL DF,DT,DA,C,E
00031       COMMON D(50),ID(50)
00032       LI= 1
00033       MN1= M-1
00034       DO 100 L=1,MN1
00035       DF= F(L+1)-F(L)
00036       DT= T(L+1)-T(L)
00037       DA= A(L+1)-A(L)
00038C
00039       JP= 0
00040       IF (L.EQ.1) THEN
00041                   IF ( D(1) .EQ. 0.0 ) JP= 1
00042       ELSE IF (L.EQ.MN1) THEN
00043                   IF ( D(M-2) .EQ. 0.0 ) JP= 1
00044       ELSE
00045                   C= D(L-1)*D(L)
00046                   IF ( C .EQ. 0.0 ) JP= 1
00047       END IF
00048C
00049C
00050C
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00051       IF (JP.EQ.1) THEN
00052C
00053       PP(4,LI)= 0.0
00054       PP(3,LI)= 0.0
00055       PP(2,LI)= DF/DT
00056       PP(1,LI)= F(L)
00057       TX(LI)= T(L)
00058       LI= LI+1
00059C
00060       ELSE IF (JP.EQ.0) THEN
00061C
00062       IF (A(L).GE.0.0 .AND. A(L+1).GE.0.0) J= 1
00063       IF (A(L).LT.0.0 .AND. A(L+1).GT.0.0) J= 2
00064       IF (A(L).GT.0.0 .AND. A(L+1).LT.0.0) J= 3
00065       IF (A(L).LE.0.0 .AND. A(L+1).LE.0.0) J= 4
00066C
00067           IF ( ID(L) .EQ. 1) THEN
00068C
00069               IF (J.EQ.1) THEN
00070C
00071       C= DF/DT - (DA/6.0 + A(L)/2.0)*DT
00072       PP(4,LI)= DA/DT
00073       PP(3,LI)= A(L)
00074       PP(2,LI)= C
00075       PP(1,LI)= F(L)
00076       TX(LI)= T(L)
00077       LI= LI+1
00078C
00079               ELSE IF (J.EQ.2) THEN
00080C
00081       TAU= T(L) - A(L)*DT/DA
00082       C= DF/DT - (A(L+1)**3)*DT/(6.0*DA*DA)
00083       PP(4,LI)= 0.0
00084       PP(3,LI)= 0.0
00085       PP(2,LI)= C
00086       PP(1,LI)= F(L)
00087       PP(4,LI+1)= DA/DT
00088       PP(3,LI+1)= 0.0
00089       PP(2,LI+1)= C
00090       PP(1,LI+1)= C*(TAU-T(L)) + F(L)
00091       TX(LI)= T(L)
00092       TX(LI+1)= TAU
00093       LI= LI+2
00094C
00095               ELSE IF (J.EQ.3) THEN
00096C
00097       TAU= T(L) - A(L)*DT/DA
00098       E= F(L) - (A(L)**3)*DT*DT/(6.0*DA*DA)
00099       C= DF/DT + (A(L)**3)*DT/(6.0*DA*DA)
00100       PP(4,LI)= DA/DT
00101       PP(3,LI)= A(L)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00102       PP(2,LI)= C + A(L)*A(L)*DT*0.5/DA
00103       PP(1,LI)= F(L)
00104       PP(4,LI+1)= 0.0
00105       PP(3,LI+1)= 0.0
00106       PP(2,LI+1)= C
00107       PP(1,LI+1)= C*( TAU-T(L) ) + E
00108       TX(LI)= T(L)
00109       TX(LI+1)= TAU
00110       LI= LI+2
00111C
00112               ELSE IF (J.EQ.4) THEN
00113C
00114       PP(4,LI)= 0.0
00115       PP(3,LI)= 0.0
00116       PP(2,LI)= DF/DT
00117       PP(1,LI)= F(L)
00118       TX(LI)= T(L)
00119       LI= LI+1
00120C
00121               END IF
00122C
00123           ELSE IF ( ID(L) .EQ. 0 ) THEN
00124C
00125       C= DF/DT - (DA/6.0 + A(L)/2.0)*DT
00126       PP(4,LI)= DA/DT
00127       PP(3,LI)= A(L)
00128       PP(2,LI)= C
00129       PP(1,LI)= F(L)
00130       TX(LI)= T(L)
00131       LI= LI+1
00132C
00133           ELSE IF ( ID(L) .EQ. -1 ) THEN
00134C
00135               IF (J.EQ.4) THEN
00136C
00137       C= DF/DT - (DA/6.0 + A(L)/2.0)*DT
00138       PP(4,LI)= DA/DT
00139       PP(3,LI)= A(L)
00140       PP(2,LI)= C
00141       PP(1,LI)= F(L)
00142       TX(LI)= T(L)
00143       LI= LI+1
00144C
00145               ELSE IF (J.EQ.3) THEN
00146C
00147       TAU= T(L) - A(L)*DT/DA
00148       C= DF/DT - (A(L+1)**3)*DT/(6.0*DA*DA)
00149       PP(4,LI)= 0.0
00150       PP(3,LI)= 0.0
00151       PP(2,LI)= C
00152       PP(1,LI)= F(L)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
00153        PP(4,LI+1)= DA/DT
00154        PP(3,LI+1)= 0.0
00155        PP(2,LI+1)= C
00156        PP(1,LI+1)= C*(TAU-T(L)) + F(L)
00157        TX(LI)= T(L)
00158        TX(LI+1)= TAU
00159        LI= LI+2
00160C
00161                ELSE IF (J.EQ.2) THEN
00162C
00163        TAU= T(L) - A(L)*DT/DA
00164        E= F(L) - (A(L)**3)*DT*DT/(6.0*DA*DA)
00165        C= DF/DT + (A(L)**3)*DT/(6.0*DA*DA)
00166        PP(4,LI)= DA/DT
00167        PP(3,LI)= A(L)
00168        PP(2,LI)= C + A(L)*A(L)*DT*0.5/DA
00169        PP(1,LI)= F(L)
00170        PP(4,LI+1)= 0.0
00171        PP(3,LI+1)= 0.0
00172        PP(2,LI+1)= C
00173        PP(1,LI+1)= C*( TAU-T(L) ) + E
00174        TX(LI)= T(L)
00175        TX(LI+1)= TAU
00176        LI= LI+2
00177C
00178                ELSE IF (J.EQ.1) THEN
00179C
00180        PP(4,LI)= 0.0
00181        PP(3,LI)= 0.0
00182        PP(2,LI)= DF/DT
00183        PP(1,LI)= F(L)
00184        TX(LI)= T(L)
00185        LI= LI+1
00186C
00187                END IF
00188C
00189            END IF
00190C
00191        END IF
00192C
00193 100    CONTINUE
00194        PP(4,LI)= 0.0
00195        PP(3,LI)= 0.0
00196        PP(2,LI)= 0.0
00197        PP(1,LI)= F(M)
00198        TX(LI)= T(M)
00199        RETURN
00200        END
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Subroutines TRID and DATAFL are listed in Appendix A.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Autobiographical Statement

The author was born in Ironton, Ohio on July 15, 1959. He received the Bachelor of Science degree in mathematics from Georgetown College in Georgetown, Kentucky in May 1981. He received the Master of Science degree in computational and applied mathematics from Old Dominion University in December 1982. As a doctoral candidate the author was supported by a fellowship by the Institute for Computational and Applied Mechanics. He is co-author along with Drs. Philip W. Smith and Samuel P. Marin of "Constrained Interpolation and Smoothing," General Motors Research Report Number MA-312.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.